The goal of this exercise session is to study two important aspects of approximation algorithms: first, fully polynomial randomized approximation schemes, and second gap-preserving reductions as a tool to obtain inapproximability results.

1 Counting the solutions to a DNF formula

In this exercise, we want to approximate the number of solutions to a Boolean formula in DNF (disjunctive normal form): Let \( f = C_1 \lor \cdots \lor C_m \) be a DNF Boolean formula (an \( \lor \) of \( \land \)-clauses as opposed to conjunctive normal form formulas, used in SAT, which considers an \( \land \) of \( \lor \)-clauses) on \( n \) variables \( x_1, \ldots, x_n \). Each clause \( C_j \) is of the form \( C_j = l_1 \land \cdots \land l_{k_j} \), where each \( l_i \) is a literal (i.e., a variable or its negation) and \( k_j \) denotes the number of literals in \( C_j \). We assume that every clause is satisfiable and non-redundant (i.e., contains each variable at most once, negated or not). The goal is to compute \( \# f \), the number of truth assignments of the variables \( (x_i) \) that satisfy \( f \).

We want to evaluate \( \# f \) by sampling randomly the \( 2^n \) possible truth assignments of variables \( (x_i) \).

1a. We first consider uniform Monte-Carlo sampling: Draw a truth assignment \( \tau \) uniformly at random in \( \{0, 1\}^n \), and set \( X = 2^n \) if \( \tau \) satisfies \( f \) and \( X = 0 \) otherwise.

Show that \( \mathbb{E}[X] = \# f \). How many random bits does every draw of \( X \) use?

1b. However, \( X \) fails to estimate \( \# f \) in polynomial time. Indeed, even if \( \# f > 0 \), the probability for \( X \) to be non-zero can be exponentially small. Thus, a polynomial number of draws of \( X \) is not enough to estimate \( \# f \) within any constant factor. Indeed, assume for the next two questions only that \( n \) is even and that \( f \) consists of the single clause: \( f = x_1 \land x_2 \land \cdots \land x_n/2 \).

1b\text{a}) What is the value of \( \# f \)? What is the probability for \( X \) to be non-zero for this \( f \)?

1b\text{b}) Let \( X_1, \ldots, X_k \) be the values of \( k \) independent draws of \( X \) for this \( f \). What is the probability that at least one of the draws is non-zero? Show that if \( k \) is polynomial in \( n \), this probability is \( o(1) \), i.e., the probability goes to 0 when \( n \) goes to \( \infty \).
1c. Let us now go back to the general case. Since uniform sampling does not work, we
decide to use a biased variable which samples only satisfying assignments. Let $T_j$ denote
the set of truth assignments that satisfy clause $C_j$ (which has $k_j$ literals). Remark that
$\# = \#(\bigcup_j T_j)$, our goal is thus to count the number of elements in the union of the $T_j$.

1ca) What is the size of $T_j$?

1cb) Let $c_\tau$ denote the number of clauses satisfied by truth assignment $\tau$. Express
$M := \sum c_\tau$ in terms of the $\#T_j$'s.

1cγ) Design a (polynomial-time, in $n$ and $m$) randomized algorithm that draws a truth
assignment $\tau$ with probability $c_\tau / M$, i.e. with probability proportional to $c_\tau$.
Explain in details your random sampling procedure. How many random bits
does your algorithm use (as a $O(\cdot)$ of $n$ and $m$)?

1cδ) Consider the random variable $Y$ defined as follows: draw a truth assignment $\tau$
with probability $c_\tau / M$, and set $Y = M / c_\tau$.
Show that $E[Y] = \#$.

1cε) Let $\sigma^2(Z) = E[(Z - E[Z])^2]$ denote the variance of a random variable $Z$.
Show that $\sigma^2(Y) \leq ((m-1)E[Y])^2$, where $m$ is the number of clauses in $f$.

1cζ) Let $Z_1$ and $Z_2$ be two independent random non-negative variables and $Z = Z_1 + Z_2$. Show that $E[Z_1Z_2] = E[Z_1]E[Z_2]$ and that $\sigma^2(Z) = \sigma^2(Z_1) + \sigma^2(Z_2)$.

1cη) Recall Chebyshev's inequality which claims that for all random variable $Z$ and
all $a \in \mathbb{R}_+$,
\[ \Pr\{|Z - E[Z]| \geq a\} \leq \frac{\sigma^2(Z)}{a^2}. \]
Let $Y_1, \ldots, Y_k$ be the values of $k$ independent draws of $Y$, and set $Z = \frac{Y_1 + \cdots + Y_k}{k}$.
Show that for all $\varepsilon > 0$, and all $k \geq 4(m-1)^2/\varepsilon^2$,
\[ \Pr\{|Z - \#| < \varepsilon \cdot \#\} \geq 3/4. \]

1d) Write down a polynomial time (in $n$, $m$ and $1/\varepsilon$) randomized algorithm which
outputs, for all $\varepsilon > 0$, a value $\theta$ such that $(1 - \varepsilon) \# \leq \theta \leq (1 + \varepsilon) \#$ with constant
probability $> \frac{1}{2}$ (independent of $n$, $m$ and $\varepsilon$).

1d) Can we use this algorithm to solve SAT? Justify your answer. Can we obtain from it a
PTAS for Max-SAT? Justify your answer.

2 Gap preserving reductions: Vertex-Cover($d$) and Steiner-Tree
We admit the following hardness of approximation result for Max-3SAT($k$), the restriction
of Max-3SAT to instances where each variable occurs at most in $k$ clauses:
Theorem 1. There exist an absolute constant $\epsilon_0 > 0$ and a polynomial-time gap-introducing reduction from SAT to Max-3SAT(29) that transforms a Boolean formula $\varphi$ to a Boolean formula $\psi$ such that:

- if $\varphi$ is satisfiable, then $\text{OPT}_{\text{SAT}}(\psi) = m$, and
- if $\varphi$ is not satisfiable, then $\text{OPT}_{\text{SAT}}(\psi) < (1 - \epsilon_0)m$,

where $m$ is the number of clauses in $\psi$.

2a. Let Vertex-Cover($d$) be the restriction of unweighted Vertex Cover to instances in which each vertex has degree at most $d$. We will prove the following result:

Theorem 2. There exists a polynomial-time gap preserving reduction from Max-3SAT(29) to Vertex-Cover(30) that transforms a Boolean formula $\psi$ to a graph $G = (V, E)$ such that:

- if $\text{OPT}_{\text{SAT}}(\psi) = m$, then $\text{OPT}_{\text{VC}}(G) \leq \frac{2}{3}|V|$, and
- if $\text{OPT}_{\text{SAT}}(\psi) < (1 - \epsilon_0)m$, then $\text{OPT}_{\text{VC}}(G) > (1 + \epsilon_{VC})\frac{2}{3}|V|$,

where $m$ is the number of clauses in $\psi$, $\epsilon_0$ is as in Theorem 1, and $\epsilon_{VC} = \epsilon_0 / 2$.

Assume w.l.o.g. that each clause of $\psi$ has exactly 3 literals (this can be easily accomplished by repeating the literals within a clause, if necessary). We will use the following standard transformation: $G$ has 3 vertices for each clause of $\psi$; each of these vertices is labeled with one literal of the clause (thus, $|V| = 3m$); $G$ has two types of edges:

- for each clause, $G$ has 3 edges connecting its 3 vertices, and
- for each $u, v \in V$, if the literals labeling $u$ and $v$ are negations of each other, then $(u, v)$ is an edge in $G$.

Each vertex of $G$ has two edges of the first type and at most 28 edges of the second type. Hence, $G$ has degree at most 30. As an illustration, consider the formula $(x_1 \lor x_2 \lor x_3) \land (x_1 \lor x_2 \lor x_3)$. The graph produced by the reduction is given below.

2a) Show that the size of a maximum independent set in $G$ is precisely $\text{OPT}_{\text{SAT}}(\psi)$.

2b) Show that, in any graph, the complement of a maximum independent set is a minimum vertex cover. Conclude.

2b. The Steiner-Tree problem is as follows: given two disjoint sets of vertices $R$ and $S$ (required and Steiner) and a cost metric $c$ on $R \sqcup S$, find a tree whose nodes $V = R \sqcup S'$ are all required vertices of $R$ and a subset $S' \subseteq S$ of Steiner vertices, such that the sum of edge cost is minimum.

We now show the following result:
Theorem 3. There exists a polynomial-time gap preserving reduction from Vertex-Cover(30) to the Steiner-Tree problem that transforms an instance $G = (V, E)$ of Vertex-Cover(30) to an instance $H = (R, S, c)$ of Steiner-Tree such that:

- if $\text{OPT}_{VC}(G) \leq \frac{2}{3}|V|$, then $\text{OPT}_{ST}(H) \leq |R| + \frac{2}{3}|S| - 1$, and
- if $\text{OPT}_{VC}(G) > (1 + \epsilon_{VC})\frac{2}{3}|V|$, then $\text{OPT}_{ST}(H) > (1 + \epsilon_{ST})(|R| + \frac{2}{3}|S| - 1)$, where $\epsilon_{VC}$ is as in Theorem 2 and $\epsilon_{ST} = 2\epsilon_{VC}/47$.

The instance $H = (R, S, c)$ will be such that $G$ has a vertex cover of size $c$ iff $H$ has a Steiner tree of cost $|R| + c - 1$. $H$ will have a required vertex $r_e$ corresponding to each edge $e \in E$ and a Steiner vertex $s_v$ corresponding to each vertex $v \in V$. The edge costs are as follows. An edge between a pair of Steiner vertices is of cost 1, and an edge between a pair of required vertices is of cost 2. An edge $\{r_e, s_v\}$ is of cost 1 if edge $e$ is incident at vertex $v$ in $G$, and it is of cost 2 otherwise.

2bα) Prove that the cost function is indeed a metric.

2bβ) Show that if $G$ has a vertex cover of size $c$ then $H$ has a Steiner tree of cost $|R| + c - 1$.

2bγ) Assume that $H$ has a Steiner tree $T$ of cost $|R| + c - 1$ and that the edges of $G$ are not all disjoint. Explain how to transform $T$ into a Steiner tree of the same cost but using only edges of weight 1. Conclude that $G$ has a vertex cover of size $c$.

2bδ) Conclude.