Aula Prática 13

ASA 2024/2025

Q1 (T2 19/20) Uma matriz de incompatibilidades é uma matriz quadrada cujas células guardam valores decimais entre 0 e 1. Intuitivamente, dada uma matriz de incompatibilidades M, $n \times n$, a célula M_{ij} guarda a incompatibilidade entre os índices i e j; $M_{ij} = 0$ se i e j são completamente compatíveis e $M_{ij} = 1$ se i e j são completamente incompatíveis.
Dado um sub-conjunto de índices $I \subseteq \{1, ..., n\}$, o nível de incompatibilidade do conjunto é dado por: $\sum_{i,j \in I} M_{ij}$. O problema das incompatibilidades define-se formalmente da seguinte maneira:

$$\begin{aligned}\textbf{Incompat} = \{\langle M, k, v \rangle \mid M & \text{ contém um sub-conjunto de índices} \\ & \text{de tamanho } k \text{ e incompatibilidade igual ou inferior a } v\}\end{aligned}$$

1. Mostre que o problema **Incompat** está em **NP**.
2. Mostre que o problema **Incompat** é NP-difícil por redução a partir do problema **ISet**, que é sabido tratar-se de um problema NP-completo e que se define em baixo. Não é necessário provar formalmente a equivalência entre os dois problemas; é suficiente indicar a redução e a respectiva complexidade.
   Pista: Dado um grafo G indique como construir uma matriz de incompatibilidades cujos índices correspondem aos vértices de G tendo em conta o problema **ISet**.

Problema ISet: Seja $G = (V, E)$ um grafo não dirigido; dizemos que $V' \subseteq V$ é um conjunto de vértices independentes em G se e apenas se $\forall u, v \in V'.(u, v) \notin E$. O problema **ISet** define-se formalmente da seguinte maneira:

$$\textbf{ISet} = \{\langle G, k \rangle \mid G \text{ contém um conjunto de vértices independentes de tamanho } k\}$$

Solução:

1. O algoritmo de verificação recebe como input uma possível instância $\langle M, k, v \rangle$ e um conjunto de índices I (o certificado). O algoritmo tem de verificar que $|I| = k$ e que $\sum_{i,j \in I} M_{ij} \leq v$. Observamos os certificados têm tamanho $O(n)$ e que a verificação se faz em tempo $O(n^2)$, o tempo de calcular o somatório.

2. Dada uma possível instância $\langle G, k \rangle$ do problema **ISet**, começamos por construir uma matriz de incompatibilidades M_G cujos índices correspondem aos vértices de G. Para tal, admitimos que $|V| = n$ e que os vértices de V estão numerados de 1 a n, sendo v_i o i-ésimo vértice. Assim sendo, definimos a matriz M_G como se segue:

   $$(M_G)_{ij} = \begin{cases} 1 & \text{se } (v_i, v_j) \in E \\ 0 & \text{caso contrário} \end{cases}$$

   Uma vez estabelecida a matriz M_G, a redução é definida da seguinte maneira:

   $$f(\langle G, k \rangle) = \langle M_G, k, 0 \rangle$$

   - Equivalência a estabelecer: $\langle G, k \rangle \in \textbf{ISet} \iff \langle M_G, k, 0 \rangle \in \textbf{Incompat}$
   - Complexidade da redução: $O(|V|^2)$.

Q2 (R2 19/20) Seja $C = \{c_1, ..., c_n\}$ uma colecção de cromos e $\mathcal{A} = \{a_1, ..., a_m\}$ um grupo de amigos que coleccionam cromos. Cada membro do grupo detém um subconjunto de C; seja C_i o conjunto de cromos detido por a_i e $\mathcal{C} = \{C_i \mid 1 \leq i \leq m\}$ o conjunto dos conjuntos de cromos de todos os membros do grupo. Os membros do grupo pretendem determinar o mais pequeno conjunto de cromos que contém pelo menos um cromo detido por cada membro do grupo. Formalmente, este problema pode ser modelado através do seguinte problema de decisão:

$$\textbf{SharedStickers} = \{\langle C, \mathcal{C}, k\rangle \mid \exists X \subseteq C.\, |X| = k \wedge \forall_{1\leq i\leq m}.\, C_i \cap X \neq \emptyset\}$$

1. Mostre que o problema **SharedStickers** está em **NP**.
2. Mostre que o problema **SharedStickers** é **NP**-difícil por redução a partir do problema da *Cobertura de Vértices* que foi estudado nas aulas e que recordamos em baixo.

Problema da Cobertura de Vértices: Seja $G = (V, E)$ um grafo não dirigido; dizemos que $V' \subseteq V$ é uma cobertura de vértices se e só se: $\forall (u, v) \in E.\, u \in V' \vee v \in V'$. O problema da cobertura de vértices, **VCover**, define-se formalmente da seguinte maneira:

$$\textbf{VCover} = \{\langle G, k\rangle \mid G \text{ contém uma cobertura de vértices de tamanho } k\}$$

Solução:

1. O algoritmo de verificação recebe como input uma possível instância $\langle C, \mathcal{C}, k\rangle$ e um conjunto de cromos X (o certificado). O algoritmo tem de verificar que $|X| = k$ e que $X \cap C_i \neq \emptyset$, para todo o C_i em $\mathcal{C}$. Observamos os certificados têm tamanho $O(n)$ e que a verificação se faz em tempo $O(m.n^2)$, a complexidade de se calcular a intersecção de cada um dos conjuntos em $\mathcal{C}$ com X utilizando um algoritmo naif de complexidade quadrática.

2. Dada uma possível instância $\langle G, k\rangle$ do problema **VCover**, começamos por construir uma colecção de cromos C e um conjunto de conjuntos $\mathcal{C}$. Intuitivamente, os vértices do grafo G correspondem aos cromos da colecção e cada arco corresponde a um conjunto de dois vértices. Formalmente:

   $$\langle G, k\rangle \in \textbf{VCover} \Leftrightarrow \langle C, \mathcal{C}, k'\rangle \in \textbf{SharedStickers}$$

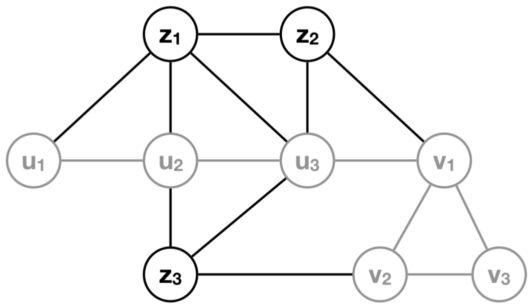
   onde:

   - $C = V$;
   - $\mathcal{C} = \{\{u, v\} | (u, v) \in E\}$;
   - $k' = k$.

   Complexidade da redução: $O(V + E)$.

Q3 (T2 20/21) Um grafo diz-se um *kite*[1] de grau n se é constituído por $2.n$ vértices, tais que n vértices formam um clique e os restantes n vértices formam uma cauda ligada a um dos vértices do clique.

Dado um grafo $G = (V, E)$ e um inteiro k, o problema **Kite** consiste em determinar se G contém um *kite* de grau k. Por exemplo, o grafo em baixo contém vários kites de grau 3, um dos quais está identificado em cinzento.



Formalmente, o problema **Kite** pode ser modelado através do seguinte problema de decisão:

$$\textbf{Kite} = \{\langle G, k\rangle \mid G \text{ contém um } \textit{kite} \text{ de grau } k\}$$

1. Mostre que o problema **Kite** está em **NP**.
2. Mostre que o problema **Kite** é NP-difícil por redução a partir do problema **Clique** estudado nas aulas. Não é necessário provar formalmente a equivalência entre os dois problemas; é suficiente indicar a redução e a respectiva complexidade.

Solução:

1. O algoritmo de verificação recebe como input uma possível instância $\langle G = (V, E), k\rangle$ e um certificado na forma de um triplo $\langle V_1, V_2, u, v\rangle$ tal que:

   - *Restrição 1:* $V_1 \cap V_2 = \emptyset$, $|V_1| = |V_2| = k$;
   - *Restrição 2:* $u \in V_1$, $v \in V_2$, $(u, v) \in E$;
   - *Restrição 3:* $G_1 = (V_1, E_1)$, com $E_1 = \{(w, z) \mid (w, z) \in E \ \wedge\ w, z \in V_1\}$, forma uma linha com k elementos;
   - *Restrição 4:* $G_2 = (V_2, E_2)$, com $E_2 = \{(w, z) \mid (w, z) \in E \ \wedge\ w, z \in V_2\}$, forma um clique de tamanho k.

   Em primeiro lugar, observamos que o certificado tem tamanho $O(V)$. O algoritmo de verificação tem de verificar que as restrições enunciadas em cima são verificadas. Analisamos cada restrição separadamente:

   - *Restrição 1:* $O(V)$;
   - *Restrição 2:* $O(1)$;
   - *Restrição 3:* $O(V)$ (encontrar o vértice sem nós incidentes em V_1 e efectuar uma DFS modificada que não explora arcos em V_2);
   - *Restrição 4:* $O(V^2)$ (verificar se cada vértice em V_2 está ligado a todos os outros vértices em V_2).

[1] Em português *kite* diz-se papagaio.

2. Dada uma instância $\langle G = (V, E), k\rangle$ do problema **Clique** temos de construir uma instância $\langle G' = (V', E'), k\rangle$ do problema **Kite** tal que:

$$\langle G, k\rangle \in \textbf{Clique} \ \Leftrightarrow \langle G', k\rangle \in \textbf{Kite}$$

Intuitivamente definimos o grafo G' acrescentando a cada vértice $v \in V$ uma cauda com k vértices. Admitindo que os vértices de G se encontram numerados: $V = \{v_1, ..., v_n\}$, definimos formalmente o grafo $G' = (V', E')$ como se segue:

- $V' = V \cup \{u_1^1, ..., u_1^k\} \cup \{u_2^1, ..., u_2^k\} \cup ... \cup \{u_n^1, ..., u_n^k\}$
- $E' = E \cup \{(u_i^j, u_i^{j+1}) \mid 1 \leq i \leq n \ \wedge \ 1 \leq j \leq k{-}1\} \cup \{(u_i^k, v_i) \mid 1 \leq i \leq n\}$

Complexidade da redução: $O(V^2)$

Q4 (E1 21/22) Uma família com n membros $\{M_1, ..., M_n\}$ prepara-se para cozinhar a maior pizza de sempre. Para tal têm de escolher os ingredientes a incluir na pizza de entre k ingredientes disponíveis $\{I_1, ..., I_k\}$. Cada familiar M_i deve indicar os ingredientes que não deseja incluir na pizza e os ingredientes que deseja incluir. Assim sendo, associamos a cada familiar M_i um par com dois conjuntos de ingredientes, (C_i, C'_i), onde C_i contém os ingredientes a incluir e C'_i os ingredientes a excluir.

Tratando-se de uma família pouco conflituosa para que um familiar se considere satisfeito basta que uma das suas escolhas seja atendida: um dos seus ingredientes preferidos seja incluído ou um dos preteridos não o seja. Por exemplo, suponha que o pai quer fiambre e queijo e não quer ananás; para que a pizza escolhida satisfaça o pai, basta que contenha fiambre ou queijo ou não contenha ananás.

O problema da escolha de ingredientes para pizza, **PizzaIngredients**, consiste em determinar se existe um conjunto de ingredientes que satisfaça todos os membros da família e é modelado formalmente através do seguinte problema de decisão:

$$\textbf{PizzaIngredients} = \{\langle \mathcal{I} \rangle \mid \text{ existe uma escolha de ingredientes compatível com } \mathcal{I}\}$$

Onde $\mathcal{I}$ denota o conjunto de pares que representam as escolhas da família.

1. Mostre que o problema **PizzaIngredients** está em **NP**.
2. Mostre que o problema da escolha dos ingredientes é **NP**-difícil por redução a partir do problema **3-CNFSAT** estudado nas aulas. Não é necessário provar formalmente a equivalência entre os dois problemas; é suficiente indicar a redução e a respectiva complexidade.

Solução:

1.
   - *Certificado:* o conjunto X de ingredientes a incluir.
   - *Tamanho do Certificado:* $|X| \in O(k)$
   - *Algoritmo de verificação:* Verificar se X é compatível com cada elemento do conjunto $\mathcal{I}$. Começamos por calcular o conjunto $\bar{X} = \{I_1, ..., I_k\} \backslash X$. Para cada par (C_i, C'_i), verificamos se: $X \cap C_i \neq \emptyset$ ou $\bar{X} \cap C'_i \neq \emptyset$.
   - *Complexidade do algoritmo de verificação:* A intersecção de conjuntos faz-se em tempo linear, pelo que a verificação de cada par para custa: $O(k)$. Assim, a verificação de todos os n pares faz-se em tempo $O(k.n)$.
2. Há que mostrar que **3-CNFSAT** $\leq_P$ **PizzaIngredients**.
   - *Redução:* Cada cláusula corresponde a um membro da família e cada variável a um ingrediente. As variáveis negadas na cláusula correspondem aos ingredientes a excluir e as variáveis não negadas aos ingredientes a incluir. Por exemplo, a cláusula $(x_1 \vee \neg x_2 \vee x_3)$ é mapeada no par $(\{x_1, x_3\}, \{x_2\})$. A redução tem complexidade: $O(n)$, onde n é o número de cláusulas.