Problem Set 8 Solutions

Problem 1.

We want to show that there exists a vector $\pi \geq 0$ such that $\pi P = \pi$, and $\sum \pi_i = 1$. This can be formulated as the following linear program:

$$\begin{array}{rcl} & & \min\ 0, \\ \pi \begin{pmatrix} & & 1 \\ & P - I & 1 \\ & & \vdots \\ & & 1 \end{pmatrix} & = & \begin{pmatrix} 0 & 0 & \dots & 0 & 1 \end{pmatrix}, \\ \pi & \geq & 0. \end{array}$$

Its dual is

$$\begin{array}{c} \max\ y, \\ \begin{pmatrix} & & 1 \\ & P - I & 1 \\ & & \vdots \\ & & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ y \end{pmatrix} \leq 0. \end{array}$$

The dual is obviously feasible ($x = 0$, $y = 0$), and we will show that y cannot be greater than 0. Fix an arbitrary vector x, and let k be an integer such that $x_k = \min_i x_i$. If we consider the k^{th} constraint, it looks like

$$p_{k,1}x_1 + \cdots + p_{k,k-1}x_{k-1} + (p_{k,k} - 1)x_k + p_{k,k+1}x_{k_1} + \cdots p_{n,n}x_n + y \leq 0,$$

and we can transform it into

$$\left(\sum_i p_{k,i}x_i\right) - x_k \leq -y.$$

Since all $p_{k,i}$ are nonnegative, and their sum equals 1, the sum in the inequality is a weighted average of x_i's, and is not less than x_k, so the left hand side of the inequality is nonnegative,

$$0 \leq -y,$$

and eventually,

$$y \leq 0.$$

We have shown that y, which we maximize, is bounded by 0 for any x, what together with the feasibility of the dual implies feasibility of the primal, and the required π exists.

Problem 2.

(a) If Alice's mixed strategy is known, then for each Bob's mixed strategy, Bob's payoff is a weighted average of Bob's payoffs for pure strategies. This implies that Bob can simply choose the pure strategy of the maximum payoff, and it serves as his best response.

(b) & (c) By part (a) the first of the two problems turns into:

$$\min_{\sum x_i = 1} (\text{maximum of columns of } xA).$$

We create an additional variable γ, an upper bound on all columns. The following linear program describes our situation:

$$\begin{aligned} & \min \gamma, \\ xA &\leq (\gamma\ \gamma\ \ldots\ \gamma), \\ \sum x_i &= 1, \\ x &\geq 0. \end{aligned}$$

Two last constraints guarantee that x is a mixed strategy. The first one says that γ is an upper bound on all columns, what in turn means that it is not smaller than the maximum of all columns of xA. Since we minimize γ, it will actually equal the maximum of all columns of xA, and we will choose such x that this maximum is minimum.

The second problem (by analogous argumentation) turns into:

$$\begin{aligned} & \max \delta, \\ Ay &\geq \begin{pmatrix} \delta \\ \delta \\ \vdots \\ \delta \end{pmatrix}, \\ \sum y_i &= 1, \\ y &\geq 0. \end{aligned}$$

(d) If we take the dual of the first linear program, we get

$$\begin{aligned} & \max \delta, \\ Ay &\leq -\begin{pmatrix} \delta \\ \delta \\ \vdots \\ \delta \end{pmatrix}, \\ -\sum y_i &= 1, \\ y &\leq 0, \end{aligned}$$

which after the substitution $y := -y$ is the second linear program. Since both linear programs are obviously feasible, by strong duality optimum γ and δ must be equal.

Problem 3.

(a) We will create consecutive clusters. Let S be the set of points that have not been assigned yet to any cluster. If S is empty, all remaining clusters will be empty. Otherwise, pick arbitrary point p in S, remove it with all points that are at most d far from it, and create of them a new cluster.

There is some optimal clustering $\mathcal{C}$. Any point that does not belong at some point to clusters that we have already created, has to belong to a different cluster in $\mathcal{C}$ than any point that we have picked up as a center so far. Since there are k clusters in $\mathcal{C}$, after k rounds S must be empty.

Furthermore, by the triangle inequality the distance between any two points in the same created cluster is not greater than $2d$, and therefore, we achieve a 2-approximation.

(b) As long as there is a point that is farther than d from each center chosen so far, it belongs to a different cluster in $\mathcal{C}$, and if there exists such a point, it must be the point of the maximum distance from already chosen centers, and since there are k clusters in $\mathcal{C}$, no point will be farther than d from all centers after k turns. This yields 2-approximation.

Problem 4.

(a) If jobs, in a feasible set of jobs, are not scheduled in order of increasing deadline, there are two adjacent that are not in this order, and it is enough to show that they can switch their positions. If we interchange them, the number of inversions will decrease by 1, and repeating this procedure, finally we get a scheduling in the required order.

Let A and B be the two aforementioned jobs of due dates, respectively, d_A, and d_B, where $d_A > d_B$, and job A precedes job B. If we switch A and B, B will still be done on time, and job A will be done before the deadline for B, so before the deadline for itself as well.

(b) Assume that jobs have assigned numbers from 1 to n, where n is the number of jobs, in order of increasing deadline. Jobs that will be completed on time can be processed in this order, the other can be completed at the end, since they miss their deadlines anyway.

Let $A(j, w)$, where $0 \leq j \leq n$ and $0 \leq w \leq \sum w_i$, be the minimum processing time of first j jobs under a penalty w. Jobs responsible for w are scheduled somewhere far in the future, and we pay due penalties for them. $A(j, w)$ can be recursively described as the minimum of the following two numbers

- $A(j-1, w-w_j)$ — we do not schedule job j,

- $A(j-1,w)+p_j$ — if we schedule job j, and we consider this option **only if** $A(j-1,w)+p_j \le d_j$, i.e. job j does not miss its deadline.

We can assume that boundary A's have value ∞, except $A(0,0)$ which is equal to 0. The table of numbers $A(\cdot,\cdot)$ can be computed dynamically in time $O(n \cdot \sum p_j)$. Next we find finite $A(n,w)$ of the minimum value w, and $A(n,w)$ is the fastest completing time of some feasible subset of the maximum weight (which corresponds to the minimum cost of the jobs completed after deadlines). Using table A, we can quickly reconstruct the appropriate feasible set.

(c) Let w^* be the optimum total penalty. We can assume that $w^* > 0$. It can be checked in the beginning whether the set of all jobs is a feasible set, to avoid the situation when $w^* = 0$.

We scale each penalty w_j to $\frac{n}{\varepsilon w^*} w_j$, and round down to $\lfloor \frac{n}{\varepsilon w^*} w_j \rfloor$. The optimum penalty will be first scaled to $\frac{n}{\varepsilon}$, and since rounding down decreases the penalty of any scheduling by less than n, the optimum solution under the modified weights has had, before the rounding down, a penalty of at most $\frac{n}{\varepsilon} + n = (1+\varepsilon)\frac{n}{\varepsilon}$, so finding the optimum penalty for the modified weights, we find a good approximation for the original problem.

We can find the exact solution for the modified weights in time $O(\frac{n^2}{\varepsilon})$, using the dynamic programming from part (b).

We do not know w^*, but we can quickly determine its good approximation, using binary search, as we did for other problems in class. We know that $w^* \ge \min w_j$, and $w^* \le n \max w_j$, so we need only a polynomial, in size of representations of numbers, number of guesses to get into $1 \pm \varepsilon$ of w^*.