Homework2
Algorithm Design

Lombardo Andrea 1893440
Di Spazio Fabio 1876540

Engineering in Computer Science La Sapienza

10/01/2020

1 Exercise 1- Andrea Lombardo 1893440-Di Spazio Fabio 1876540

It's an instance of an unbalanced assignment problem. Giving the following relationship $|F| \leq |O|$ where F is the set of Michele' friends and O the set of Michele' outfits we analyze the unbalanced assignment problem as F the set of resource and O the set of tasks.The problem is composed by a bipartite graph composed by the two sets O and F where each edge between the two sets has a weight $w_{(f,o)}$ >0. We define variable $x_{(f,o)}$ that could be 0 or 1: if the variable is 1 is the case that the selected edge (f,o) is contained in the matching and 0 otherwise. The assignment problem could be solved by presenting it as an integer linear program.

$$\begin{aligned} minimize \sum_{f\,\in\,F}\sum_{o\,\in\,O} w_{(f,o)}x_{(f,o)} \\ subject\ to \sum_{o\,\in\,O} x_{(f,o)} = 1\ with\ f = 1,2..|F| \\ \sum_{f\,\in\,F} x_{(f,o)} \leq 1\ with\ o = 1,2..|O| \\ x_{(f,o)} \in \{0,1\}\ with\ f = 1,2..,|F| \\ and\ o = 1,2...,|O| \end{aligned} \qquad \begin{aligned} minimize \sum_{f\,\in\,F}\sum_{o\,\in\,O} w_{(f,o)}x_{(f,o)} \\ subject\ to \sum_{o\,\in\,O} x_{(f,o)} = 1\ with\ f = 1,2..|F| \\ \sum_{f\,\in\,F} x_{(f,o)} \leq 1\ with\ o = 1,2..|O| \\ x_{(f,o)} \geq 0\ with\ f = 1,2..,|F| \\ and\ o = 1,2...,|O| \end{aligned}$$

How we can see above: left part represents the developing of the ILP instead the right part represents problem's LP-relaxation. Relaxing the problem the last constraint (integer constraint) is the only element that changes between them.

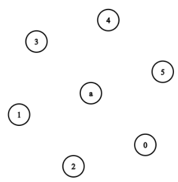
For the second part first we change the second constraint in $\sum_{f\,\in\,F} x_{(f,o)} = 1\ with\ o = 1,2..|O|$. Then we decide to transform our problem from an unbalanced to a balanced problem by adding to the set F a number of $|O| - |F|$ vertices. These vertices are linked through edges with the $|O|$ vertices with the components $x_{(f,o)}$ such that is true the constraint above mentioned. Now given a fractional optimal solution for the LP, let denote the number of nonintegral components $x_{(f,o)}$ by **counter** variable. If counter=0 it means that there are not more nonintegral values so we have reached an integral optimal solution; in case counter>0 means that exists a cycle composed by a set of nonintegral components $x_{(f,o)}$. Now from the edges cycle we selected or the edge with lowest component $x_{(f,o)}$ value from which subtracts a α quantity such that $x_{(f,o)} = x_{(f,o)} - \alpha$=0 or the edge with higher component $x_{(f,o)}$ value from which adds a α quantity such that $x_{(f,o)} = x_{(f,o)} + \alpha$=1. Now according to the first choice i.e. we've selected the higher component $x_{(f,o)}$ value we should add to it the α quantity subtracting the same α quantity to next $x_{(f,o)}$ and continuously adding and subtracting the same quantity the following $x_{(f,o)}$. Now we have at least one less fraction variable, so **counter** is decreased by 1. By repeating the procedure we arrive to the solution in which all variables are integral, this happens after at most "counter" step.

2 Exercise 2– Andrea Lombardo 1893440-Di Spazio Fabio 1876540

part a The problem of Chris can be treated as a vertex-cover problem. The set B' of vertices returned by the algorithm is a vertex cover, since the algorithm checks all the edges $e_1, e_2, ..., e_m$ of the edge set E of G is already contained in B'. Now we want to prove that $E[|B'|] \leq 2\,|OPT(B)|$ we take into account the fact that when the algorithm add a vertex to B' the probability that this vertex belongs to OPT(B) is at least $\frac{1}{2}$ because each edge is covered at least 1 vertex of the OPT(B). Now we build the two sets A=B'∩OPT(B) and C=B'-OPT(B) where respectly the first one is composed by the set of vertices selected by the algorithm and present also in the OPT(B), the second one instead contains the set of vertices selected by the algorithm but not present in the OPT(B). Let be X_e, where e is an edge $\in$ E, a random variable which is set to 1 in case the endpoint of e selected by the algorithm is contained in A, instead is set to 0 in case the endpoint of e selected by the algorithm is in C.
Thus this allows us to write $E[X_e] \geq \frac{1}{2}$, $|A| = \sum_{e\,\in\,E} X_e$ and $|C| = \sum_{e\,\in\,E}(1 - X_e)$ so follow:
E$[|A|] = E[\sum_{e \in E} X_e] = \sum_{e \in E} E[X_e] \geq \frac{|E|}{2}$ similarly
E$[|C|] = E[\sum_{e \in E}(1 - X_e)] = \sum_{e \in E} E[1] - \sum_{e \in E} E[X_e] = |E| - E[|A|] \leq \frac{|E|}{2}$
In this way we can observe that combining the two inequalities we have: E$[|A|] \geq E[|C|]$.
By definitions we have $|A| \leq |OPT(B)|$ hence $E[|A|] \leq |OPT(B)|$.
$E[|B'|] = E[|C| + |A|] = E[|C|] + E[|A|] \leq 2E[|A|] \leq 2\,|OPT(B)|$

part b We can distinguish two cases:

Graph with c vertices and no edges:

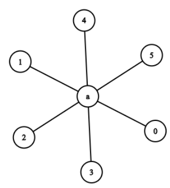


The algorithm will add all isolated vertices to B' so: $|B'| = C$ but $|OPT| = 0 \Rightarrow \forall\, c \geq 1\,|B'| \geq c\,|OPT|$ =$|B'| \geq 0$

Star graph with $c + 1$ vertices: In this case $|OPT|$=1

worst case The algorithm might add c vertices(all vertices expect the center "a".) Finally we get $|B'| = c\,|OPT|$ for any fixed values of c

luckiest case the algorithm might add "a", the center, to B' $\Rightarrow |OPT| = 1, |B'| = 1$
then $|B'| = |OPT|$



part c

3 Exercise 3- Andrea Lombardo 1893440-Di Spazio Fabio 1876540

	Rock	Paper	Scissor
Rock	a,0	a-1,1	a+1,-1
Paper	+1,-1	0,0	-1,1
Scissor	-1,+1	+1,-1	0,0

The probabilities that Philip plays the Rock is p_R,the Paper is p_P, the Scissor is p_S; now if we consider the probabilities of a second player, for example Bob, let be b_R,b_P and b_S the probabilities that he will play respectly Rock,Paper,Scissor. In order to compute the mixed Nash Equilibrium Philip adopts a strategy such that Bob' choices must not be influenced by own choices. The same concept is applied by Bob.

case $\alpha < 1$**:** From Bob's point of view the Mixed Nash Equlibrium is:

E[R]= $\alpha b_R + (\alpha - 1)b_P + (\alpha + 1)b_S$

E[P]= $+1b_R + 0b_P - 1b_S$

E[S]= $-1b_R + 1b_P + 0b_S$

$$\begin{cases} E[R] = E[P] \\ E[P] = E[S] \\ b_R + b_P + b_S = 1 \end{cases}$$

Solving the system we obtain: $b_R = \frac{1}{3}$, $b_P = \frac{1}{3} + \frac{\alpha}{3}$, $b_S = \frac{1}{3} - \frac{\alpha}{3}$
From Philips's point of view the Mixed Nash Equlibrium is:

E[R]= $0p_R + -1p_P + 1p_S$

E[P]= $+1p_R + 0p_P - 1p_S$

E[S]= $-1p_R + 1b_P + 0p_S$

$$\begin{cases} E[R] = E[P] \\ E[P] = E[S] \\ p_R + p_P + p_S = 1 \end{cases}$$

Solving the system we obtain: $p_R = p_P = p_S = \frac{1}{3}$
Now the expected Philip's payoff is evaluated:
Payoff=$p_R b_R \alpha + p_R b_P(\alpha - 1) + p_R b_S(\alpha + 1) + p_P b_R - p_P b_S - p_S b_R + p_S b_P = \frac{\alpha}{3}$

case $\alpha \geq 1$

- $1 \leq \alpha < 2$

  From the point of view of Philip, considering the rows of matrix we can state that$\forall\ \alpha \in [1, 2)$ Philip will play anyway Rock on Paper because Rock strategy dominates Paper strategy. Thus we can remove the second row of the matrix, then after this operation from the point of view of Bob we can delete the Scissor column from the matrix because the rock strategy dominates Scissor strategy. Let's now compute the Nash Equilibrium of the new matrix 2x2. Solving the systems for both players we obtain:
  $p_R = \frac{2}{3}$ $p_P = 0$ $p_S = \frac{1}{3}$ $b_R = \frac{2-\alpha}{3}$ $b_P = \frac{1+\alpha}{3}$ $b_S = 0$
  Payoff=$p_R b_R \alpha + p_R b_P(\alpha - 1) - p_S b_R + p_S b_P = \frac{2\alpha}{3} - \frac{1}{3}$

- $\alpha \geq 2$

  With the same procedure, that we've explained above, Philip choices to play Rock strategy over Scissor strategy instead Bob prefers playing Paper strategy over Rock strategy.Now we delete the corresponding Scissor row and Rock column obtaining a 1x1 matrix. The calculus of the Mixed Nash Equilibrium for both players are:
  $p_R = 1$ $p_P = 0$ $p_S = 0$ $b_R = 0$ $b_P = 1$ $b_S = 0$
  Payoff=$p_R b_P(\alpha - 1) = \alpha - 1$

4 Exercise 4- Andrea Lombardo 1893440-Di Spazio Fabio 1876540

In order to prove that there are $O(log\ n)$ students worthy of an award let's sort the n students according to their scores i.e. $a_1 > a_2 > ... > a_n$; in this way the first student gets the award. Then we will have that the student 2 gets an award if he scored better in "b" exercise, so if $b_2 > b1$; the student 3 gets an award if $b_3 > b_2, b_1$ and so on.
Let be A_i a random variable $\in 0, 1$ which value is 1 if the i^{th} student gets the award, 0 otherwise. We have

$$Pr[A] = \sum_{i=1}^{n} E[A_i] = \sum_{i=1}^{n} \frac{1}{i}$$

This expression $\sum_{i=1}^{n} \frac{1}{i} = 1 + \frac{1}{2} + \frac{1}{3} + ... + \frac{1}{n}$ is the *harmonic number H(n)* and as a function of n it is closely shadows the value $\int_1^{n+1} \frac{1}{x} dx = log(n+1)$. So $log(n+1) < H(n) < 1 + log\ n$
In order to show that there are $O(log\ n)$ students worthy of an award with high probability we can use Chernoff Bound. Since the scores are uniformly distributed in [0,1] and there aren't relationship among students, let X be the sum of n independent random variables; through Chernoff Bound we will bound the probability that X deviates above its expected value.

$$Pr(X > (1+\delta)\mu) \leq e^{\frac{-\delta^2\mu}{3}},\ 0 \leq \delta \leq 1 \tag{1}$$

Let $\mu = E[X] = log\ n$ the sum's expected value and set $\delta = \frac{1}{2}$. Substituting in (1) we obtain:

$$Pr(X > \frac{3}{2} log\ n) \leq e^{\frac{-\log(n)}{12}} \tag{2}$$

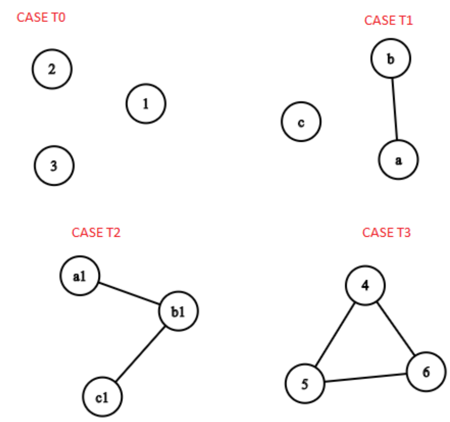
$$Pr(X > \frac{3}{2} log\ n) \leq \frac{1}{\sqrt[12]{n}} \tag{3}$$

From (3) we can see that the probability that the winners are asymptotically more than $log\ n$ goes to zero quickly as n goes to infinity.

5 Exercise 5 -Andrea Lombardo 1893440-Di Spazio Fabio 1876540

In order to see that the $|T_1| + 2|T_2| + 3|T_3| = |E|(|V| - 2)$ is true, let's consider a run of the algorithm. The algorithm first selects an edge, then it chooses a node $v \in V$ different from the endpoint of the selected edge and it establishes through the four sets $T_{i,\ i\in[0,3]}$ if there is a connection among the vertices. So, when the algorithm is executed, four type of subgraph exist depending on the number of the edges connecting vertices: It's simple to verify that for each of the above structure, the statement holds.

- **case T0**$\Rightarrow$ we have $|E| = 0, |T_0| = 1, |T_1| = 0, |T_2| = 0, |T_3| = 0$ so we have $\Rightarrow$ 0(3-2)=0+2(0)+3(0)
- **case T1**$\Rightarrow$ we have $|E| = 1, |T_0| = 0, |T_1| = 1, |T_2| = 0, |T_3| = 0$ so we have $\Rightarrow$ 1(3-2)=1+2(0)+3(0)
- **case T2**$\Rightarrow$ we have $|E| = 2, |T_0| = 0, |T_1| = 0, |T_2| = 1, |T_3| = 0$ so we have $\Rightarrow$ 2(3-2)=0+2(1)+3(0)
- **case T3**$\Rightarrow$ we have $|E| = 3, |T_0| = 1, |T_1| = 0, |T_2| = 0, |T_3| = 1$ so we have $\Rightarrow$ 3(3-2)=0+2(0)+3(1)



How we can see there are different case which depends on the sub-graph composed by the 3 vertices we consider: if we are in the first case $|T_0| = 1$ but we don't take in consideration in the formula because it doesn't have any edge; in the second case T1, we have exactly one edge in the vertex triple, so it could help to build the equation because we have that only $|T_1| = 1$; number of times that I select the three vertices but every time I selected a different edge. So in the second caseT2 we have that $|T_2|$=1 but in the formula we selected both possibilities 2 $|T_2|$ (so in the case we have two edges) and by 3 in the last case we have the same pattern used above so $|T_3|$=1 but in the formula we selected both possibilities 3 $|T_3|$.
Now starting from the algorithm provided by the text we 're able to say that it returns a value x that could be 1 if it's detected a triangle otherwise 0. Now we can say that the expected value E[x]= $\frac{\sum_{x=1}^{s} x_i}{s}$ where s is the number of all the outcome. From this point we can say also that in case there's a $|T_0|, |T_1|, |T_2|$ the following algorithm return always 0, instead for $|T_3|$ we can't say the same. We can define at this point that E[x]=$\frac{3|T_3|}{|T_1|+2|T_2|+3|T_3|}$ that means for each edge(a,b) which takes a vertex, from the set V-{a,b}, that detect exactly a triangle the algorithm add to $|T_3|$=$|T_3|+1$. The number 3 on the numerator is for the fact we consider as (a,b) all the possibilities edges of triples of vertices' triangle. So the expectation return all the outcomes possibilities over all the possibilities configurations value .Then according the equality shows in the first step we can say E[x]=$\frac{3|T_3|}{|E|(|V|-2)} = \frac{\sum_{x=1}^{s} x_i}{s}$ so if we estimate $|T_3|$ we can show as required that $|T_3| = \frac{|E|(|V|-2)\sum_{x=1}^{s} x_i}{3s}$
We know $E[X]$ so through Chernoff Bound, we can bound the probability that X deviates above its expected value $(X > (1+\epsilon)E[X])$ so we can bound:

$$Pr(X > (1+\delta)\epsilon) \leq e^{\frac{-\epsilon^2 E[X]}{3}},\ 0 \leq \delta \leq 1$$

$$Pr(\frac{1}{s}\sum_{i=1}^{s} x_i > (1+\epsilon)E[X]) \leq e^{\frac{-\epsilon^2 s E[X]}{3}}$$

If we set $s = \frac{1}{\epsilon^2}\frac{|T_1|+2|T_2|+3|T_3|}{|T_3|}log(\frac{1}{\delta})$ we have at the exponent in the right member:

$$exp(-\epsilon^2 \frac{3|T_3|}{3|T_1| + 2|T_2| + 3|T_3|}\frac{1}{\epsilon^2}\frac{|T_1| + 2|T_2| + 3|T_3|}{|T_3|}log(\frac{1}{\delta}))$$

so:

$$Pr(\frac{1}{s}\sum_{i=1}^{s} x_i > (1+\epsilon)E[X]) \leq e^{-log(\frac{1}{\delta})} = \delta$$

References

[1] Algorithm Design John- Kleinberg,Eva Tardos

[2] Algoritmi e strutture dati- C.Demetrescu, I.Finocchi ,G.P.Italiano