Algorithm Design

Homework 2

Pietro Spadaccino, 1706250

References

[1] S. Canzar, T. Marschall, S. Rahmann, C. Schwiegelshohn. *Solving the Minimum String Cover Problem*, SIAM, 2011.

Exercise 1

Algorithm

We can represent the problem by a graph $G(V, E)$ with people as nodes and friendship as a weighted undirected edges. We define a function $\delta(x \in V)$ returning the weights of edges incident to a node: $\delta(x) = \sum_{x_1 \in V} w(x, x1)$. We will use the following greedy algorithm: it initializes by setting $S_0 = G$, and then iterates for all $i \in \{1, ..., |V|/2 - 1\}$ by finding $m = \arg\min_{x \in S_i \cap M} \delta(x)$ and $f = \arg\min_{x \in S_i \cap F} \delta(x)$ and finally setting $S_{i+1} = S_i \setminus \{m, f\}$. In words, the algorithm removes from the current S_i the nodes m, f with minimum "weight" $\delta(m)$ and $\delta(f)$ calculated on S_i. Once the loop is ended the algorithm will return the densest subgraph among all S_i. The algorithm is terminated and it provides a 2-approximation of the optimum.

Running time

The time complexity is equal to a call of δ (costing $|E|$) times $|V|$ nodes in a S_i times $|V|/2$ iterations, $O(|E||V|^2)$.

Proof of correctness

Before starting, we extend the function δ to set of nodes $\delta(S \subseteq V) = \sum_{x \in S} \delta(x)$. We call $\delta^* = \frac{\delta(S^*)}{2|S^*|}$ the optimal density, where we divide by 2 because otherwise we would count twice the weight of an edge. We first want to show that $\delta(\{m, f\}) \geq 2\,\delta^*$, with $m, f \in S^*$: since S^* has the optimal density, we have $\frac{\delta(S^*)}{|S^*|} \geq \frac{\delta(S^* \setminus \{m,f\})}{|S^*|-2} \geq \frac{\delta(S^*)-2\delta(\{m,f\})}{|S^*|-2}$. The term $-2\delta(\{m, f\})$ is due to the fact that when we remove m, f from S, we have that $\delta(S)$ is surely decreased by $\delta(m) + \delta(f)$, and in addition every node n that was connected to m or f will have its $\delta(n)$ reduced, because there will no longer be the edge connecting n to m or f. Therefore $\delta(S)$ cannot be decreased more than $2\delta(\{m, f\})$. Back to the last inequality, we can rearrange it and obtain $\delta(\{m, f\}) \geq \frac{\delta(S^*)}{|S^*|} = 2\delta^*$.

Now we consider a suboptimal run of the algorithm, where some nodes $m', f' \in S^*$ are removed from the current S. Note that these nodes may be removed from S in different iterations. By definition of the algorithm, for every $m, f \in S$, we have $\delta(m) \geq \delta(m')$, $\delta(f) \geq \delta(f') \rightarrow \delta(\{m, f\}) \geq \delta(\{m', f'\}) \geq 2\delta^*$ for the first claim. Since we are under the assumption of $|M| = |F|$ we have $|S|/2$ couples m, f and we can write $\delta(S) \geq \frac{|S|}{2} 2\delta^*$. From here the density is given by $\frac{\delta(S)}{2|S|} \geq \frac{\frac{|S|}{2} 2\delta^*}{2|S|} = \frac{\delta^*}{2}$.

Exercise 2

We denote by S the set of requested skills, with $n = |S|$, and by P the set of people, where every person $p \in P$ is represented as a subset of skills $p \subseteq S$. We use boolean variables $x_{p \in P}$ telling whether or not person p is taken. The integer programming model can be obtained from the one shown in class and changing the constraints on the minimum number of times that a skill has to appear from 1 to 3. Our algorithm will make use of the solution of the relaxed problem, with variables $x_p^* \in [0, 1]$, solvable in polynomial time.

repeat
  for $i \in \{0,...,K\text{-}1\}$ **do**
    $C_i = \emptyset$
    for $p \in P$ **do**
      $C_i = C_i \cup \{p\}$ with probability x_p^*
    end
  end
  $C_K = C_0 \cup ... \cup C_{K-1}$
until C_K is not admissible

For a solution C_K to be admissible it must be feasible *and* its cost must fall below some approximation factor f, so we have $Pr(C_K \text{ admissible}) = Pr(C_K \text{ feasible}) Pr(\text{cost}(C_K) \leq f\text{OPT}_{IP}) = \bar{p}$. We define a random variable Y which will count the number of rounds required by the algorithm to find an admissible solution C_K. We can write the probability distribution of Y like so $Pr(Y = r) = \bar{p}(1-\bar{p})^{r-1}$ and its expected value is equal to $\mathbf{E}[Y] = \frac{1}{\bar{p}}$. Let's bound the expected number of rounds (number of times the outer loop is executed) to a number R, thus obtaining:

$$\mathbf{E}[Y] = \frac{1}{\bar{p}} \leq R \tag{1}$$

We have found a relation between the expected number of rounds R and the probability $\bar{p}$ to find a solution in a generic round. Now we want to explicit the two factors of $\bar{p}$, starting from the probability of a fair approximation of the cost and then the probability of its feasibility.

We start by defining a boolean random variable X_p which has value 1 if $p \in C_i$ for some i, 0 otherwise (as we defined the algorithm we have $P(X_p = 1) = x_p^*$). We have that the cost of an approximated solution is given by $\text{cost}(C_K) = \sum_{p \in C_K} w(p)$ and its expected value is $\mathbf{E}[\text{cost}(C_K)] \leq \mathbf{E}[\sum_{i \in K} \sum_{p \in P} w(p) X_p]$ which for the linearity is equal to $K \sum_{p \in P} w(p) \mathbf{E}[X_p] = K \sum_{p \in P} w(p) x_p^* = K\text{OPT}_{LP} \leq K\text{OPT}_{IP}$. For Markov's inequality we have:

$$Pr(\text{cost}(C_K) \geq apx\, K\, \text{OPT}_{IP}) \leq \frac{1}{apx} \tag{2}$$

where $apx > 1$. Reasoning about the complementary event, we have found that the probability for a C_K to approximate the optimal cost with a factor of $(Kapx)$ is greater than $1 - 1/apx$. We will go back to this equation later.

Our next goal is to find the probability that a solution C_K is feasible in a generic round. We define an auxiliary random variable Z_s counting how many times s is covered in a single C_i. The probabilities $Pr(0 \leq Z_s < 3)$ have upper bounds, which for now are $Pr(Z_s = 0) \leq b_0 = \frac{1}{e^3}$, $Pr(Z_s = 1) \leq b_1$, $Pr(Z_s = 2) \leq b_2$. I can count how many times a skill s is covered in a generic C_K by adding the times that it was covered in the K different C_i, so I can write $Pr(s \text{ covered less 3 times in } C_K) \leq Pr(Z_s = 0)^K + K\,Pr(Z_s = 1)Pr(Z_s = 0)^{K-1} + K\,Pr(Z_s = 2)Pr(Z_s = 0)^{K-1} + K^2\,Pr(Z_s = 1)^2Pr(Z_s = 0)^{K-2} \leq b_0^K + Kb_1b_0^{K-1} + Kb_2b_0^{K-1} + K^2b_1^2b_0^{K-2} \leq K^2b_0^K\left(1 + \frac{b_1}{b_0} + \frac{b_2}{b_0} + \frac{b_1^2}{b_0^2}\right) = K^2b_0^Kb$. Now we want to set the number of repetitions $K = d\log n$ and, remembering that $b_0 = \frac{1}{e^3}$, we obtain $Pr(s \text{ not covered in } C_K) \leq d^2(\log n)^2\frac{b}{n^{3d}}$. I can set the parameter d such that:

$$Pr(s \text{ not covered in } C_K) \leq d^2(\log n)^2\frac{b}{n^{3d}} \leq \frac{1}{rn} \tag{3}$$

for some parameter $r > 1$ which we will discuss later. Since we have n skills and remembering that in order to C_K to be unfeasible at least one skill must not be well covered, we can write $Pr(C_K \text{ not feasible}) \leq nPr(s \text{ not covered in } C_K) \leq \frac{1}{r}$.

Now we are ready to explicit equation 1. We know that a C_K is admissible when it respects the constraints and when its cost is a $(Kapx)$-approximation of the optimum and so we have $Pr(C_K \text{ admissible}) = \bar{p} = (1 - Pr(C_K \text{ not feasible}))(1 - Pr(\text{cost}(C_K) \geq apx\,K\,\text{OPT}_{IP}))$. For equation 1 we can write:

$$\mathbf{E}[Y] = \frac{1}{\bar{p}} \leq \frac{r}{r-1}\frac{apx}{apx-1} \leq R \quad apx, r > 1 \tag{4}$$

This inequality gives us a relation between the expected number of rounds R and the approximation factor $(Kapx)$. We can see that increasing K, which is the number of times that people get selected with prob. x_p^*, the approximation factor increases too, because we would have a solution composed by more people, thus more costly. On the other hand, if K increases we can increase r using equation 3 as its own upper bound. We have that the parameter r influences the number of rounds: indeed the probability that a solution is not feasible is inversely proportional to r as we can see from equation 3 and this behavior is reflected by the leftmost part of equation 4 tending to 1 when $r \to \text{inf}$. But r influences also the approximation factor $(Kapx) = d\log n\,apx$ since it is related to d by the equation 3: the more we increase r the more d has to increase and thus also the approximation factor. Specifically, if n is sufficiently large, d increases with $\approx \log_n(r)$. Also the parameter apx will act as a balance between the approximation and the expected number of rounds, but it will impact linearly on the approximation factor, so it makes sense to choose apx as low as possible. *Observation on finding b*: for $b_0 = e^{-3}$ it was obtained by setting all probabilities to $3/k$ and then tending k to inf. If we do the same with the other bounds we have $b_1 = 3e^{-3}, b_2 = 6e^{-3}$ thus obtaining $b = 13$.

Exercise 3

Let c be a cut of the graph, $w(c)$ will be the weight of the cut-set of c.

We start by considering $F^* \subseteq E$ to be the optimal choice of F. If we remove F^* from E we obtain k different connected components $C_1, ..., C_k$. *Observation 1:* We know that C_i is a cut isolating a single $t_i = s_j$ for some j and, since the proposed solution takes F_i as the mincut isolating t_i, we can write $w(F_i) \leq w(C_i)$ for $i = 1, ..., k$. *Observation 2:* $\sum_{i=1}^{k} w(C_i) = 2w(F^*)$, because each edge in F^* has endpoints in two different connected components and its weight will be counted twice in the sum. Combining the two observations we obtain:

$$w(F) \leq \sum_{i=1}^{k} w(F_i) \leq \sum_{i=1}^{k} w(C_i) = 2w(F^*)$$

This proves the approximation factor.

Exercise 4

We will use a model similar to the one reported in [1]. Before starting I define a function $I(g \in G) := \{(i_1, j_1), (i_2, j_2), ...\}$ returning –in poly-time– a set of tuples (i, j) if gene g is a substring of D from index i to j. Note that generally $|I(g)|$ can be greater than one since g can appear multiple times in D. We define a directed unweighted graph having as nodes the indices of our string $V := \{0, 1, ..., |D|\}$ and as edges $E := \cup_{g \in G} I(g)$ representing all tuples of indices for each gene. Our goal is to find a path in this graph, from node 0 to node $|D|$. Moreover we define $\delta^-(v \in V)$ the set of edges going to node v, $\delta^+(v)$ the ones outgoing from v and $V^* = V \setminus \{0, |D|\}$ the set of nodes without the first and the last index of D. I use x_g variables telling whether or not a gene g is taken and variables z_e telling whether or not an edge e is taken, and I can write the model:

$$\min \sum_{g \in G} w(g) x_g \tag{5a}$$

$$\sum_{e \in I(g)} z_e \leq |I(g)|\, x_g \quad \forall g \in G \tag{5b}$$

$$\sum_{e \in \delta^+(0)} z_e = 1 \tag{5c}$$

$$\sum_{e \in \delta^-(v)} z_e = \sum_{e \in \delta^+(v)} z_e \quad \forall v \in V^* \tag{5d}$$

$$x_g, z_e \in \{0, 1\} \tag{5e}$$

Now I define $S(e \in E) = \{g \in G \mid e \in I(g)\}$ returning the set of genes that have created the edge e. Note that $|S(e)| = 1$ since an edge cannot be created by two different g_1, g_2, unless $g_1 = g_2$. We also define $T^{\pm}(e \in E) = \{v \in V^* | e \in \delta^{\pm}(v)\}$. The following is the dual problem of the relaxation of the primal:

$$\max q \tag{6a}$$

$$-\sum_{v \in T^+(e)} r_v + q \leq \sum_{g \in S(e)} p_g \quad \forall e \in \delta^+(0) \tag{6b}$$

$$-\sum_{v \in T^+(e)} r_v + \sum_{v \in T^-(e)} r_v \leq \sum_{g \in S(e)} p_g \quad \forall e \notin \delta^+(0) \tag{6c}$$

$$|I(g)|\, q_g \leq w(g) \quad \forall g \in G \tag{6d}$$

$$p_g \geq 0 \tag{6e}$$

The structure of the dual problem was obtained by assigning one variable per constraint and one constraint per variable in the primal problem. Its objective function has to be maximized and it is given by the only non-zero constant term in the primal, in constraint 5c, times its associated variable q. Finally the constraints of the dual were obtained isolating the coefficient matrices in the primal problem, transposing them and then writing them as a sum of variables.

Exercise 5

Comet and Dasher will refuse to play if the opponent of one of them has an expected win value greater than its own one, so both players must have the same expected value of hay balls. Let be h_C, t_C, h_D, t_D respectively the probabilities of the events head for Comet, tail for Comet, head for Dasher and tail for Dasher. Given the rules of the game, I can write the equality between expected values for both players:

$$4h_C h_D + 2t_C t_D - 1h_C t_D - 2t_C h_D = -4h_C h_D - 2t_C t_D + 1h_C t_D + 2t_C h_D$$

Given that we are biasing coins, we have $h_C + t_C = 1$ and $h_D + t_D = 1$, so the above equation can be rewritten as

$$9h_C h_D - 3h_C - 4h_D + 2 = 0$$

The biasing must respect this equality.

Exercise 6

I define $P_n = Pr(\text{home} \mid \text{start at} = n)$ as the probability of waking up in hospital having started to walk at position n. From the specifications of the problem I can set up the following recurrence equation:

$$P_n = pP_{n-1} + qP_{n+1}$$

having as base cases $P_{-1} = 0$ and, if we assume that the home is at position N, $P_N = 1$. Our goal is to find a formulation of $1 - P_0$, the event of going to the hospital starting at position 0 for $N \to \text{inf}$.

Since $p + q = 1$, I can write $P_n = pP_n + qP_n = pP_{n-1} + qP_{n+1}$ and obtain $P_{n+1} - P_n = \frac{p}{q}(P_n - P_{n-1})$. Using this equality, and the base case $P_1 - P_0 = \frac{p}{q}P_0$, I can write the recurrence equation in a closed form:

$$P_{n+1} - P_n = P_0 \left(\frac{p}{q}\right)^{n+1} \tag{7}$$

Observation: we have $\sum_{i=0}^{n}(P_{i+1} - P_i) = P_{n+1} - P_0$ because if we expand the sum all terms will cancel out except for $P_{n+1} - P_0$. Using this observation and the equality 7 we have:

$$P_{n+1} - P_0 = \sum_{i=0}^{n}(P_{i+1} - P_i) = P_0 \sum_{i=0}^{n} \left(\frac{p}{q}\right)^{i+1} \to P_{n+1} = P_0 \sum_{i=0}^{n+1} \left(\frac{p}{q}\right)^{i}$$

We can now set $n = N - 1$:

$$P_N = P_0 \sum_{i=0}^{N} \left(\frac{p}{q}\right)^{i} \tag{8}$$

We observe that we have a geometric series and it converges when $\frac{p}{q} < 1 \to p < q$. Since $P_N = 1$ by definition, when $N \to \text{inf}$ we have:

$$P_N = 1 = P_0 \frac{1}{1 - \frac{p}{q}} \to P_0 = 1 - \frac{p}{q},\ p < q$$

If we want the probability of waking up in hospital to be less than 0.5 we have to set $1 - P_0 \leq 0.5$ resulting in $0 \leq p \leq \frac{1}{3}$. If otherwise $p \geq q$ we can use equation 8 to write $P_N = 1 \geq P_0(N + 1)$ and if N tends to infinity P_0 tends to 0. Therefore for $p \geq q$ Giorgio will always wake up in hospital.