HOMEWORK 2 Students VALERIO VALENTE 1894954 - GABRIELE DE STRADIS 1947850

Exercise 1

1.a

This is the ILP definition for the problem of helping Santa planning his elves deliveries.

We start by defining a new set D = $\bigcup_{e \in E} H_e$ which represents all the possible deliveries for the n houses.

Let us first define a cost function C: d → $\mathbb{R}^+$, ∀ d ∈ D. We represented the integer problem (ILP) as follows:

$$\begin{aligned} \min \ & \textstyle\sum_{d \in D} c(d) * x(d) \\ \text{s.t.} \ & \textstyle\sum_{d \in D} x(d) \geq n \\ & x(d) \in \{0,1\} \end{aligned}$$

The x function represents whether a specific delivery is performed or not. Hence, the first constraint represents the fact that we have to cover the n deliveries for the n houses.

The corresponding LP-relaxation of the problem is the following:

$$\begin{aligned} \min \ & \textstyle\sum_{d \in D} c(d) * x(d) \\ \text{s.t.} \ & \textstyle\sum_{d \in D} x(d) \geq n \\ & x(d) \in [0,1] \end{aligned}$$

1.b

We want to prove the following relation: ALG = DUAL^{unf} = f * DUAL^{feas} ≤ f * OPT^{LP} ≤ f * OPT.

Therefore, we constructed the dual of our problem which is:

$$\begin{aligned} \max \ & n \textstyle\sum_{d \in D} y(d) \\ \text{s.t.} \ & y(d) \leq c(d) \quad \forall d \in D \\ & y(d) \geq 0 \end{aligned}$$

First of all, we developed a greedy algorithm for our problem which maintains a set C_i initially equal to H that at every iteration (i.e. ∀ d ∈ D) will be reduced by removing the house for which a delivery d is chosen (with C_i we indicate the set C, at iteration i).

We choose such d by considering the minimum returned value of the formula $\alpha = \frac{|F_e|+|U_e|+c(d)}{|C_i|}$, which will consider the average cost for each possible delivery to the house of the current iteration. The set F_e for the elf e ∈ E contains the deliveries already given to e, while the set U_e = H_e/{houses already assigned}.

We proceeded by also assigning such value α to a variable price(d) and, by adding d to the set T containing all the optimum deliveries. Finally, we remove h from C_i and repeat until C is empty.

So, at the moment we have that ALG = DUAL^{unf} = $n \sum_{d \in D} price(d)$.

Proof

Let's consider a possible approximation for y(d) = price(d), which will be of type $y'(d) = r * price(d)$. In particular, we want to prove that $y'(d) = \frac{price(d)}{H_n}$ is dual feasible. We know that at iteration k, given an element h_j of {h_1, h_2, …, h_j, …, h_n}, we have $|C_k| = k - j + 1$, hence d_k can cover house h_j by paying $price(d_k) \leq \frac{c(d_k)}{n-j+1}$.

We can intersect the two relations:

$$\begin{cases} y'(d_k) = \frac{price(d_k)}{H_n} \\ price(d_k) \leq \frac{c(d_k)}{k-j+1} \end{cases}$$

and then we get $y'(d_k) \leq \frac{1}{H_n} \frac{c(d_k)}{k-j+1}$

Summing up the various $y'(d_k)$ will give us the feasible dual solution, hence we can write:

$n \sum_{i=1 \ to \ k} y'(d_i) \leq \frac{c(T)}{H_n} \left(\frac{1}{i} + \frac{1}{i-1} + \cdots + 1\right) = c(T)\frac{H_k}{H_n} \leq c(T)$

HOMEWORK 2 Students VALERIO VALENTE 1894954 - GABRIELE DE STRADIS 1947850

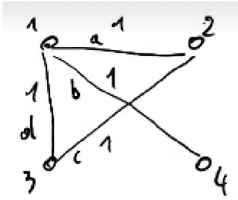
Exercise 2

Our goal is to prove Santa that the problem is NP-Complete. Therefore, we proceed with a Karp reduction to a known NP-Complete problem that will show that unless P=NP it doesn't exist a Poly-time solution to this problem. Let us consider the Scheduling With Release Times Problem, our aim is to prove that Scheduling-Release-Times $\leq_p$ Sleigh-Problem. In order to make this reduction, we start by considering a single-job instance for the Scheduling Problem. This complains a single job j_1 having release time $r_1 = 0$, processing time $t_1 = 5$ and deadline $d_1 = 7$. Even with a single job, we can observe two constraints: $d_i – t_i \leq r_i$ and $ti < d_i$. If we project this single-job instance to our Sleigh Problem, we can notice that we also have a similar situation with at least two constraints: having at least one green piece and a red one for each sleigh. Therefore, a solution for the Scheduling Problem will also constitute a solution for our problem. By extending the instance of the Scheduling problem with more than one job, another constraint comes into play. In fact, we should consider that a solution is feasible if and only if the release time r_i and processing t_j are compatible for i<j. At the same, for our problem we must ensure that no two slides are equal. We have yet showed that every constraint of the Scheduling Problem can be associated with a constraint of our Sleigh Problem. Moreover, if we consider that Santa has a long list of additional constraints, we can think about our problem as a more complex version of what showed above. Hence, we have finally proved the polynomial-time reduction and so that the Sleigh Problem is NP-complete.

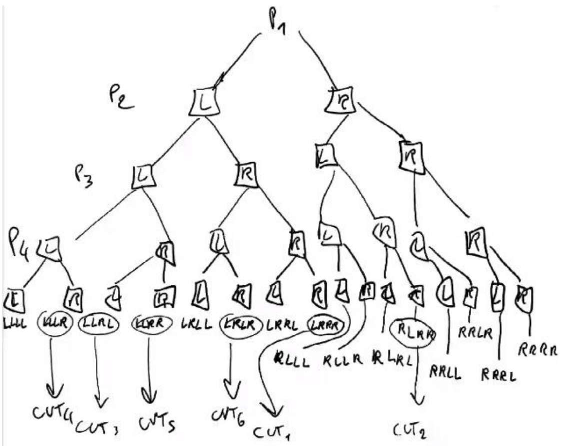
HOMEWORK 2 Students VALERIO VALENTE 1894954 - GABRIELE DE STRADIS 1947850

Exercise 3

We have to design a game for which the value of PriceOfAnarchy is equal to 2. First of all, we have to think about some properties of a graph containing as many vertices as the number of the players. For example, we can start from the following graph G:

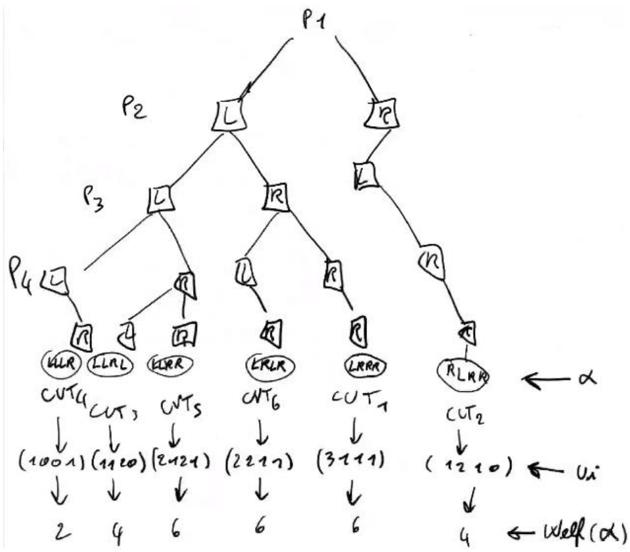


Here, we should notice that only certain cut can be made, allowing only certain solution to our problem, since every player will be labelled L or R depending on the cut made.
Possible cuts for our instance are: {1,234}, {2,134}, {3,124}, {4, 123}, {12,34}, {13,24}.
To represent the possibility described above, we can construct a tree with all the possible combinations of α, given a final representation of both the feasible and unfeasible cut that can be made.



By choosing only the feasible one, we subsequently compute a value given by the following function:

$$Welf(\alpha) = \sum_{i\,\epsilon\,|V|} u_i(\alpha)$$



Let us assume that our game is to be played in sequence by the players. With this assumption we can then assume that each player P_i will be labelled L or R by his dominant strategy, given by his payoff u_i. Since we consider the dominant strategy, the sequence of the player will also determine the final states that will represent a Nash Equilibrium and also for these states we then compute the Welf(α). Since $\text{PriceOfAnarchy} = \frac{\max Welf(\alpha)\forall\alpha\epsilon S}{\max Welf(\alpha)\forall\alpha\epsilon NashEQ}$, the game G solution of this problem will be the one with a specific sequence of players that will come up with a value 2 for the PriceOfAnarchy formula.

Exercise 4

4.2

Our algorithm works as follows. It maintains a set C, initially empty, that will contain $|U| = k$ cities. At the first iteration, it picks the biggest edge $e = (u,v)$ with $u,v \in S$ and put either u or v into C. Then it repeatedly chooses the furthest city from the latter added to the set until k cities are into $C = \{s_1, \ldots, s_k\}$, with s_i closest city to optimal antenna a_i. This represents a situation in which each city belongs to one of the k optimal antennae. We'd like to prove that this constitutes a 3-approximation.

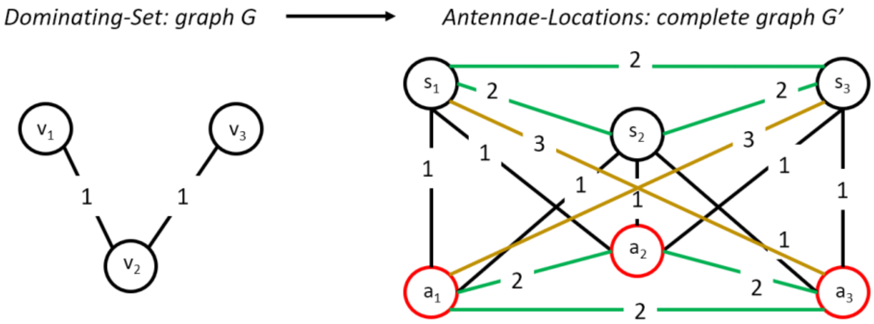
If we consider a random city $c \notin C$ and its closest optimal antenna $a_i \in U$, then we will have that thanks to the triangle inequality, the following property holds: $d(c, a_i) \leq d(c, s_i) + d(s_i, a_i) \leq 2OPT + OPT = 3OPT$. Therefore, we assume that no city c is at a distance $d > 2OPT$ from one of C. By contradiction, if we assume that it exists a city such that its distance is greater than 2OPT, this would represent the situation in which we need $|C| = k+1$ cities, and hence k+1 antennae, but this contradicts the problem specifications asking us a k-size solution.

4.3

In order to prove that finding α-approximation with $\alpha < 3$ is also NP-hard, we need to take advantage of the polynomial reduction.

We want to prove that Dominating-Set $\leq_p$ Antennae-Locations. We then start by taking an arbitrary instance of the Dominating-Set problem and transform it into an instance of our problem.

Let's consider a graph G=(V,E), instance of the Dominating Set Problem. We'd like to convert it into an instance of our problem (Karp Reduction). Since we have 2 sets of nodes, we split each node $v \in V$ into 2 nodes: one $a_v \in A$ and one $s_v \in S$. We subsequently connect every split node to its "copy" with an edge such that it will have weight = 1. Every edge of the type $e=(s_i,s_j)$ $s_i,s_j \in S$, and $e=(a_i,a_j)$ $a_i,a_j \in A$, will have weight = 2. In order to get a complete graph, we add all the needed edges. These will have weight = 3 if e=(a,s), weight = 2 otherwise and, with this constraints, the Triangle Inequality still holds.



For the instance yet created, we have that finding a solution for the Antennae-Locations problem for a subset $U \subset A$, with $|U| = k$ will correspond to a possible solution of either 1 or 3. Therefore, this 3 constitutes an upper bound to the instance yet obtained and, hence, also for the dominating set graph G. But then, since we know that the best solution for G is the vertex in the middle (OPT=1), we come up with a solution 3-approximated with regard to the optimum. This proved that finding an α-approximation for the Antennae-Locations problem, with $\alpha = (3 - \varepsilon)$ is NP-hard to find.