# CSC 544 Homework 3

Calvin Higgins

September 2025

## Problem 1

Consider the following model of cellphone conversations:

We have an undirected graph $G = (V, E)$ where the vertices are people, and each edge indicates that two people are within range of each other. Whenever two people are talking, their neighbors must stay silent on that frequency to avoid interference. Thus, a set of conversations consists of a set of edges $C \subseteq E$, where the vertices in different edges in $C$ cannot be neighbors of each other.

The cellphone capacity $c(G)$ is the largest number of conversations that can take place simultaneously on one frequency, i.e. the size of the largest such set $C$. The **Cellphone Capacity Problem (CCP)** asks: given a graph $G$ and an integer $k$, is $c(G) \geq k$?

Prove that **CCP** is in **NP-Complete**.

### Solution

*Proof.* Clearly **CCP** is in **NP**. Given set of conversations $C$, we can check that no interference occurs and that $|C| \geq k$, so $c(G) \geq k$. This takes polynomial time with respect to the number of vertices and edges. Now, we show that **CCP** is **NP-Hard**.

We proceed by reduction from **3-Coloring**. Suppose $G = (V, E)$ is an instance of the **3-Coloring** problem. We aim to construct an instance of **CCP**, that is a graph $G' = (V', E')$ and integer $k$, such that $c(G') \geq k$ if and only if $G$ is 3-colorable.

For each vertex $v \in V$, create a new $K_3$, with vertices $v_r, v_g$ and $v_b$. The purpose of this gadget is to encode a vertex and its color. Exactly one of the edges in each $K_3$ will be selected as part of the maximum conversation set. The color of $v$ is encoded by the vertex that is not incident to the selected edge. For example, if $\{v_g, v_b\}$ is selected as the edge, then the color of vertex $v$ is red.

For each edge $\{u, v\} \in E$, create a new $K_{12}$ and twelve new edges uniquely pairing up each new vertex to a vertex in the $u$, $v$ vertex gadgets. The purpose of this gadget is to encode edges and the color constraint. If the $u, v$ vertex gadgets encode the same coloring, none of the new edges can be selected as part of the maximum conversation set. There are two cases. If an edge in the $K_{12}$ is taken, then no other edges incident to a vertex in the $K_{12}$ can be taken. Then, at most two additional edges can be taken, one from each $K_3$. If an edge from a $K_3$ to the $K_{12}$ is taken, then no other edges in that $K_3$ or the $K_{12}$ can be taken, so at most one additional edge in the other $K_3$ can be taken. In this manner, the $K_{12}$ ensures that if valid coloring exists for $u, v$, they will be colored as such in the maximum conversation set, since otherwise only two edges will be added instead of three.

Let $k = |E| + |V|$. The purpose of this gadget is to ensure that every vertex receives a color. Each vertex gadget can contribute at most one edge, and each edge gadget can contribute at most one edge to the maximum conversation set.

Clearly, this reduction is polynomial in $|V|$ and $|E|$. We aim to show that the reduction yields a yes instance if and only if $G$ is 3-colorable.

First, we show that if $G$ is 3-colorable, then the reduction yields a yes instance. Suppose $G$ is 3-colorable. For each vertex, take the edge in the gadget corresponding to its color. For example, if the vertex $v$ is red, take $\{v_g, v_b\}$. Then, consider each edge gadget. Since the incident vertices are different colors, there are four vertices, the color vertices from the $K_3$'s, and their neighbors in the $K_{12}$, such that the edge between the two vertices in the $K_{12}$ can be taken, because neither of the color vertices are taken, and none of the other vertices in the $K_{12}$ are taken. Thus, the total number of taken edges is $|V|+|E|$ as desired.

Next, we show that if the reduction yields a yes instance, then $G$ is 3-colorable. Suppose the reduction yields a yes instance. Color each vertex in $G$ according to the color of the corresponding vertex gadget. Since the reduction yields a yes instance, every edge gadget must have a taken edge, so the colors of the vertex gadgets must be different. Hence this is valid 3-coloring.

Since there is a polynomial time reduction from **3-Coloring** to **CCP**, and **3-Coloring** is in **NP-Hard**, **CCP** is in **NP-Hard**.

Since **CCP** is in **NP** and **NP-Hard**, it is in **NP-Complete**. //

# Problem 2(a)

TLC Coffee has decided to expand to maximize profits over all of South County, by building new TLCs on many street corners. The management turns to URI's Computer Science majors for help. The students come up with a representation of the problem:

The street network is described as an arbitrary undirected graph $G = (V, E)$, where the potential restaurant sites are the vertices of the graph. Each vertex $u$ has a nonnegative integer value $p(u)$, which describes the potential profit of site $u$. Two restaurants cannot be built on adjacent vertices (to avoid self-competition). The students further come up with an exponential-time algorithm that chooses a set $U \subseteq V$, but it runs in $O(2^{|V|}|E|)$ time. The TLC ownership is dismayed, and is convinced they should have hired UConn's CS majors instead. In order to prove that nobody else could have come up with a better algorithm, you need to prove the hardness of the **COFFEE** problem.

Define **COFFEE** to be the following decision problem: given an undirected graph $G = (V, E)$, given a mapping $p$ from vertices $u \in V$ to nonnegative integer profits $p(u)$, and given a nonnegative integer $k$, decide whether there is a subset $U \subseteq V$ such that no two vertices in $U$ are neighbors in $G$, and such that $\sum\limits_{u \in U} p(u) \geq k$.

Prove that **COFFEE** is in **NP-Hard**.

## Solution

*Proof.* We proceed by reduction from **3-Coloring**. Suppose $G = (V, E)$ is an instance of the **3-Coloring** problem. We aim to construct an instance of **COFFEE**, that is a graph $G' = (V', E')$, a profit function $p$ and integer $k$, such that it is a yes instance if and only if $G$ is 3-colorable.

For each vertex $v \in V$, create a new $K_3$, with vertices $v_r$, $v_g$ and $v_b$. The purpose of this gadget is to encode a vertex and its color. A coffee shop can be built on at most one of the vertices in each $K_3$. The color of $v$ is encoded by the chosen vertex. For example, if a coffee shop is built on $v_r$, then the color of vertex $v$ is red.

For each edge $\{u, v\} \in E$, create three new edges $\{u_r, v_r\}$, $\{u_g, v_g\}$ and $\{u_b, v_b\}$. The purpose of this gadget is to encode edges and the color constraint. The $u$ and $v$ gadgets cannot encode the same coloring, since otherwise, two coffee shops would be adjacent.

Finally, take $k = |V|$. The purpose of this gadget is to ensure that every vertex is colored. Since there is at most one coffee shop per vertex gadget and there are $|V|$ vertex gadgets, this ensures every vertex gadget has at least one coffee shop, that is, the vertex is colored.

More formally, let $V' = \bigcup\limits_{v \in V} \{v_r, v_g, v_b\}$, $E' = \bigcup\limits_{\{u,v\} \in E} \{\{u_r, v_r\}, \{u_g, v_g\}, \{u_b, v_b\}\}$, $G' = (V', E')$, and $k = |V|$. Clearly, this reduction is polynomial in $|V|$ and $|E|$.

We aim to show that $(G', k)$ is a yes instance if and only if $G$ is 3-colorable.

First, we show that if $G$ is 3-colorable, then $(G', k)$ is a yes instance. Suppose $G$ is 3-colorable. Take $U = \{v_c \mid c \text{ is the color of } v\}$. By construction, $U$ contains exactly one vertex from each $K_3$ vertex gadget. Moreover, no two vertices $u_c, v_c \in U$ are adjacent, since otherwise, the vertices $u, v \in V$ would have the same color $c$ but, by construction, $\{u, v\} \in E$, which is impossible. Also $|U| = |V| \geq k = |V|$, so $(G', k)$ is a yes instance.

Next, we show that if $(G', k)$ is a yes instance, then $G$ is 3-colorable. Suppose $(G', k)$ is a yes instance. Then there exists some set $U \subseteq V'$, such that $U$ is valid and $|U| \geq k = |V|$. By construction, each of the $|V|$ vertex gadgets can have at most one coffee shop, so $U$ must contain exactly $|V|$ coffee shops, one from each vertex gadget. Take the color of $v$ to be the "color" of the vertex selected in $v$'s gadget. Then, we have a color for each vertex. For all edges $\{u, v\}$ in $G$, $G'$ contains edges between the same color vertices in $v$ and $u$'s gadgets, $u$ and $v$ must have different colors. Therefore, we have a valid 3-coloring of $G$.

Since there is a polynomial time reduction from **3-Coloring** to **COFFEE**, and **3-Coloring** is in **NP-Hard**, **COFFEE** is in **NP-Hard**. //

# Problem 2(b)

Explain why **COFFEE** in **NP-Hard** implies that, if there is a polynomial-time algorithm to solve **COFFEE**, i.e., to output a subset $U$ that maximizes the total profit, then $\mathbf{P} = \mathbf{NP}$.

## Solution

*Proof.* Suppose there exists a polynomial-time algorithm to solve **COFFEE**. Fix some problem $\Pi \in \mathbf{NP}$. We will show that there is a polynomial-time algorithm to solve $\Pi$. Since **COFFEE** is in **NP-Hard**, there is polynomial-time reduction from $\Pi$ to **COFFEE**. Then, we can solve **COFFEE** in polynomial-time by the assumption. Since $\Pi$ can be solved in polynomial-time, $\mathbf{NP} \subseteq \mathbf{P}$, so $\mathbf{P} = \mathbf{NP}$. //

# Problem 3

**1-in-3-SAT** is a variant of **3-SAT** where each clause is satisfied if exactly one of its literals is true. So for instance, the sentence $(x, y, \neg z) \wedge (\neg x, \neg y, \neg z)$ could be satisfied by setting $x$ to true, $y$ to false, and $z$ to true. However, $(x, y, z) \wedge (\neg x, \neg y, \neg z)$ cannot be satisfied.

Prove that **1-in-3-SAT** is in **NP-Complete**.

## Solution

*Proof.* Clearly **1-in-3-SAT** is in **NP**. Given a satisfying assignment, we can check that the sentence is satisfied and that exactly one true literal in each clause in true in polynomial-time with respect to the number of clauses. Now, we show that **1-in-3-SAT** is **NP-Hard**.

We proceed by reduction from **3-Coloring**. Suppose $G = (V, E)$ is an instance of the **3-Coloring** problem. We aim to construct an instance of **1-in-3-SAT**, that is a set of literals $L$ and clauses $C$, such that it is a yes instance if and only if $G$ is 3-colorable.

For each vertex $v \in V$, create literals $r_v, g_v$ and $b_v$ and clause $(r_v, g_v, b_v)$. The purpose of this gadget is to encode a vertex and its color. Exactly one of $r_v, g_v$ and $b_v$ must be true. The color of $v$ is encoded by the true literal. For example, if $r_v$ is true, the color of vertex $v$ is red.

For each edge $\{u, v\} \in E$, create three new clauses $(r_u, r_v, x_{ruv})$, $(g_u, g_v, x_{guv})$ and $(b_u, b_v, x_{buv})$. The purpose of this gadget is to encode edges and the color constraint. The $u$ and $v$ gadgets cannot encode the same coloring, since otherwise, two literals in a clause would be true. The $x$ literals ensure that when neither vertex is a given color, the clause is still satisfiable.

More formally, let

$$L = \left( \bigcup_{v \in V} \{r_v, g_v, b_v\} \right) \cup \left( \bigcup_{\{u,v\} \in E} \{x_{ruv}, x_{guv}, x_{buv}\} \right)$$

and

$$C = \left( \bigcup_{v \in V} \{(r_v, g_v, b_v)\} \right) \cup \left( \bigcup_{\{u,v\} \in E} \{(r_u, r_v, x_{ruv}), (g_u, g_v, x_{guv}), (b_u, b_v, x_{buv})\} \right)$$

Clearly, this reduction is polynomial in $|V|$ and $|E|$. We aim to show that $(L, C)$ is a yes instance if and only if $G$ is 3-colorable.

First, we show that if $G$ is 3-colorable, then $(L, C)$ is a yes instance. Suppose $G$ is 3-colorable. For each vertex $v$, set the literal corresponding to its color to true, and the others to false. Then, all clauses without an $x$ must be true since each node has exactly one color. Consider each clause with an $x$. Each clause either has exactly one true literal or zero since adjacent vertices have different colors. If the clause is false, set the $x$ to true. Then, all clauses have exactly one true literal as desired.

Next, we show that if $(L, C)$ is a yes instance, then $G$ is 3-colorable. Suppose $(L, C)$ is a yes instance. Then exactly one of the color literals associated with vertex $v$ is true. Set $v$ to that corresponding color. Each node has a color, and adjacent nodes have different colors, since otherwise, one of the three edge clauses would have two true literals.

Since there is a polynomial time reduction from **3-Coloring** to **1-in-3-SAT**, and **3-Coloring** is in **NP-Hard**, **1-in-3-SAT** is in **NP-Hard**.

Since **1-in-3-SAT** is in **NP** and **NP-Hard**, it is in **NP-Complete**. //

# Problem 4

Consider the following variant of **3-Coloring**:

In addition to the graph $G$, we are given a forbidden color $f_v$ for each vertex $v$. For instance, vertex 1 may not be red, vertex 2 may not be green, and so on (with still a total palette of 3 colors). We can ask whether $G$ has a proper 3-coloring in which no vertex is colored with its forbidden color.

The **Constrained-3-Coloring** problem asks, given a graph $G = (V, E)$ with $n$ vertices and a list $f_1, \ldots, f_n \in \{1, 2, 3\}$, determine if $G$ has a proper 3-coloring $c_1, \ldots, c_n$ where $c_v \neq f_v$ for all vertices $v \in V$.

Show **Constrained-3-Coloring** is in **NP-Complete** or show that **Constrained-3-Coloring** is in **P**.

## Solution

*Proof.* We will show that **Constrained-3-Coloring** is in **P**.

We proceed by reduction to **2-SAT**. Suppose $G = (V, E)$ where $f_v$ is the restricted color for vertex $v$. We aim to construct an instance of **2-SAT** such that $G$ is colorable if and only if the **2-SAT** instance is satisfiable.

For each vertex $v \in V$, arbitrarily label its two possible colors $a$ and $b$. Create two variables $v_a, v_b$, and two clauses $(v_a \vee v_b)$ and $(\neg v_a \vee \neg v_b)$. The purpose of this gadget is to represent vertex colors and their restriction. Exactly one of $v_a$ and $v_b$ must be true: this is the encoded color of $v$.

For each edge $\{u, v\} \in E$ and color $c$, create a clause $(\neg u_c \vee \neg v_c)$. The purpose of this gadget is to enforce the restriction that two adjacent nodes cannot have the same color.

Clearly, this reduction is polynomial in $|V|$ and $|E|$. We aim to show that $G$ is colorable if and only if the **2-SAT** instance is satisfiable

First, we show that if $G$ is colorable, then **2-SAT** instance is satisfiable. Suppose $G$ is colorable. For each vertex $v \in V$, set $v_a, v_b$ according to the color of $v$. Then, the vertex gadget clauses are true because exactly one of $v_a, v_b$ will be true. Also, the edge clauses will be true since no adjacent vertices have the same color. Hence, the **2-SAT** instance is satisfiable as desired.

Next, we show that if **2-SAT** instance is satisfiable, then $G$ is colorable. Suppose **2-SAT** instance is satisfiable. For each $v_a, v_b$, by the vertex clause gadget, exactly one of $v_a, v_b$ is true. Set $v$ to the corresponding coloring. Then every vertex in $G$ has a color, and no two adjacent vertices have the same color, since otherwise, some clause $(\neg u_c \vee \neg v_c)$ would be false. Hence $G$ is colorable as desired.

Since there is a polynomial time reduction from **Constrained-3-Coloring** to **2-SAT**, and **2-SAT** is in **P**, **Constrained-3-Coloring** is in **P**. //