Algorithm Design 2020/2021

Exercise Sheet 2

Sultan Umarbaev, Matricola: 1954544

Deadline: 10/01/2021

1 Exercise 1

Define a set of homes H with size n, set of elves E with size $m < n$, H_e set of doable houses $\forall e \in E$ and delivery time 1 hour.

1.1 Part A

Using set of doable houses H_e $\forall e \in E$ and $T_e = \sum_{h \in H_e} t_h$ ($t_h = 1$ due to 1 hour delivery) to denote the resulting time, we seek to minimize $\max_e T_e$ and model the problem as an ILP. The first constraint ensures there are no double assignments. The second says that assignment of houses to elves has time at most T which is the maximum time of the assignment.

$$\begin{aligned} \min \quad & T \\ \text{s.t.} \quad & \sum\nolimits_e x_{eh} = t_h = 1, && \forall h \in H \\ & \sum\nolimits_h x_{eh} \leq T, && \forall e \in E \\ & x_{eh} \in \{0, 1\}, && \forall e \in E \text{ and } \forall h \in H_e \\ & x_{eh} = 0, && \forall e \in E \text{ and } \forall h \notin H_e \end{aligned} \tag{1}$$

To obtain LP, we relax the constraint $x_{eh} \in \{0, 1\}$ to $x_{eh} \geq 0$.

1.2 Part B

First, compute an optimal solution x^* to LP in polynomial time using any standard linear programming algorithm. Assign houses to elves independently with the probability x^*_{eh}.
Let X_{eh} be a 0/1-valued random variable such that $X_{eh} = 1$ indicates that house h is assigned to elf e and $X_{eh} = 0$ otherwise. Therefore the number of houses or time assigned to the busiest elf e is $\sum_h X_{eh}$. The expectation of X_{eh} is equal to the probability that $X_{eh} = 1$, thus the expected time assigned to the busiest elf e is.

$$E[T_e] = \sum\nolimits_h P[X_{eh} = 1] = \sum\nolimits_h x^*_{eh} = OPT \tag{2}$$

In particular, it is bounded by the optimal solution.
If we select a random elf e, the probability that house h is assigned to him is x^*_{eh}. Thus, the probability that h is not assigned to any elf is $\prod\limits_{e \in E}(1 - x^*_{eh})$. From which derives that the house h is assigned to any elf e with probability $1 - \prod\limits_{e \in E}(1 - x^*_{eh})$. Since $(1 - x^*_{eh}) \leq \exp(-x^*_{eh})$ and by LP constraint $\sum_e x_{eh} = 1, \forall h \in H$:

$$P[h \text{ is assigned to any } e] = 1 - \prod_{e \in E}(1 - x^*_{eh}) \geq 1 - \prod_{e \in E} \exp(-x^*_{eh}) = 1 - \exp(-\sum\nolimits_e x^*_{eh}) \geq 1 - 1/\exp \tag{3}$$

Hence, for every house $h \in H$, the probability that h is assigned to any elf e is at least $1 - 1/\exp$.

1.3 Part C

2 Exercise 2

Given a set P of n different parts and a formal depiction of Santa's constraints, output 187 different sets of parts.

2.1 Reduction

Current problem can be reduced from **Clique problem** (CP). The reduction takes an arbitrary CP instance as input, and transforms it to instance of the current problem.
Given integer k number of parts and an undirected graph $G = (V, E)$ where V is P and an edge $(u, v) \in E$ such that different models of the same part are not connected, for example different models of a front light $(p_1, ..., p_l)$. Therefore, each model of the one part is connected only with models of other parts.

If there is a clique of size k in the graph G, it implies that this subset of vertices forms a combination of k parts for sleigh.
Consider an instance graph G in Figure 1. Suppose G has a clique X of size $k = 4$. Since no two models of the same part are connected in G, no two or more vertices in X are from one category but rather every vertex in X is a model of different part. Edges in X form a combination of these models and this unique combination of models define unique sleigh.
If there is a sleigh with k parts, it implies that a subgraph of these k parts construct a *k-clique* in the graph G.
Given a set P with elements $(p1, ..., p9)$ where:

- *p1* and *p2* are models of *part 1*
- *p3*, *p4* $->$ *part 2*
- *p5*, *p6* $->$ *part 3*
- *p7*, *p8*, *p9* $->$ *part 4*

Suppose there is a sleigh S with $k = 4$ parts, having a combination of models *p1*, *p3*, *p5*, *p8*. Build an undirected graph $G = (V, E)$ with $V = P$ and edges $(u, v) \in E$ only if vertices u and v are not of a same part, so vertices *p1* and *p2* do not have an edge, Figure 1. Consider subgraph X in G with vertices *p1*, *p3*, *p5*, *p8*. X is a *complete subgraph* since for every $u, v \in X$, (u, v) is an edge.

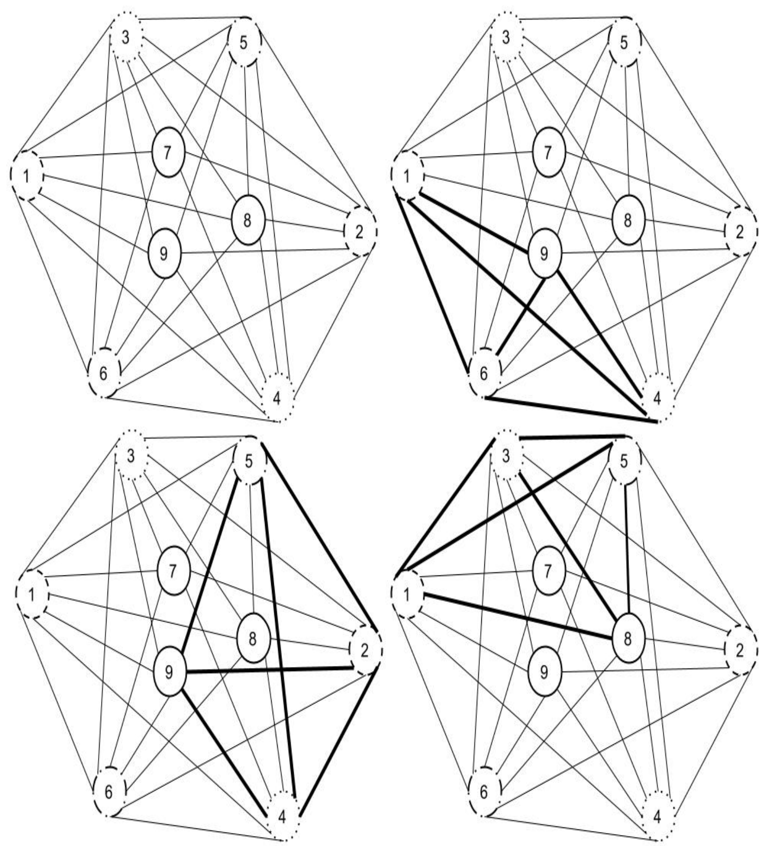

Figure 1: Graph instances.

Therefore, it is computationally hard to build even 1 sleigh.

3 Exercise 3

Given an undirected graph $G = (V, E)$ with $|V|$ players, where player i controls $v_i \in V$ and $e \in E$ with nonnegative weight w_e. Each player selects the set, *LEFT* or *RIGHT*, in which his controlled vertex will be, maximizing his payoff. The strategy space of each player i is $\alpha_i = \{LEFT, RIGHT\}$ and let $\alpha = (\alpha_1, ..., \alpha_n)$ be the corresponding strategy profile. With $CUT(\alpha)$ and N_i being a set of edges that have one endpoint in each set and the set of neighbours of v_i in the graph G, the payoff of player i is defined as $u_i(\alpha) = \sum_{e \in CUT(\alpha) \cap N_i} w_e$,

3.1 Part 1

Consider a graph G with 4 vertices and unit weights. One of the possible configurations for the game in which the *Price of Anarchy* equals 2 is shown in the Figure 2, where edges are (v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_1). In order to compute the *Price of Anarchy* of game we need to calculate maximum **social utility** in any state and minimum **social utility** at the equilibrium. There are in total 6 strategy equilibria in this graph configuration, 4 of which has minimum *social utility* which equals 4:

- $\alpha = (RIGHT, RIGHT, LEFT, LEFT)$
- $\alpha = (LEFT, LEFT, RIGHT, RIGHT)$
- $\alpha = (RIGHT, LEFT, RIGHT, LEFT)$
- $\alpha = (LEFT, RIGHT, LEFT, RIGHT)$

And 2 strategies where maximum *social utility* is 8:

- $\alpha = (RIGHT, LEFT, LEFT, RIGHT)$
- $\alpha = (LEFT, RIGHT, RIGHT, LEFT)$

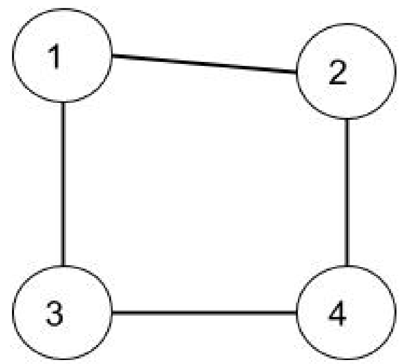

Figure 2: Game instance.

Thus the *Price of Anarchy* in this game is equal to $8/4 = 2$.

3.2 Part 2

4 Exercise 4

Given a complete graph $G = (V, E)$ with $V = A \cup S$ (A set of antennae, S set of cities), weight of each $(x, y) \in E$ is distance $d : V^2 \rightarrow R$, satisfying the triangle inequality. Define $U \subset A$ with size k, $d(x, y)$ as a distance between vertices x and y, $\min_{y \in U} d(x, y)$ as distance from x to closest y-antenna. Find U with size k, that minimizes $\max_{x \in S} \min_{y \in U} d(x, y)$.

4.1 Part 2

The idea is to apply *Center Selection algorithm*, shown in class, to the set S of cities. This results in a set $S' \subseteq S$ of k cities. Then $\forall s_i \in S'$ select the closest antenna $a_i \in A$ such that $d(s_i, a_i) \leq OPT$ with OPT being the optimal. As a result, we get $A' \subseteq A$ with size of k, which is U that needs to be find.
This results in a 3-approximation. Consider any city $x \in S$ and if there is a city $s_i \in S'$ at a distance at most $2OPT$ from x (refer to *Center Selection problem* in class), then by the triangle inequality $d(x, a_i) \leq d(x, s_i) + d(s_i, a_i) \leq 2OPT + OPT = 3OPT$, hence $d(x, a_i) \leq 3OPT$.
Assume that there is no $s_i \in S'$ at a distance at most $2OPT$ from x. Then by construction of the algorithm, x is included to *S'*, because for any $s_i, s_j \in S'$, $d(s_i, s_j) > 2OPT$. As a result the set $S' \cup \{x\}$ consists of $k + 1$ cities which is impossible since in that case at least $k + 1$ antennae are required for a solution with OPT.

4.2 Part 3

In order to show that finding α-approximation with $\alpha < 3$ is NP-hard, we reduce from the *Dominating Set problem* (DS) like in case with *Center Selection algorithm* shown in class. Given an instance of DS problem $G = (V, E)$, define an instance of the current problem G' such that $\forall v \in V$ there is a city s_v and antenna a_v. And for any $v, w \in V$, the distances:

- $d(s_v, s_w) = d(a_v, a_w) = 2$
- $d(s_v, a_w) = 1$ if $(v, w) \in E$
- $d(s_v, a_w) = 3$ if $(v, w) \notin E$

Note that *G'* satisfies the triangle inequality. Assume that there a dominating set D of size k. The solution for the instance of the current problem are the k antennae that correspond with the vertices in D. Thus, the value of the solution is 1. Now, assume that there is a solution of value 1 for the instance of the current problem. The set of k antennae has a corresponding dominating set of vertices D which is a solution to the DS problem instance. Therefore, G has dominating set of size k iff there exists k antennae set U with $OPT = 1$.
Note that any solution for this instance of the current problem has value either 1 or 3, this gives us the lower bound of 3 on the approximation ratio. Assume we have α-approximation algorithm with $\alpha < 3$, it can be used to solve the DS problem. Assume an instance G of DS problem, where we want to find if there exists dominating set with at most k vertices. Reduce it to the instance of the current problem as shown above and apply α-approximation algorithm.

- If such dominating set exists, then OPT of the instance of the current problem is 1 and the answer given by the algorithm will be < 3.
- If there is no dominating set of value at most k, then OPT of the instance of the current problem is 3 and the answer given by the algorithm will be at least 3.

Therefore, by looking at the output of the algorithm, we can define if there is a dominating set of size at most k. However, there is no polynomial time algorithm that solves DS problem, unless $P = NP$. Therefore, no α-approximation algorithm with $\alpha < 3$ exists, assuming that $P \neq NP$.