[P5] [*Linear Programming Relaxations for weighted vertex cover*] *Integer linear programs* (ILP) are mathematical optimization problems of the form

$$\begin{aligned} \mathrm{opt}_{\mathrm{ILP}} := \underset{\mathbf{x}=(x_1,\ldots,x_n)^T\in\mathbb{R}^n}{\text{minimize}} \quad & \langle \mathbf{c}, \mathbf{x}\rangle = \sum_{i=1}^n c_i x_i \\ \text{subject to} \quad & \langle \mathbf{a}_j, x\rangle \geq b_j \quad j = 1,\ldots,m, \\ & x_i \in \{0,1\} \end{aligned} \tag{1}$$

with $\mathbf{c}, \mathbf{a}_1, \ldots, \mathbf{a}_m \in \mathbb{R}^n$ and $b_1, \ldots, b_m \in \mathbb{R}$. Solving an ILP is NP-hard in general (see point (5) below), but replacing the integrality constraints $x_i \in \{0,1\}$ in (1) with inequalities $0 \leq x_i \leq 1$ results in a *linear program* that can be solved in polynomial time (in n and m). Doing so, is called a *relaxation*.

(1) Show that the optimal solution of any ILP is always lower-bounded by the minimum of the relaxed LP, i.e. $\mathrm{opt}_{\mathrm{LP}} \leq \mathrm{opt}_{\mathrm{ILP}}$. (1 P.)

(2) For an undirected graph (V, E), *minimal vertex cover* is the task to find a subset of vertices $S \subseteq V$ of minimal size that "touch" all the graph's edges, i.e. for every $(u, v) \in E$, either u, or v (or both) belong to S. A generalization thereof is *weighted vertex cover*: in this variant each vertex $v \in V$ is assigned a cost $c_v \geq 0$. The task is now to find a vertex cover S that admits minimal cost $\mathrm{cost}(S) := \sum_{v\in S} c_v$. Show that solving weighted vertex cover can be formulated as an ILP. Relate the ILP *minimizer* $\mathbf{x}^\sharp_{\mathrm{ILP}} \in \mathbb{R}^m$ – i.e. the vector which achieves $\mathrm{opt}_{\mathrm{ILP}}$ – to the corresponding vertex cover. (2 P.)

(3) Relax the weighted vertex cover ILP to an LP. Argue that, unlike $\mathbf{x}^\sharp_{\mathrm{ILP}}$, the LP minimizer $\mathbf{x}^\sharp_{\mathrm{LP}}$ in general doesn't admit a direct interpretation in terms of a vertex cover. Devise a "rounding procedure" to overcome this lack of interpretability: replace $\mathbf{x}^\sharp_{\mathrm{LP}}$ by a binary vector $\mathbf{x}^*$ ($x_i^* \in \{0,1\}$). Prove that such a $\mathbf{x}^*$ always describes a valid vertex cover $S^* \subseteq V$ of the graph (V, E) and the cost it affords obeys

$$\mathrm{cost}(S_{\mathrm{opt}}) \leq \mathrm{cost}(S^*) \leq 2\mathrm{cost}(S_{\mathrm{opt}}),$$

where $\mathrm{cost}(S_{\mathrm{opt}})$ denotes the cost of the optimal vertex cover. (2 P.)

(4) Numerical solvers for LPs are available for many mathematical programming languages (e.g. Mathematica). Use one of them to test the performance of your LP relaxation to weighted vertex cover numerically. To this end, consider star-shaped graphs as depicted in Figure 1 as a benchmark. Analyze the performance of your LP relaxation for different numbers N of surrounding vertices in the three interesting parameter ranges for the central cost V: (i) $V \ll N$, (ii) $V \simeq N$, (iii) $V \gg N$. (2 P.)

(5) *(optional)* Prove that the original vertex cover problem (with a unit cost associated to each vertex) is NP-hard by reduction from 3-SAT.
Hint: Similar to the reduction from 3NAE-SAT to MAX-CUT, represent each pair of literals $(x_i, \bar{x}_i)$ by two vertices that are connected by an edge. Then, represent each 3SAT-clause by a triangle connecting the appropriate triple of vertices. Finally, show that determining whether there exists a vertex cover of the resulting graph that has "cost" $n + 2m$ (where n is the number of literals, and m is the number of clauses in the 3SAT formula), would allow one to check satisfiability of the original 3SAT formula. (+3 P.)

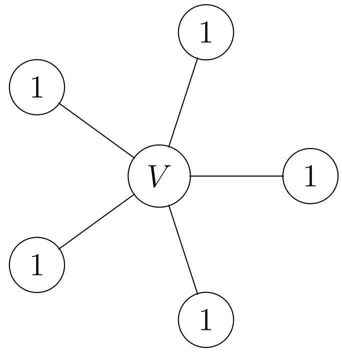

Figure 1: Example of a star-shaped graph with $N = 5$ vertices surrounding the center. The central vertex has associated cost $V \geq 1$, while exhibit unit cost.

[P5] [*NP-hardness of Graph isomorphism implies collaps of polynomial hierarchy*]
TBA (sorry about the delay) (3 P.)