1 Exercise 1

1.1 (a) ILP and corresponding LP

The exercise can be related to a complete bipartite graph, where each node $f \in F$ is connected to every node $o \in O$, by the edge $w(f, o) \geq 0$. Knowing that, a matching is a subset of the edges for which every vertex belongs to exactly one of the edges, we want to find the solution for which the sum of the edges weight is minimum. Moreover, by definition we know that if $|F| > Im\{F\}$ (where, $Im\{F\} = |O|$), the problem has no solution, but for the assumptions we can preclude this case.

Now, to find the solution, we can model the problem as an Integer Linear Program:

$$
\begin{array}{ll}
Minimize & \sum_{e \in E} w(f,o) * x_e \\
\text{s.t.} & \forall o \in O \ \sum_{j \to o} x_{j \to o} \leq 1 \\
& \forall f \in F \ \sum_{f \leftarrow i} x_{f \leftarrow i} = 1 \\
& \forall e \in E \ x_e \in \{0, 1\}
\end{array}
$$

E denotes all the edges in the graph, and x_e is a (boolean) decision variable which becomes 1 if $f \leftrightarrow o$ match, 0 otherwise. The first constraint avoids to have more friends on the same outfit, the second one declares that each friend must receive the outfit (or it will just put everything to 0).

To relax a ILP to the corresponding LP, we need to remove the integrality constraint ($x_e \in \{0, 1\}$), and replace it with $0 \leq x_e \leq 1$. This process is commonly used to transform an NP-hard optimization problem into a related problem that is solvable in polynomial time.

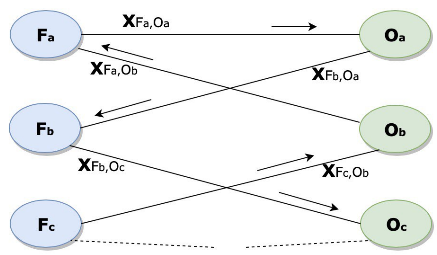
1.2 (b) Integer optimal

Unfortunately, the relaxed LP could introduce extra fractional solutions, which are not allowed in the ILP. Therefore, we want an approximation of the solution for LP which is also feasible for ILP.

Before running the LP solver, if $|F| < |O|$ we will add $|O| - |F|$ nodes to F, connected by edges with weight 0 to each node of O. So that, also the first constraint becomes $= 1$ (square matrix), and the optimum remain the same.

First at all, we know that $opt(LP) \leq opt(ILP)$ and a feasible solution for ILP is also feasible for LP (trivial). Now, let $\bar{x}_{fo}$ be an optimal fractional solution of the LP, and $frac(x)$ the number of non-integer variables in the solution. We will have:

- If $(frac(x) = 0) \Rightarrow$ All the solutions are integers, then LP solution = ILP solution
- $(frac(x) \neq 0) \Rightarrow$ Given that, the sum of adjacent variables to a vertex f_b is 1 (by the constraint). If we take a fractional variable $\bar{x}_{f_b o_a}$, there must be another variable $\bar{x}_{f_b o_c}$ adjacent to f_b, with a fractional value. Similarly, on o_c there must be another adjacent variable $\bar{x}_{f_d o_c}$, with a fractional value. We can iterate the process, and since the graph is finite, at some point we must have an even (bipartite property) cycle that ends back in the vertex o_a.



  Now, let's put every odd edge is in a set A, and every even edge in a set B. We can define the variable v as the smallest difference (inside the cycle) between an odd variable and 0, or an even variable and 1.

  At this point, we can subtract v from all the odd variables, and add v to all the even variables in the cycle. Therefore, without breaking any constraint, we will have a new integer variable, and decrease $frac(x)$. Moreover, the process does not influence the total sum of the cycle, then the new matching has the same final cost.

  Repeat until all the variables become integers.

If other friends f were added, remove the corresponding x_e from the final solution.

The algorithm will find the solution in at most $|O|$ steps.

2 Exercise 2

2.1 (a) Expectation

First at all, let's notice that the problem can easily be traced to the well known NP-complete unweighted Vertex-cover, where B' corresponds to the vertex cover given by the provided algorithm.

Now, given the optimal solution OPT for the vertex cover, we can consider the following event when the algorithm checks an edge $e(u, v)$ and adds u or v to B':

- X = added $vertex \in OPT$
- $\neg X$ = added $vertex \notin OPT$

Therefore, we can have the following situations:

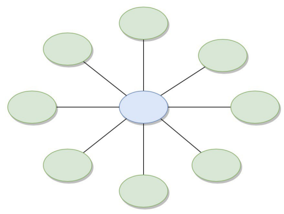
- $u, v \in B' \Rightarrow \; X = \neg X = 0$
- $u, v \notin B'$
  - $(u, v \in OPT) \;\Rightarrow\; P[X] = 1,\; P[\neg X] = 0 \;\Rightarrow\; X = 1,\; \neg X = 0$
  - $(u \notin OPT \vee v \notin OPT) \;\Rightarrow\; P[X] = P[\neg X] = 1/2$ (the vice versa implies that each edge must be covered, as in the vertex cover. Then, at least 1 vertex must be in OPT)

Therefore we always have that $P[\neg X] \leq P[X]$ and so $E[\neg X] \leq E[X]$. Now, considering:

- $|B'| = \sum X + \sum \neg X$
- $\sum X \leq |OPT|$

It is trivial that: $E[B'] = E[\sum X + \sum \neg X] \leq 2E[\sum X] \leq 2E[|OPT|]$

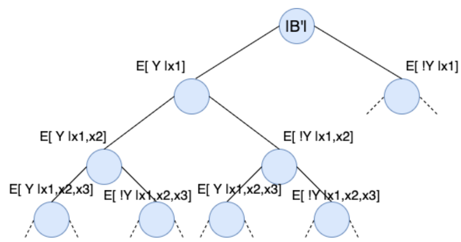
2.2 (b) Worst case



As proved, the algorithm performs pretty well in expectation, but unfortunately there are some cases where this randomized version could perform very bad. As we can see in the image, when we have a graph with a star topology, $|OPT| = 1$ (blue node), but the algorithm could repeatedly choose an external vertex (green nodes) and ends up with a $|B'| = n - 1$ solution (where n is the number of vertices in the graph). Therefore, we can even have $|B'| = (n - 1)|OPT|$.

Since we can have $|B'| = n - 1$, while $|OPT| = 1$, then $|B'| \geq c|OPT| \Rightarrow n - 1 \geq c$

2.3 (c) Conditional expectation



To use the method of conditional expectation, we first need to run the randomized algorithm. Then, we have to select, step by step, an edge of the B' solution.
At each iteration, we have to compute again the expectation, considering the already considered edges and the expectation of the edges we still have to check, which obviously gives bad performances.

3 Exercise 3

3.1 (a) $\alpha < 1$

Philip/P2	Rock	Paper	Scissors
Rock	$\alpha, 0$	$\alpha - 1, 1$	$\alpha + 1, -1$
Paper	$1, -1$	$0, 0$	$-1, 1$
Scissors	$-1, 1$	$1, -1$	$0, 0$

Suppose that $p_r,\ p_p,\ p_s$ are the probabilities for *Philip* to play respectively Rock, Paper and Scissors, and $p_s = 1 - p_r - p_p$. Similarly, for *Player 2*, we define $q_r,\ q_p,\ q_s$ and $q_s = 1 - q_r - q_p$.

Trivially, we have that Philip's mixed strategy is the the same of the classic game $(\frac{1}{3};\ \frac{1}{3};\ \frac{1}{3})$. For Player 2's MS instead, we have to consider α, therefore we have the following:

$$E(R) = \alpha \cdot q_r + (\alpha - 1) \cdot q_p + (\alpha + 1) \cdot (1 - q_r - q_p) = 1 + \alpha - q_r - 2q_p$$
$$E(P) = 1 \cdot q_r + 0 \cdot q_p + (-1) \cdot (1 - q_r - q_p) = 2q_r + q_p - 1$$
$$E(S) = (-1) \cdot q_r + 1 \cdot q_p + 0 \cdot (1 - q_r - q_p) = q_p - q_r$$

Now, let's set the condition $E(R) = E(S)$ and $E(P) = E(S)$

$$\begin{cases} 1 + \alpha - q_r - 2q_p = q_p - q_r \quad \Rightarrow \quad 1 + \alpha - 3q_p = 0 \quad \Rightarrow \quad q_p = \frac{(\alpha+1)}{3} \\ 2q_r + q_p - 1 = q_p - q_r \quad \Rightarrow \quad 3q_r - 1 = 0 \quad \Rightarrow \quad q_r = \frac{1}{3} \end{cases}$$

$$q_r = \frac{1}{3},\ q_p = \frac{(\alpha+1)}{3},\ q_s = 1 - q_r - q_p = \frac{(1-\alpha)}{3}$$

Finally, from the results above, we have $(\frac{1}{3};\ \frac{(\alpha+1)}{3};\ \frac{(1-\alpha)}{3})$ mixed strategy for Player 2.
In conclusion, having $E[R] = E[P] = E[S] = \frac{\alpha}{3}$, we can compute Philip's *expected payoff* as:

$$Payoff_{Philip} = \frac{1}{3} \cdot E[R] + \frac{1}{3} \cdot E[P] + \frac{1}{3} \cdot E[S] = \frac{\alpha}{3}$$

3.2 (b) $\alpha \geq 1$

For the second point of this exercise, we have to compute Philip's expected payoff for $\alpha \geq 1$. It require to split the problem in two subcases $1 \leq \alpha < 2$ and $\alpha \geq 2$, having different results.

3.2.1 (b.1) $1 \leq \alpha < 2$

First, we have to make some changes on the matrix, which loses the second row and the last column, since we have that:

- Philip: Rock strategy dominates Paper strategy
- Player 2: Paper strategy dominates Scrissors strategy

Philip/P2	Rock	Paper
Rock	$\alpha, 0$	$\alpha - 1, 1$
Scissors	$-1, 1$	$1, -1$

Indeed, does not make sense anymore to use those moves. Now, we can compute:

Mixed Nash equilibrium of both players

$(\frac{2}{3};\ 0;\ \frac{1}{3})$ and $(\frac{2-\alpha}{3};\ \frac{(\alpha+1)}{3};\ 0)$

Philip's payoff

$$Payoff_{Philip} = \frac{2\alpha-1}{3}$$

3.2.2 (b.2) $\alpha \geq 2$

Using the same considerations as before, here we also have:

- Philip: Rock strategy dominates Scissors strategy
- Player 2: Paper strategy dominates Rock strategy

Philip/P2	Paper
Rock	$\alpha - 1, 1$

Then, we can compute:

Mixed Nash equilibrium of both players

$(1;\ 0;\ 0)$ and $(0;\ 1;\ 0)$

Philip's payoff

$$Payoff_{Philip} = \alpha - 1$$

4 Exercise 4

In this exerxise we base our idea on the *Guessing cards* example showed during the lectures. First of all, we define n random variables X_i such that:

$$X_i \begin{cases} 1 & \text{if } i^{th} \text{ student is among the best} \\ 0 & otherwise \end{cases}$$

This variable becomes 1 with probability P_{worthy}. We know that a student is among the best if there is no other students who is, as good, or better, on all exercises. Therefore, we have the following results:

First of all, we define the probability that a students j is better than i:

$P(j\ better\ i) = P(a_i \leq a_j \cup b_i \leq b_j)$

Then, we can derive the probability that a student j is not better than i

$P(j\ not\ better\ i) = 1 - P(j\ better\ i) = 1 - P(a_i \leq a_j \cup b_i \leq b_j)$

Through the last one we can define the probability that a student is worthy

$P(i\ is\ worthy) = P(j\ not\ better\ i, \forall j \neq i) = \prod_{j\neq i}(1 - P(j\ better\ i)) = P_{worthy}$

Furthermore, we define $S = \sum_i X_i$, so that our purpose is to proof $E[S] = O(log\ n)$. Let's define the excepted value of X_i:

$$E[X_i] = 0 \cdot P(X_i = 0) + 1 \cdot (X_i = 1) = P(X_i = 1) = \frac{1}{n} \cdot P_{worthy}$$

Now, we have $E[S] = \sum_i E[X_i]$.
This is a crucial point in the proof, because the expected values are different, in fact, at every extraction, we decrease the number of students undrafted by 1. Following this property, we have that:

$$E[S] = \sum_i E[X_i] = (\frac{1}{n} \cdot P_{worthy}) + (\frac{1}{n-1} \cdot P_{worthy}) + ... + (1 \cdot P_{worthy}) = \sum_{i=n}^{1} \frac{1}{i} \cdot P_{worthy}$$

Now, we have $\sum_{i=n}^{1} \frac{1}{i} \to log\ n$, because this is like an Harmonic series:

$$E[S] = \sum_{i=n}^{1} \frac{1}{i} \cdot P_{worthy} = log(n) \cdot P_{worthy}$$

The resulting expected value makes sense if it is "close" to its expectations with reasonably *high probability.* We demonstrate this trough *Chernoff Bounds*, that allow us to bound the probability that S exceeds its expected value:

$$Pr(S > (1+\delta)\mu) \leq e^{\frac{\delta^2 \mu}{3}} \qquad \text{where } 0 \leq \delta \leq 1 \text{ and } \mu = E[S] = log(n)$$

Setting $\delta = \frac{1}{2}$ we have:

$$Pr(S > \frac{3}{2}log(n)) \leq e^{\frac{log(n)}{12}} = \frac{1}{\sqrt[12]{n}}$$

This last result tells us that the probability that the worthies students are more than $log(n)$, asymptotically, goes to 0 faster as n goes to ∞.

5 Exercise 5

5.1 (a) $|T_1| + 2|T_2| + 3|T_3| = |E|(|V| - 2)$

First, let's suppose we have a graph where each vertex has only a single edge connected to another vertex (as for T_1), then T_2 and T_3 are empty. Moreover, the number of couples in the graph are exactly equal to the number of edges $|E|$. Therefore, considering that each couple can be associated to each other node in the graph, except those 2 forming the couple, we have $|T_1| = |E|(|V| - 2)$.

Now, let's consider a graph having also triples in T_2. T_1 is still valid for each triple where the 3^{rd} vertex is not the other vertex connected to the triple. Then, we have a similar situation as before, but where T_1 loses at least 1 for each triple, depending on the structure of the graph. However, each decreasing in T_1 implies increasing in T_2. Therefore, since each triple in T_2 has 2 edges, it has to be considered 2 times, so we can multiply it's cardinality for 2 and find $|T_1| + 2|T_2| = |E|(|V| - 2)$

The same considerations can be used for graphs having triples in T_3, where each triple has 3 edges. Then, $|T_1| + 2|T_2| + 3|T_3| = |E|(|V| - 2)$

5.2 (b) estimator for T_3

The algorithm returns $x = 1$ only if the considered triple of vertices forms a triangle, which is equivalent to say that it is contained in T_3, then $E[x] = \frac{3|T_3|}{|T_1|+2|T_2|+3|T_3|} = \frac{3|T_3|}{|E|(|V|-2)}$ (as proved).

So we have $|T_3| = \frac{E[x]*|T_1|+2|T_2|+3|T_3|}{3} = \frac{E[x]*|E|(|V|-2)}{3}$.

Therefore, considering that each execution of the algorithm is independent, and the value of a random variable ought to be "near" its expectation with reasonably high probability, we can use the estimator: $|T_3| = \frac{1}{s}\sum_{i=1}^{s} x_i \frac{|E|(|V|-2)}{3}$.

5.3 (c) value s

The value s can be simply found by using the *Chernoff's Bounds*:

$Pr\left[\frac{1}{s}\sum_{i=1}^{s}\beta_i \geq (1+\epsilon)E[\beta]\right] < e^{-\epsilon^2 E[\beta]s/3}$ $Pr\left[\frac{1}{s}\sum_{i=1}^{s}\beta_i \leq (1-\epsilon)E[\beta]\right] < e^{-\epsilon^2 E[\beta]s/2}$

Which gives us:

$s \geq \frac{3}{\epsilon^2}\frac{|T_1|+2|T_2|+3|T_3|}{T_3} ln(\frac{2}{\delta})$