Université Clermont Auvergne
BUT Info 3A
R5.A.04 - Qualité algorithmique
Année 2024-2025

Florent Foucaud
Pascal Lafourcade
Malika More

TD5 - Réductions polynomiales

1 Rappel de cours : réductions polynomiales, classes de complexité

Définition (Problème de décision)**.** Un *problème de décision* est défini par une instance (entrée), et une question à laquelle on doit pouvoir répondre par oui ou par non. Une instance pour laquelle la réponse est “oui” est appelée *positive*.

Définition (Classe NP)**.** La classe NP contient tous les problèmes de décision pour lesquels, étant donné une solution (appelée *certificat*), on peut décider en temps polynomial (en la taille de l'instance) si l'instance est positive ou pas.

Définition (Réduction polynomiale)**.** Une *réduction polynomiale* d'un problème P_1 à un problème P_2 est un algorithme f qui construit, en temps polynomial, pour toute instance I de P_1 une instance $f(I)$ de P_2, telles que I_1 est une instance positive de P_1 *si et seulement si* $f(I_1)$ est une instance positive de P_2. (C'est-à-dire : (1) si I est une instance posituve de P_1, alors $f(I)$ est une instance positive de P_2, et (2) si $f(I)$ est une instance positive de P_2, alors I est une instance positive de P_1.)

Si il existe une telle réduction polynomiale f de P_1 à P_2 et que P_2 peut être résolu en temps polynomial, alors P_1 peut aussi être résolu en temps polynomial. En effet, il suffit d'appliquer la réduction f à une instance I de P_1, et résoudre P_2 sur $f(I)$ en temps polynomial. Intuitivement, P_1 “n'est pas plus difficile à résoudre” que P_2.

Nous avons déjà vu les années précédentes des exemples de réductions polynomiales. Par exemple dans le cours de graphes en BUT1, nous avons vu comment réduire le problème des chemins disjoints entre deux sommets d'un graphe orienté, au problème de flots dans un graphe orienté et valué. Nous avions aussi réduit le problème des couplages dans un graphe biparti non orienté au problème de flot dans un graphe orienté.

Dans le cours de Méthodes d'Optimisation en BUT2, nous avions réduit le problème de flot dans un graphe orienté, à la programmation linéaire.

Cette année au TD4, nous avons vu plusieurs réductions polynomiales.

Définition (Problèmes NP-difficiles et NP-complets)**.** Un problème P_1 est *NP-difficile* si tous les problèmes de NP ont une réduction polynomiale vers P_1.

Le problème P_1 est *NP-complet* s'il est NP-difficile et dans NP. Le problème 3-SAT vu en cours est un exemple de problème NP-complet.

Si un problème P_1 est NP-difficile, *tous* les problèmes de NP peuvent être résolus par un algorithme qui résoud P_1, au prix d'un facteur polynomial de calcul de la réduction. S'il existe un algorithme polynomial pour P_1, alors il existe un algorithme polynomial pour tous les problèmes de NP ! C'est une propriété très forte.

Comment montrer que notre problème favori P_1 est NP-difficile ? Si il existe une réduction polynomiale depuis un problème P_2 connu comme étant NP-difficile vers notre problème P_1, alors P_1 est aussi NP-difficile. En effet, n'importe quel problème se réduit vers P_2, et une réduction appliquée à une réduction est aussi une réduction.

Nous avons vu en cours que le problème 3-SAT est NP-complet, et qu'il existe une réduction polynomiale de celui-ci vers 3-Coloration (qui est dans NP), ce dernier est donc aussi NP-complet.

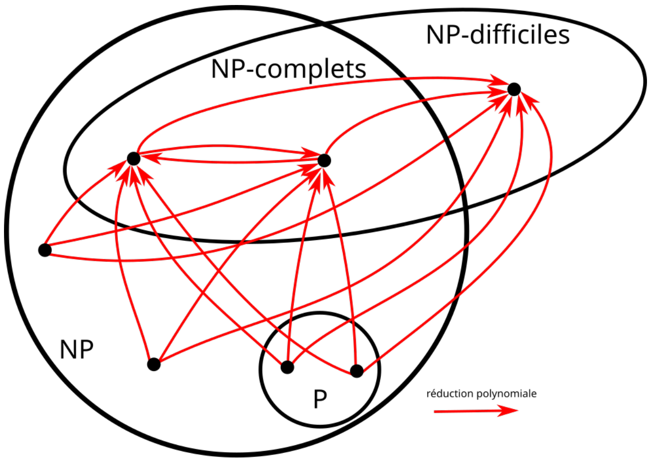


FIGURE 1 – Schéma des classes de problèmes de P, NP et NP-difficiles

2 Exercices

Exercice 1 (3-SAT et ensembles indépendants dans un graphe)**.**

Rappel : dans un graphe $G = (V, E)$, on rappelle qu'un *ensemble indépendant* est un ensemble S de sommets de G tel que aucune paire de sommets de S n'est reliée par une arête.

Soit $X = \{x_1 \dots, x_n\}$ un ensemble de variables booléennes (qui ont comme valeur vrai ou faux). Un *littéral* est soit une variable x_i, soit sa négation $\overline{x_i}$. Une *3-clause* est une formule constituée d'un OU logique (noté $\vee$) d'au plus trois littéraux. Une *formule booléenne en 3-CNF* est une formule qui est un ET logique (noté $\wedge$) d'un ensemble de 3-clauses.

Par exemple : $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee \overline{x_1}) \wedge (\overline{x_1} \vee x_2 \vee x_3)$

On définit/rappelle les deux problèmes de décision associés :

ENSEMBLE INDÉPENDANT
Instance: Un graphe G, un entier k.
Question: G admet-il un ensemble indépendant de taille *au moins* k ?

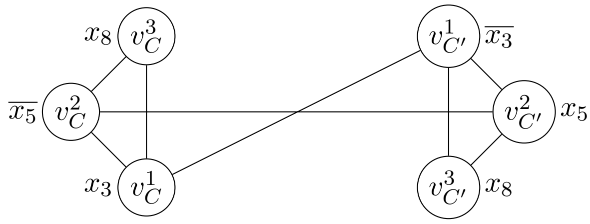
3-SAT
Instance: Une formule booléenne F en 3-CNF sur une ensemble $X = \{x_1 \dots, x_n\}$ de variables.
Question: F est-elle satisfaisable, c'est-à-dire, peut-on trouver une assignation booléenne des variables de X qui rende F vraie ?

On sait (voir cours magistral) que 3-SAT est NP-complet.

1. Trouver une assignation booléenne des variables x_1, x_2, x_3 qui rende la formule
$$(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee \overline{x_1}) \wedge (\overline{x_1} \vee x_2 \vee x_3)$$
vraie, et une autre assignation qui la rende fausse.
2. Expliquer pourquoi 3-SAT et ENSEMBLE INDÉPENDANT sont dans NP.
3. À partir d'une formule F en 3-CNF sur une ensemble X de variables booléennes, on construit le graphe G_F de la façon suivante.
Pour chaque clause $C = (l_C^1 \vee l_C^2 \vee l_C^3)$ de F (où chaque l_C^i, $i \in \{1, 2, 3\}$, est un littéral x_k ou $\overline{x_k}$, avec $x_k \in X$) on crée un nouveau sommet v_C^i pour chaque littéral l_C^i de C. On relie deux sommets v_C^i et $v_{C'}^j$ $(i, j \in \{1, 2, 3\})$ du graphe dans les situations suivantes :
   - Si $C = C'$ (donc v_C^i et $v_{C'}^j$ correspondent à deux littéraux d'une même clause)
   - Si $C \neq C'$ et l_C^i est la négation de $l_{C'}^j$ (par exemple $l_C^i = x_k$ et $l_{C'}^j = \overline{x_k}$)

Le nombre total de sommets de G_F est donc égal à la somme des tailles des clauses.

Par exemple, pour la formule $F_1 = (x_8 \vee \overline{x_5} \vee x_3) \wedge (\overline{x_3} \vee x_5 \vee x_8)$, où $C = (l_C^1 \vee l_C^2 \vee l_C^3)$ avec $l_C^1 = x_3$, $l_C^2 = \overline{x_5}$ et $l_C^3 = x_8$, $C' = (l_{C'}^1 \vee l_{C'}^2 \vee l_{C'}^3)$ avec $l_{C'}^1 = \overline{x_3}$, $l_{C'}^2 = x_5$ et $l_{C'}^3 = x_8$ le graphe G_{F_1} obtenu est :



Question : Dessiner le graphe G_{F_2} pour la formule $F_2 = (x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee \overline{x_1}) \wedge (\overline{x_1} \vee x_2 \vee x_3)$ (on notera $C_1 = (\overline{x_3} \vee x_2 \vee x_1)$, $C_2 = (\overline{x_2} \vee \overline{x_1})$ et $C_3 = (\overline{x_1} \vee x_2 \vee x_3)$).

4. Déterminer un plus grand ensemble indépendant du graphe G_{F_2}.
5. Montrer que si une formule F en 3-CNF est satisfaisable, alors G_F a un ensemble indépendant de taille au moins k, où k est le nombre de clauses de F.
6. Montrer que si G_F a un ensemble indépendant de taille au moins k (où k est le nombre de clauses de F), alors G_F est satisfaisable.
7. En déduire que la transformation précédente donne une réduction polynomiale de 3-SAT vers Ensemble Indépendant.
8. Qu'en déduit-on pour le problème Ensemble Indépendant ?
9. Qu'en déduit-on pour les problèmes Clique et Couverture Par Sommets étudiés au TD4 ?
10. En se souvenant du TD4, comment modifier la réduction ci-dessus pour obtenir une réduction directe de 3-SAT vers Clique ?

Exercice 2 (3-SAT et ensembles dominants dans un graphe)**.**

Rappel : dans un graphe $G = (V, E)$, un *ensemble dominant* est un ensemble S de sommets de G tel que tout sommet est soit dans S, soit a un voisin dans S.

On rappelle le problème de décision associé :

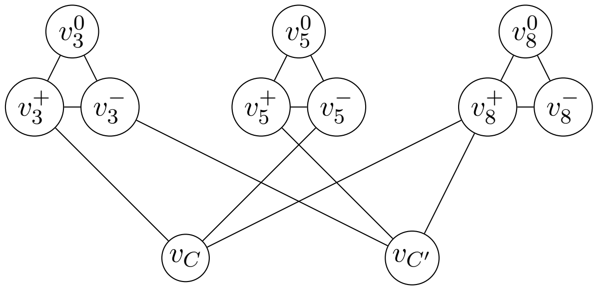
Ensemble Dominant
Instance: Un graphe G, un entier k.
Question: G admet-il un ensemble dominant de taille *au plus* k ?

On sait (voir cours magistral) que 3-SAT est NP-complet.

1. Expliquer pourquoi Ensemble Dominant est dans NP.
2. À partir d'une formule F en 3-CNF sur un ensemble X de variables booléennes, on construit le graphe H_F de la façon suivante.
   Pour chaque clause C de F, on crée un nouveau sommet v_C. Pour chaque variable x_i de X, on ajoute trois sommets : v_i^0, v_i^+ et v_i^- (v_i^+ correspond au littéral x_i tandis que v_i^- correspond au littéral $\overline{x_i}$). De plus on ajoute les arêtes suivantes :
   - Pour toute variable x_i de X, on ajoute une arête entre v_i^+ et v_i^-, entre v_i^0 et v_i^+ ainsi qu'entre v_i^0 et v_i^-.
   - Pour toute clause $C = (l_C^1 \vee l_C^2 \vee l_C^3)$ de F, on ajoute des arêtes entre v_C et les trois sommets correspondant à l_C^1, l_C^2, l_C^3. Par exemple, si $C = (x_i \vee \overline{x_j} \vee x_k)$ on relie v_C à v_i^+, v_j^- et v_k^+. (Si C ne contient que deux littéraux on adapte en conséquence.)

Le nombre total de sommets de H_F est donc égal à 3 fois le nombre de variables plus le nombre de clauses.

Par exemple, pour la formule $F_1 = (x_3 \vee \overline{x_5} \vee x_8) \wedge (\overline{x_3} \vee x_5 \vee x_8)$, où $C = (l_C^1 \vee l_C^2 \vee l_C^3)$ avec $l_C^1 = x_3$, $l_C^2 = \overline{x_5}$ et $l_C^3 = x_8$, $C' = (l_{C'}^1 \vee l_{C'}^2 \vee l_{C'}^3)$ avec $l_{C'}^1 = \overline{x_3}$, $l_{C'}^2 = x_5$ et $l_{C'}^3 = x_8$ le graphe H_{F_1} obtenu est :



Question : Dessiner le graphe H_{F_2} pour la formule $F_2 = (x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee \overline{x_1}) \wedge (\overline{x_1} \vee x_2 \vee x_3)$ (on notera $C_1 = (\overline{x_3} \vee x_2 \vee x_1)$, $C_2 = (\overline{x_2} \vee \overline{x_1})$ et $C_3 = (\overline{x_1} \vee x_2 \vee x_3)$).

3. Déterminer un plus petit ensemble dominant du graphe H_{F_2}.
4. Montrer que si une formule F en 3-CNF est satisfaisable, alors H_F a un ensemble dominant de taille au plus k, où k est le nombre de variables de F.
5. Montrer que si H_F a un ensemble dominant de taille au plus k (où k est le nombre de variables de F), alors H_F est satisfaisable.
6. En déduire que la transformation précédente donne une réduction polynomiale de 3-SAT vers Ensemble Dominant.
7. Qu'en déduit-on pour le problème Ensemble Dominant ?