

# Chapitre 12

## Quelques problèmes NP-complets

Maintenant que nous connaissons la NP-complétude d'au moins un problème (SAT), nous allons montrer qu'un très grand nombre de problèmes sont NP-complets.

Le livre [Garey and Johnson, 1979] en recensait plus de 300 en 1979. Nous n'avons pas l'ambition d'en présenter autant, mais de décrire quelques problèmes NP-complets célèbres, de façon à montrer quelques preuves de NP-complétude.

### 12.1 Quelques problèmes NP-complets

#### 12.1.1 Autour de SAT

##### Définition 12.1 (3-SAT)

*Donnée:* Un ensemble de variables  $\{x_1, \dots, x_n\}$  et une formule  $F = C_1 \wedge C_2 \wedge \dots \wedge C_\ell$  avec  $C_i = y_{i,1} \vee y_{i,2} \vee y_{i,3}$ , où pour tout  $i, j$ ,  $y_{i,j}$  est soit  $x_k$ , soit  $\neg x_k$  pour l'un des  $x_k$ .

*Réponse:* Décider si  $F$  est satisfiable : c'est-à-dire décider s'il existe  $x_1, \dots, x_n \in \{0, 1\}^n$  tel que  $F$  s'évalue en vraie pour cette valeur de ses variables  $x_1, \dots, x_n$ .

**Théorème 12.1** *Le problème 3-SAT est NP-complet.*

**Démonstration :** Notons tout d'abord que 3-SAT est bien dans NP. En effet, la donnée d'une affectation de valeurs de vérité aux variables constitue un certificat vérifiable en temps polynomial.

On va réduire SAT à 3-SAT. Soit  $F$  une formule SAT. Soit  $C$  une clause de  $F$ , par exemple  $C = x \vee y \vee z \vee u \vee v \vee w \vee t$ . On introduit de nouvelles variables  $a, b, c, d$  associées à cette clause, et on remplace  $C$  par la formule

$$(x \vee y \vee a) \wedge (\neg a \vee z \vee b) \wedge (\neg b \vee u \vee c) \wedge (\neg c \vee v \vee d) \wedge (\neg d \vee w \vee t).$$

Il est facile de vérifier qu'une assignation de  $x, y, z$  peut être complétée par une assignation de  $a, b, c, d$  de façon à rendre cette formule vraie si et seulement si  $C$  est vraie.

En appliquant cette construction à toutes les clauses de  $F$  et en prenant la conjonction des formules ainsi produites, on obtient ainsi une formule 3 – SAT dont la satisfaction de  $F'$  équivaut à celle de  $F$ .

Le temps de calcul se réduit à écrire les clauses, dont la longueur est polynomiale. Par conséquent, l'ensemble du processus de réduction se réalise donc bien en temps polynomial, et on a prouvé, à partir de SAT que 3 – SAT est NP-complet.  $\square$

**Remarque 12.1** *Le problème 2 – SAT, où l'on considérerait des clauses avec deux littéraux est lui dans P.*

**Définition 12.2 (NAESAT)**

**Donnée:** *Un ensemble de variables  $\{x_1, \dots, x_n\}$  et un ensemble de clauses  $y_{i,1} \vee \dots \vee y_{i,k_i}$ , où pour tout  $i, j$ ,  $y_{i,j}$  est soit  $x_k$ , soit  $\neg x_k$  pour l'un des  $x_k$ .*

**Réponse:** *Décider s'il existe une affectation des variables  $x_i \in \{0, 1\}$  de telle sorte que chaque clause contienne au moins un littéral vrai et au moins un littéral faux (c'est-à-dire, pour tout  $i$ , il y a un  $j$  et un  $k$  avec  $y_{i,j} = 1$  et  $y_{i,k} = 0$ ).*

**Théorème 12.2** *Le problème NAESAT est NP-complet.*

**Démonstration :** Le problème est dans NP car la donnée d'une affectation des variables est un certificat valide vérifiable aisément en temps polynomial.

On va réduire SAT à NAESAT. Soit  $F$  une formule de SAT sur les variables  $\{x_1, \dots, x_n\}$ . On ajoute une unique variable distincte  $z$  et on forme les clauses pour NAESAT en remplaçant chaque clause  $C_i = y_{i,1} \vee \dots \vee y_{i,k}$  de  $F$  en  $C'_i = y_{i,1} \vee \dots \vee y_{i,k} \vee z$ .

Cette transformation se réalise bien en temps polynomial.

Si l'instance donnée de SAT est satisfiable, la même affectation des variables tout en fixant pour  $z$  la valeur 0 fournit une affectation valide pour NAESAT.

Réciproquement, supposons que l'instance construite de NAESAT soit satisfiable. Si la valeur de vérité de  $z$  dans l'affectation correspondante est 0, alors les valeurs des variables  $x_i$  dans l'affectation donnent une affectation valide pour la formule  $F$  d'origine (pour l'instance de SAT). Si au contraire  $z$  vaut 1, on change toutes les valeurs de toutes les variables dans l'affectation. L'affectation reste valide pour NAESAT car au moins un littéral par clause dans l'affectation initiale valait 0, et vaut donc maintenant 1, tandis que  $z$  vaut 0. On a donc construit une affectation dans laquelle  $z$  vaut 0, et en vertu du cas précédent l'instance de SAT initiale est satisfaisable.

On a donc bien prouvé l'équivalence entre satisfaisabilité de  $F$  et l'instance correspondante pour NAESAT. Donc NAESAT est NP-complet.  $\square$

En utilisant la même réduction sur 3SAT, on prouve que NAE4SAT est NP-complet. On va utiliser cette remarque pour prouver :

**Corollaire 12.1** *NAE3SAT est NP-complet.*

**Démonstration :** On va réduire NAE4SAT à NAE3SAT. Soit  $C = x \vee y \vee z \vee t$  une clause à 4 littéraux. On introduit une nouvelle variable  $u_C$ , et on forme les deux

clauses  $C_1 = x \vee y \vee \neg u_C$  et  $C_2 = z \vee t \vee u_C$ . En faisant ainsi pour toutes les clauses, on construit une instance  $F'$  de NAE3SAT en temps polynomial.

Supposons que  $F'$  soit une instance positive de NAE3SAT, et considérons l'affectation des valeurs de vérité correspondante. Si  $u_C = 0$ , alors  $x$  ou  $y$  est 0, et  $z$  ou  $t$  est 1, donc  $x \vee y \vee z \vee t$  a au moins un littéral 1 et au moins un littéral 0; de même, si  $u_C = 1$ ; donc  $F$  est une instance positive de de NAE4SAT.

Inversement, si  $F$  est une instance positive de NAE4SAT, considérons l'affectation de valeurs de vérité correspondante. Dans  $x \vee y \vee z \vee t$ , si  $x$  et  $y$  sont tous deux à 1, on affecte  $u_C$  à 1; sinon si  $x$  et  $y$  sont tous les deux à 0, on affecte  $u_C$  à 0; sinon, on affecte à  $u_C$  la valeur de vérité adéquate pour la clause  $u_C \vee z \vee t$ . Cela produit une assignation démontrant que  $F'$  est une instance positive de NAE3SAT.

Là encore la réduction est polynomiale, et NAE3SAT est dans NP de façon triviale.  $\square$

### 12.1.2 Autour de STABLE

#### Définition 12.3 (STABLE)

*Donnée:* Un graphe  $G = (V, E)$  non-orienté et un entier  $k$ .

*Réponse:* Décider s'il existe  $V' \subset V$ , avec  $|V'| = k$ , tel que  $u, v \in V' \Rightarrow (u, v) \notin E$ .

**Théorème 12.3** *Le problème STABLE est NP-complet.*

**Démonstration :** STABLE est bien dans NP, car la donnée de  $V'$  est un certificat facilement vérifiable en temps polynomial.

On va réduire le problème 3 – SAT à STABLE, c'est-à-dire, étant donné une formule  $F$  du type 3 – SAT, construire en temps polynomial un graphe  $G$ , de sorte que l'existence d'un stable dans  $G$  soit équivalente à l'existence d'une affectation de valeurs de vérité qui satisfait  $F$ .

Soit  $F = \bigwedge_{1 \leq j \leq k} (x_{1j} \vee x_{2j} \vee x_{3j})$ . On construit un graphe  $G$  avec  $3k$  sommets, un pour chaque occurrence d'un littéral dans une clause.

- Pour chaque variable  $x_i$  de 3 – SAT,  $G$  possède une arête entre chaque sommet associé à un littéral  $x_i$  et chaque sommet associé à un littéral  $\neg x_i$  (ainsi un stable de  $G$  correspond à une affectation de valeurs de vérité à une partie des variables);
- Pour chaque clause  $C$ , on associe un triangle : par exemple pour une clause de  $F$  de la forme  $C = (x_1 \vee \neg x_2 \vee x_3)$ , alors  $G$  possède les arêtes  $(x_1, \neg x_2)$ ,  $(\neg x_2, x_3)$ ,  $(x_3, x_1)$  (ainsi un stable de  $G$  contient au plus un des trois sommets associés à la clause  $C$ ).

Soit  $k$  le nombre de clauses dans  $F$ . On démontre que  $F$  est satisfiable si et seulement si  $G$  possède un stable de taille  $k$ .

En effet, si  $F$  est satisfiable, on considère une assignation des variables satisfaisant  $F$ . Pour chaque clause  $C$  de  $F$ , on choisit  $y_C$  un littéral de  $C$  rendu vrai par l'assignation : cela définit  $k$  sommets formant un stable de  $G$ .

Réciproquement, si  $G$  a un stable de taille  $k$ , alors il a nécessairement un sommet dans chaque triangle. Ce sommet correspond à un littéral rendant la clause associée vraie, et forme une assignation des variables cohérente par construction des arêtes.

La réduction est clairement polynomiale.  $\square$

Deux problèmes classiques sont reliés à STABLE.

**Définition 12.4 (CLIQUE)**

*Donnée:* Un graphe  $G = (V, E)$  non-orienté et un entier  $k$ .

*Réponse:* Décider s'il existe  $V' \subset V$ , avec  $|V'| = k$ , tel que  $u, v \in V' \Rightarrow (u, v) \in E$ .

**Théorème 12.4** *Le problème CLIQUE est NP-complet.*

**Démonstration :** La réduction à partir de STABLE consiste à passer au complémentaire sur les arêtes. En effet, il suffit de prouver qu'un graphe  $G = (V, E)$  a un stable de taille  $k$  si et seulement si son graphe complémentaire  $\overline{G} = (V, \overline{E})$  (où  $\overline{E} = \{(u, v) | (u, v) \notin E\}$ ) a une clique de taille  $k$ .  $\square$

**Définition 12.5 (RECOUVREMENT DE SOMMETS)**

*Donnée:* Un graphe  $G = (V, E)$  non-orienté et un entier  $k$ .

*Réponse:* Décider s'il existe  $V' \subset V$ , avec  $|V'| \leq k$ , tel que toute arête de  $G$  ait au moins une extrémité dans  $V'$ .

**Théorème 12.5** *Le problème RECOUVREMENT DE SOMMETS est NP-complet.*

**Démonstration :** La réduction à partir de STABLE consiste à passer au complémentaire sur les sommets.  $\square$

**Définition 12.6 (COUPURE MAXIMALE)**

*Donnée:* Un graphe  $G = (V, E)$  non-orienté et un entier  $k$ .

*Réponse:* Décider s'il existe une partition  $V = V_1 \cup V_2$  telle que le nombre d'arêtes entre  $V_1$  et  $V_2$  soit au moins  $k$ .

**Théorème 12.6** *Le problème COUPURE MAXIMALE est NP-complet.*

**Démonstration :** On réduit NAE3SAT à COUPURE MAXIMALE. Supposons donc donnée une instance de NAE3SAT, dans laquelle on peut supposer sans perte de généralité qu'une clause ne contient pas simultanément une variable et son complémentaire. Quitte à remplacer  $u \vee v$  par  $((u \vee v \vee w) \wedge (u \vee v \vee \neg w))$ , on peut aussi supposer que chaque clause contient exactement 3 littéraux. Enfin, si l'on a deux clauses  $(u \vee v \vee w)$  et  $(u \vee v \vee z)$ , on peut, en introduisant deux variables  $t_1$  et  $t_2$  et en procédant comme pour réduire NAE4SAT à NAE3SAT, réécrire ces deux clauses comme  $(u \vee t_1 \vee t_2) \wedge (v \vee w \vee \neg t_1) \wedge (v \vee z \vee \neg t_2)$ . Bref, on peut donc supposer que deux clauses données ont au plus une variable en commun.

On note  $x_1, \dots, x_n$  les variables de la formule  $F$ .

On va construire un graphe  $G = (V, E)$  de la façon suivante :  $G$  possède  $2n$  sommets où pour chaque variable  $u$  de  $F$ , correspondent deux sommets  $u$  et  $\neg u$ .  $G$  possède une arête entre chaque couple de sommets  $(u, v)$  tels que  $u$  et  $v$  apparaissent dans la même clause, et une arête entre les sommets  $u$  et  $\neg u$  pour toute variable  $u$ .

Les réductions dans le premier paragraphe de la preuve permettent de voir qu'à chaque clause correspond un triangle et que deux de ces triangles ont des arêtes distinctes.

Si on note  $n$  le nombre de variables, et  $m$  le nombre de clauses, le graphe  $G$  a donc  $2n$  sommets et  $3m + n$  arêtes. Il est alors facile de voir que le nombre d'arêtes dans une coupure correspondant à une affectation NAE3SAT valide est  $2m + n$  : l'arête entre  $u$  et  $\neg u$  pour chaque variable  $u$ , et deux des arêtes du triangle  $uvw$  pour chaque clause  $u \vee v \vee w$ .

Inversement, toute coupure de  $G$  a au plus  $(2m + n)$  arêtes, car une coupure ne peut inclure que deux arêtes par triangle associé à une clause. Par conséquent, une coupure de valeur  $2m + n$  fournit immédiatement une affectation NAE3SAT valide.

En d'autres termes, résoudre l'instance de NAE3SAT revient à résoudre COUPURE MAXIMALE sur  $(G, 2m + n)$ .

La réduction étant polynomiale, et puisque COUPUREMAXIMALE est dans NP car la donnée de  $V_1$  est un certificat valide vérifiable en temps polynomial.  $\square$

### 12.1.3 Autour de CIRCUIT HAMILTONIEN

Le problème CIRCUIT HAMILTONIEN est souvent à la base de propriétés liées aux chemins dans les graphes (nous reprenons la définition de ce problème, même s'il a déjà été défini dans le chapitre précédent).

#### Définition 12.7 (CIRCUIT HAMILTONIEN)

**Donnée:** Un graphe  $G = (V, E)$  (non-orienté).

**Réponse:** Décider s'il existe un circuit hamiltonien, c'est-à-dire un chemin de  $G$  passant une fois et une seule par chacun des sommets et revenant à son point de départ.

**Théorème 12.7** *Le problème CIRCUIT HAMILTONIEN est NP-complet.*

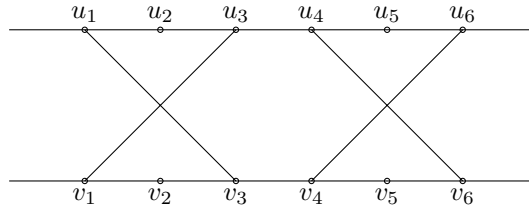
**Démonstration :** On va démontrer ce fait en réduisant RECOUVREMENT DE SOMMETS à ce problème.

La technique consiste à construire, partant d'une instance de RECOUVREMENT DE SOMMETS, un graphe dans lequel chaque arête initiale sera remplacée par un "motif" admettant exactement deux chemins hamiltoniens, c'est-à-dire des chemins visitant une fois et une seule chaque sommet. L'un de ces deux chemins correspondra au cas où le sommet correspondant appartient à la couverture, l'autre au cas où il n'y appartient pas.

Il reste ensuite à préciser comment recoller ces différents motifs pour qu'un circuit hamiltonien global corresponde exactement à une réunion de chemins hamiltoniens à travers chaque motif, et pour assurer qu'on obtient bien le bon résultat au problème RECOUVREMENT DE SOMMETS initial en résolvant CIRCUIT HAMILTONIEN dans ce graphe.

Notons  $(G = (V, E), k)$  l'instance de RECOUVREMENT DE SOMMETS étudiée.

Le motif que nous allons utiliser est le suivant :



Pour obtenir un parcours de ce motif traversant une fois et une seule chaque sommet, seules deux solutions sont possibles : soit un passage en deux fois, une fois  $u_1 u_2 u_3 u_4 u_5 u_6$  puis ultérieurement  $v_1 v_2 v_3 v_4 v_5 v_6$  (dans un sens ou dans l'autre); ou alors un passage en une seule fois  $u_1 u_2 u_3 v_1 v_2 v_3 v_4 v_5 v_6 u_4 u_5 u_6$  (ou le même en inversant les  $u$  et les  $v$ ).

À chaque arête  $(u, v)$  du graphe de départ, on associe un motif de ce type, en faisant correspondre le sommet  $u$  au côté  $u$  et le sommet  $v$  au côté  $v$ . On raccorde ensuite entre eux bout à bout tous les côtés de tous les motifs correspondant à un même sommet; on forme donc une chaîne associée à un sommet donné, avec encore deux "sorties" libres. On raccorde alors chacune de ces deux sorties à  $k$  nouveaux sommets  $s_1, \dots, s_k$ . On note le graphe ainsi construit  $H$  dans la suite.

Supposons maintenant donné un recouvrement du graphe initial de taille  $k$ , dont les sommets sont  $\{g_1, \dots, g_k\}$ . On peut alors construire un circuit hamiltonien de  $H$  de la façon suivante :

- partir de  $s_1$ ;
- parcourir la chaîne  $g_1$  de la façon suivante. Quand on traverse une arête  $(g_1, h)$ , si  $h$  est aussi dans la couverture, on traverse simplement le côté  $g_1$ , sinon, on parcourt les deux côtés simultanément;
- une fois la chaîne  $g_1$  finie, on revient en  $s_2$  et on repart par  $g_2$  et ainsi de suite.

Il est clair que tous les sommets  $s_k$  sont atteints une fois et une seule.

Considérons un sommet  $h$  du motif correspondant à une arête  $(u, v)$ . On peut toujours supposer que  $u$  est dans la couverture, disons  $u = g_1$ . Il s'ensuit que si  $h$  est du côté de  $u$ ,  $h$  sera atteint une fois au moins. On voit en vertu du second item qu'il ne sera plus atteint dans la suite. Si  $h$  est du côté de  $v$ , et que  $v$  n'est pas dans la couverture,  $h$  est parcouru lors du parcours de  $u$ . Si  $v = g_i$ ,  $h$  est parcouru lors du parcours de la chaîne correspondant à  $g_i$  et à ce moment-là seulement. On a donc bien un circuit hamiltonien.

Réciproquement, supposons que l'on dispose d'un circuit hamiltonien de  $H$ . La construction de notre motif impose que venant d'un sommet  $s_i$ , on traverse entièrement une chaîne  $u$  puis l'on passe à un autre des sommets  $s_j$ . On traverse ainsi  $k$  chaînes; les  $k$  sommets correspondants forment alors une couverture. En effet, si  $(u, v)$  est une arête, le motif correspondant est parcouru par le chemin hamiltonien; or il ne peut l'être que lors du parcours d'une boucle correspondant à une des deux extrémités de l'arête.

Enfin, la réduction est trivialement polynomiale. CIRCUIT HAMILTONIEN est donc bien NP-complet.  $\square$

Comme dans la section précédente, on peut alors en déduire la NP-complétude de nombreuses variantes.

**Définition 12.8** (VOYAGEUR DE COMMERCE)

**Donnée:** Un couple  $(n, M)$ , où  $M$  est une matrice  $n \times n$  d'entiers et un entier  $k$ .

**Réponse:** Décider s'il existe une permutation  $\pi$  de  $[1, 2, \dots, n]$  telle que

$$\sum_{1 \leq i \leq n} M_{\pi(i)\pi(i+1)} \leq k.$$

**Corollaire 12.2** *Le problème VOYAGEUR DE COMMERCE est NP-complet.*

Ce problème porte ce nom, car on peut voir cela comme l'établissement de la tournée d'un voyageur de commerce devant visiter  $n$  villes, dont les distances sont données par la matrice  $M$  de façon à faire moins de  $k$  kilomètres.

**Démonstration :**

On réduit CIRCUIT HAMILTONIEN à VOYAGEUR DE COMMERCE. Pour ce faire, étant donné un graphe  $G = (V, E)$ , on pose  $V = \{x_1, \dots, x_n\}$ . On considère alors la matrice  $M$   $n \times n$  d'entiers telle que

$$M_{i,j} = \begin{cases} 1 & \text{si } (i, j) \in E; \\ 2 & \text{sinon.} \end{cases}$$

Montrons alors que CIRCUIT HAMILTONIEN  $(V, E)$  si et seulement si VOYAGEUR DE COMMERCE  $(n, M, n)$ .

S'il existe un circuit hamiltonien dans  $G$  on peut en effet construire la permutation  $\pi$  comme décrivant l'ordre de parcours des sommets du graphe  $G$  : par construction, la somme des distances des arêtes sur ce circuit vaudra  $n$ .

Inversement, étant donnée une permutation avec cette propriété, le fait que les  $n-1$  termes de la somme soient au moins égaux à 1 implique qu'ils sont tous égaux à 1, et donc que les arêtes  $(\pi(i), \pi(i+1))$  existent dans le graphe  $G$  : on a donc bien un circuit hamiltonien.  $\square$

**Définition 12.9** (CIRCUIT LE PLUS LONG)

**Donnée:** Un graphe  $G = (V, E)$  non-orienté, avec des distances sur chaque arête, un entier  $r$ .

**Réponse:** Décider s'il existe un circuit de  $G$  ne passant pas deux fois par le même sommet dont la longueur est  $\geq r$ .

**Corollaire 12.3** *Le problème CIRCUIT LE PLUS LONG est NP-complet.*

**Démonstration :** On construit une réduction à partir de CIRCUIT HAMILTONIEN : pour cela, à un graphe  $G$  pour CIRCUIT HAMILTONIEN, on associe à chaque arête le poids 1. La recherche d'un cycle hamiltonien est alors trivialement identique à la recherche d'un circuit de longueur  $\geq n$  dans le graphe.  $\square$

**12.1.4 Autour de 3-COLORABILITE**

Nous avons déjà défini ce que nous appelions un (*bon*) coloriage de graphes dans le chapitre précédent. Le plus petit entier  $k$  tel qu'un graphe soit (bien) coloriable est appelé le *nombre chromatique* du graphe.

Il est connu que les graphes planaires sont toujours coloriables avec 4 couleurs.

**Définition 12.10** (3-COLORABILITE)

*Donnée:* Un graphe  $G = (V, E)$  non-orienté.

*Réponse:* Décider s'il existe un coloriage du graphe utilisant au plus 3 couleurs.

**Théorème 12.8** Le problème 3-COLORABILITE est NP-complet.

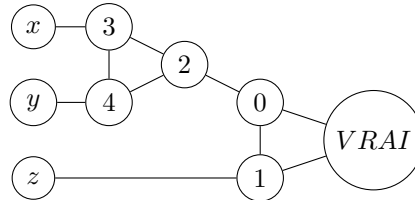
**Démonstration :** 3-COLORABILITE est dans NP, car la donnée des sommets colorés par chacune des 3 couleurs constitue un certificat vérifiable en temps polynomial.

On va réduire 3-SAT à 3-COLORABILITE. On se donne donc une conjonction de clauses à 3 littéraux, et il nous faut à partir de là construire un graphe. Comme dans les autres réductions de 3-SAT, il faut parvenir à traduire deux contraintes : une variable peut prendre la valeur 0 ou 1 d'une part, et les règles d'évaluation d'une clause d'autre part.

On construit un graphe ayant  $3 + 2n + 5m$  sommets, les trois premiers sont notés *VRAI*, *FAUX*, *NSP*. Ces trois sommets sont reliés deux à deux en triangle, de sorte qu'ils doivent être tous trois de couleurs différentes. On appellera les couleurs correspondantes *VRAI*, *FAUX*, *NSP*.

On associe un sommet à chaque variable et au complémentaire de chaque variable. Pour assurer qu'une variable prenne la valeur *VRAI* ou *FAUX*, pour chaque variable  $x_i$  on construit un triangle dont les sommets sont  $x_i$ ,  $\neg x_i$ , et *NSP*. Cela impose que soit  $\text{couleur}(x_i) = \text{VRAI}$  et  $\text{couleur}(\neg x_i) = \text{FAUX}$ , ou  $\text{couleur}(x_i) = \text{FAUX}$  et  $\text{couleur}(\neg x_i) = \text{VRAI}$ .

Il nous reste donc à encoder les règles d'évaluation d'une clause. Pour ce faire, on introduit le sous-graphe suivant, qui correspond à une clause  $x \vee y \vee z$  :



Il est facile de voir que si ce motif (où les trois sommets distingués et les triangles construits sur les variables sont implicites) est 3-coloriable, alors les sommets 0 et 1 sont *FAUX* et *NSP*. Si 1 est *FAUX*, comme un sommet correspondant à une variable doit être *VRAI* ou *FAUX*, on a  $\text{couleur}(z) = \text{VRAI}$ . Si 0 est *FAUX*, alors 2 ne peut pas être *FAUX*, donc 3 ou 4 l'est, et la variable correspondante est colorisée *VRAI*.

Réciproquement, si l'une des variables est vraie, on peut facilement construire une 3-coloration du motif.

Considérons alors le graphe formé des trois sommets distingués, des triangles formés sur les variables, et des motifs donnés. Si ce graphe est 3-coloriable, alors en particulier tout sous-graphe est coloriable. Les triangles de variables sont en particulier coloriables. À partir d'une 3-coloration du graphe, on construit une affectation de valeurs de vérité en mettant à 1 toutes les variables colorées par *VRAI*. Cette affectation est cohérente (une variable et son complémentaire ont bien une valeur opposée)



et au moins une variable par clause est à 1, en vertu des propriétés du motif ci-dessus. Inversement, étant donné une affectation de valeurs de vérité, il est aisé de déduire une 3-coloration du graphe.

L'existence d'une 3-coloration du graphe est donc équivalente à la satisfaisabilité de la formule initiale.

La réduction est manifestement polynomiale ; on a donc bien prouvé que 3-SAT se réduisait à 3-COLORABILITE ; ce dernier est donc bien NP-complet.  $\square$

### 12.1.5 Autour de SOMME DE SOUS-ENSEMBLE

**Définition 12.11** (SOMME DE SOUS-ENSEMBLE)

*Donnée:* Un ensemble fini d'entiers  $E$  et un entier  $t$ .

*Réponse:* Décider s'il existe  $E' \subset E$  tel que  $\sum_{x \in E'} x = t$ .

**Théorème 12.9** *Le problème SOMME DE SOUS-ENSEMBLE est NP-complet.*

**Démonstration :** Le fait que SOMME DE SOUS-ENSEMBLE est dans NP vient du fait que la donnée de  $E'$  constitue un certificat vérifiable en temps polynomial.

On va maintenant réduire une version de RECOUVREMENT DE SOMMETS dans laquelle on cherche à construire une couverture de taille exactement  $k$  à SOMME DE SOUS-ENSEMBLE. Il est facile de voir que cette version de RECOUVREMENT DE SOMMETS est essentiellement équivalente à la version classique, car il suffit de décider pour chaque  $l \leq l_0$  s'il existe une couverture de taille  $l$  pour savoir s'il existe une couverture de taille au plus  $l_0$ .

On se donne un graphe  $G = (V, E)$  dans lequel on souhaite construire un recouvrement de sommets de taille  $k$ . On numérote les sommets et arêtes ; soit  $B = (b_{ij})$  la matrice d'incidence sommets-arêtes, c'est-à-dire que  $b_{ij} = 1$  si l'arête  $i$  est incidente au sommet  $j$ ,  $b_{ij} = 0$  sinon.

On va construire un ensemble  $F$  d'entiers, tel que chaque sommet corresponde à un entier de  $F$ . On veut pouvoir lire sur la somme des éléments d'un sous-ensemble les arêtes dont un des sommets au moins appartient au sous-ensemble, de façon à ce qu'on ait bien une couverture de sommets si et seulement si c'est le cas pour toutes les arêtes.

Cela peut se faire en exprimant les entiers dans une base  $b$  à fixer ultérieurement, et en faisant correspondre au sommet  $s_j$  l'entier  $\sum b_{ij}b^i$ . On veut aussi pouvoir compter le nombre de sommets présents dans la couverture, qui doit être égal à  $k$  ; pour ce faire, il suffit d'ajouter un terme  $b^n$ , si les arêtes sont numérotées de 0 à  $n - 1$ .

Il subsiste une difficulté : pour qu'un sous-ensemble de somme  $t = \sum_{i=0}^n a_i b^i$  corresponde à une couverture de sommets, le choix que l'on vient juste de faire impose que (noter que tous les  $a_i$  sauf  $a_n$  valent au plus 2, car une arête est incidente à deux sommets exactement) :

- $a_n = k$  ;
- $a_i \geq 1$ .

Les spécifications du problème ne permettent pas de prendre en compte le second item. On peut contourner le problème en ajoutant à  $F$  les entiers  $b^i$  correspondant à l'arête  $i$ .

On peut alors choisir comme but

$$t = kb^n + \sum_{i=0}^{n-1} 2b^i. \quad (12.1)$$

À ce stade, on va imposer que la base  $b$  soit au moins égale à 4; comme dans toute somme d'éléments de  $F$  il y a au plus trois termes  $b^i$  pour  $i < n$ , cela entraîne qu'aucune retenue ne peut se produire dans l'addition (on permet toujours au coefficient de  $b^n$  d'excéder  $b - 1$ ), et donc que l'on peut lire sur l'équation (12.1) le nombre d'occurrences du terme  $b^i$ .

Pour tout sous-ensemble  $F'$  de somme  $t$ , on a donc :

- il y a  $k$  termes correspondant à des sommets dans  $F'$  (cf. le terme en  $b^n$  pour chaque arête  $i$ );
- $F'$  contient au moins un terme correspondant à une des deux extrémités de  $i$ ; en effet,  $t$  contenant 2 termes  $b^i$ , il peut contenir soit les deux extrémités de l'arête, soit l'entier correspondant à l'arête et celui correspondant à l'une des deux extrémités.

Inversement, étant donnée une couverture de sommets, on construit l'ensemble  $F'$  en prenant les sommets de la couverture, et les arêtes dont un seul des deux sommets est dans la couverture.

On a donc bien réduit RECOUVREMENT DE SOMMETS à SOMME DE SOUS-ENSEMBLE.

Il est facile de voir que la réduction s'effectue en temps polynomial, et donc que l'on a bien prouvé le théorème.  $\square$

On peut en déduire :

**Définition 12.12 (SAC A DOS)**

*Donnée:* Un ensemble de poids  $a_1, \dots, a_n$ , un ensemble de valeurs  $v_1, \dots, v_n$ , un poids limite  $A$ , et un entier  $V$ .

*Réponse:* Décider s'il existe  $E' \subset \{1, 2, \dots, n\}$  tel que  $\sum_{i \in E'} a_i \leq A$  et  $\sum_{i \in E'} v_i \geq V$ .

**Corollaire 12.4** *Le problème SAC A DOS est NP-complet.*

**Démonstration :** A partir de SOMME DE SOUS-ENSEMBLE : étant donnée  $E = \{e_1, \dots, e_n\}$  et  $t$  une instance de SOMME DE SOUS-ENSEMBLE, on considère  $v_i = a_i = e_i$ , et  $V = A = t$ .  $\square$

**Définition 12.13 (PARTITION)**

*Donnée:* Un ensemble fini d'entiers  $E$ .

*Réponse:* Décider s'il existe  $E' \subset E$  tel que  $\sum_{x \in E'} x = \sum_{x \notin E'} x$ .

**Théorème 12.10** *Le problème PARTITION est NP-complet.*

**Démonstration :** On va réduire SOMME DE SOUS-ENSEMBLE à PARTITION. Soit  $(E, t)$  une instance de SOMME DE SOUS-ENSEMBLE. On pose  $S = \sum_{x \in E} x$ .

Quitte à changer  $t$  en  $S - t$  (ce qui revient à changer l'ensemble obtenu en son complémentaire), on peut supposer que  $2t \leq S$ .

L'idée naturelle consisterait à ajouter l'élément  $u = S - 2t$  à  $E$ ; le résultat de partition serait alors deux sous-ensembles ( $A'$  et son complémentaire) de somme  $S - t$ ; l'un des deux contient l'élément  $S - 2t$ , donc en enlevant ce dernier, on trouve un ensemble de termes de  $E$  de somme  $t$ . Malheureusement, cette technique échoue si  $S - 2t$  est déjà dans  $E$ .

Au lieu de cela, on prend le nombre  $X = 2S$  et  $X' = S + 2t$ , et on applique PARTITION à  $E' = E \cup \{X, X'\}$ . Il existe une partition de  $E'$  si et seulement si il existe un sous-ensemble de  $E$  de somme  $t$ . En effet, s'il existe une partition de  $E'$ , il existe deux sous-ensembles complémentaires de somme  $2S + t$ . Chacun des deux sous-ensembles doit contenir soit  $X$ , soit  $X'$ , car sinon sa somme ne peut excéder  $2S + t$ ; donc un des deux ensembles contient  $X$  et non  $X'$ , et on obtient en enlevant  $X$  un sous-ensemble  $F$  de  $E$  de taille  $t$ . Réciproquement, étant donné un tel  $F$ ,  $(F \cup \{X\}, E - F \cup \{X'\})$  constitue une partition de  $E'$ . On a donc bien réduit SOMME DE SOUS-ENSEMBLE à PARTITION.

Reste à justifier que la réduction est bien polynomiale. L'essentiel de la réduction est le calcul de  $A$  et  $A'$ , qui est bien polynomial en la taille des entrées (l'addition de  $k$  nombres de  $n$  bits se fait en temps  $\mathcal{O}(k \log n)$ ).  $\square$

## 12.2 Exercices

### 12.2.1 Variantes polynomiales

**Exercice 12.1.** Un graphe  $G = (V, E)$  est dit Eulérien s'il existe un cycle empruntant exactement une fois chaque arête du graphe  $G$ .

Montrer qu'un graphe connexe est Eulérien si et seulement si chacun de ses sommets a un degré pair.

Proposer un algorithme polynomial qui détermine si le graphe est Eulérien.

### Exercice 12.2.

On s'intéresse à 2SAT.

1. Proposer une valuation  $t$  qui satisfait  $\phi_1(u_1, u_2, u_3) = (u_2 \vee \neg u_3) \wedge (\neg u_2 \vee u_3) \wedge (u_2 \vee \neg u_1)$ .
2. Que se passe-t-il pour  $\phi_2(u_1, u_2) = (u_2 \vee u_3) \wedge (\neg u_2 \vee u_3) \wedge (\neg u_3 \vee u_1) \wedge (\neg u_3 \vee \neg u_1)$  ?
3. Montrer que  $u \vee v = (\neg u \Rightarrow v) \wedge (\neg v \Rightarrow u)$ .  
A partir d'une instance de 2SAT nous construisons un graphe orienté  $G_\phi = (V, E)$  avec
  - un sommet par littéral
  - et un arc par implication (en transformant chaque clause par deux implications)
4. Dessiner les graphes  $G_{\phi_1}$  et  $G_{\phi_2}$ .

5. Montrer qu'il existe une variable  $u$  telle que  $G_\phi$  contient un cycle entre  $u$  vers  $\neg u$  dans  $G$ , si et seulement si  $\phi$  n'est pas satisfiable.
6. Montrer que 2-SAT peut se résoudre en temps polynomial.

**Exercice 12.3.** [Chevaliers de la table ronde] Etant donnés  $n$  chevaliers, et connaissant toutes les paires de féroces ennemis parmi eux, est-il possible de les placer autour d'une table circulaire de telle sorte qu'aucune paire de féroces ennemis ne soit côte à côte ?

### 12.2.2 NP-complétude

#### Exercice 12.4.

**Définition :** une chaîne hamiltonienne est une chaîne qui passe une fois et une seule par chaque sommet du graphe.

Prouver que le problème suivant est NP-complet.

CHAÎNE HAMILTONIENNE

Données : un graphe non-orienté  $G$  de  $n$  sommets, deux sommets  $u$  et  $v$  distincts de  $G$ .

Question :  $G$  contient-il une chaîne hamiltonienne dont ses extrémités sont les sommets  $u$  et  $v$  ?

**Exercice 12.5.** Prouver que le problème suivant est NP-complet.

CHAÎNE

Données : un graphe non-orienté  $G$  de  $n$  sommets, deux sommets  $u$  et  $v$  distincts de  $G$ .

Question :  $G$  contient-il une chaîne de longueur  $n/2$  entre  $u$  et  $v$  ?

**Exercice 12.6.** Prouver que le problème suivant est NP-complet.

**Problème** ARBRE

Données : Un graphe  $G = (V, E)$ , et un entier  $k$ .

Question : Existe-t-il un arbre couvrant tous les sommets de  $G$  ayant moins de  $k$  feuilles ?

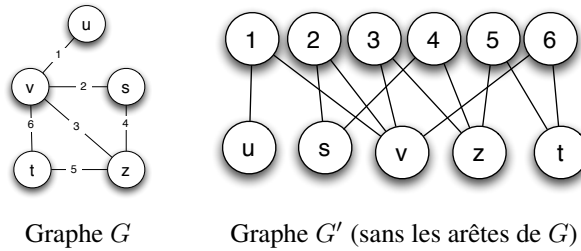
**Exercice 12.7.** Soit  $G = (V, E)$  un graphe.

- Une couverture de sommets  $S$  du graphe  $G$  est un sous-ensemble de sommets tel que toutes les arêtes de  $G$  sont incidentes à au moins un sommet de  $S$ .
- Un ensemble dominant  $C$  du graphe  $G$  est un sous-ensemble de sommets tel que tout sommet est soit dans  $C$  soit voisin d'un sommet de  $C$ .

Soit  $G = (V, E)$  un graphe connexe. Nous allons construire un graphe  $G' = (V', E')$  à partir de  $G$  tel que

- $V' = V \cup E$ ;
- $E' = E \cup \{(v, a) | v \in V, a \in E, v \text{ est extrémité de l'arête } a \text{ dans } G\}$

1. Montrer que si  $S$  est une couverture de sommets du graphe  $G$ , alors  $S$  est un ensemble dominant de  $G'$ .



2. Montrer que si  $S'$  est un ensemble dominant de  $G'$ , alors il existe une couverture  $S \subseteq V$  du graphe  $G$  et de cardinalité inférieure ou égale à  $S'$ .
3. Exprimer le problème de minimisation de l'ensemble dominant sous forme de problème de décision.
4. Montrer que ce problème est dans NP.
5. Montrer que ce problème est dans NP-complet.

**Exercice 12.8.** Nous allons nous concentrer sur le problème du  $k$ -centre : étant donné un ensemble de villes dont les distances sont spécifiées, choisir  $k$  villes afin d'installer des entrepôts de façon à minimiser la distance maximale d'une ville à l'entrepôt le plus proche. Un tel ensemble de  $k$  villes est appelé  $k$ -centre.

Le problème associé de décision est le suivant :

Problème  $k$ -CENTRE

**Données :** un graphe complet  $K = (V, E)$  muni d'une fonction de poids  $w$  sur les arêtes, et des entiers strictement positifs  $k$  et  $b$ .

**Question :** Existe-il un ensemble  $S$  de sommets tel que  $|S| = k$  et tel que tout sommet  $v$  de  $V$  satisfait la condition suivante

$$\max\{w(v, u) : u \in S\} \leq b$$

1. Montrer que  $k$ -CENTRE est dans NP.
2. Montrer que  $k$ -CENTRE est NP-complet sachant que DOMINANT est NP-complet.

## 12.3 Notes bibliographiques

**Lectures conseillées** Le livre [Garey and Johnson, 1979] recense plus de 300 problèmes NP-complets et est d'une lecture assez agréable. On peut trouver plusieurs mises à jour de cette liste sur le web.

**Bibliographie** Ce chapitre est repris du polycopié [Cori et al., 2010] du cours INF550 de l'école polytechnique.

# Index

- $(A_g, r, A_d)$ , voir arbre binaire  
 $(V, E)$ , voir graphe  
 $(q, u, v)$ , voir configuration d'une machine de Turing  
 $\cdot$ , voir concaténation  
 $A^c$ , voir complémentaire  
 $F(G/p)$ , voir substitution  
 $L(M)$ , voir langage accepté par une machine de Turing  
 $\Leftrightarrow$ , voir double implication  
 $\Rightarrow$ , voir définition inductive / différentes notations d'une, voir implication  
 $\Sigma$ , voir alphabet  
 $\Sigma^*$ , voir ensemble des mots sur un alphabet  
 $\cap$ , voir intersection de deux ensembles  
 $\cup$ , voir union de deux ensembles  
 $\epsilon$ , voir mot vide  
 $\equiv$ , voir équivalence entre formules, voir équivalence entre problèmes  
 $\exists$ , voir quantificateur  
 $\forall$ , voir quantificateur  
 $\leq_m$ , voir réduction  
 $|w|$ , voir longueur d'un mot  
 $\leq$ , voir réduction  
 $\mathcal{P}(E)$ , voir parties d'un ensemble  
 $\models$ , voir conséquence sémantique  
 $\neg$ , voir négation  
 $\not\models$ , voir conséquence sémantique  
 $\subset$ , voir partie d'un ensemble  
 $\times$ , voir produit cartésien de deux ensembles  
 $\vdash$ , voir démonstration, voir relation successeur entre configurations d'une machines de Turing  
 $\vee$ , voir disjonction  
 $\wedge$ , voir conjonction  
 $uqv$ , voir configuration d'une machine de Turing  
 $\langle\langle M \rangle, w\rangle$ , voir codage d'une paire  
 $\langle M \rangle$ , voir codage d'une machine de Turing  
 $\langle m \rangle$ , voir codage  
 $\langle \phi \rangle$ , voir codage d'une formule  
 $\langle w_1, w_2 \rangle$ , voir codage d'une paire  
 2-SAT, 2  
 3-COLORABILITE, 7  
 3-SAT, 1  
*Arith*, voir expressions arithmétiques  
*Arith'*, voir expressions arithmétiques parenthésées  
 circuit  
     hamiltonien d'un graphe, 5  
 CIRCUIT HAMILTONIEN, 5  
 CIRCUIT LE PLUS LONG, 7  
 CLIQUE, 4  
 codage  
     notation, voir  $\langle \cdot \rangle$   
     d'une formule  
         notation, voir  $\langle \phi \rangle$   
     d'une machine de Turing  
         notation, voir  $\langle M \rangle$   
     d'une paire  
         notation, voir  $\langle\langle M \rangle, w\rangle$ , voir  $\langle w_1, w_2 \rangle$   
 coloriage d'un graphe, 7  
 complémentaire  
     notation, voir  $A^c$   
 du problème de l'arrêt des machines de Turing  
     notation, voir  $\overline{\text{HALTING - PROBLEM}}$   
 complétude  
     RE-complétude, voir RE-complet

- concaténation  
*notation, voir .*
- configuration d'une machine de Turing  
*notation, voir  $(q, u, v)$ , voir  $uq^v$*
- COUPURE MAXIMALE, 4
- équivalence  
entre problèmes  
*notation, voir  $\equiv$*
- graphe  
*notation, voir  $(V, E)$*
- HALTING – PROBLEM, *voir* problème de l'arrêt des machines de Turing
- HALTING – PROBLEM, *voir* complémentaire du problème de l'arrêt d'une machine de Turing
- intersection de deux ensembles  
*notation, voir  $\cap$*
- langage  
accepté par une machine de Turing  
*notation, voir  $L(M)$*
- longueur d'un mot  
*notation, voir  $|w|$*
- mot  
vide  
*notation, voir  $\epsilon$*
- NAE3SAT, 2
- NAESAT, 2
- nombre chromatique d'un graphe, 7
- NP, 1
- NP-complétude, 1–5, 7–10
- partie  
d'un ensemble  
*notation, voir  $\subset$*
- parties  
d'un ensemble  
*notation, voir  $\mathcal{P}(E)$*
- PARTITION, 10
- problème  
de l'arrêt des machines de Turing  
*notation, voir* HALTING – PROBLEM
- produit cartésien  
de deux ensembles  
*notation, voir  $\times$*
- R, *voir* décidable
- RE, *voir* récursivement énumérable
- RECOUVREMENT DE SOMMETS, 4
- récursivement énumérable  
*notation, voir* RE
- réduction  
*notation, voir  $\leq$ , voir  $\leq_m$*
- relation successeur entre configurations d'une machine de Turing  
*notation, voir  $\vdash$*
- SAC A DOS, 10
- SAT, 1, 2
- satisfaction  
d'une formule, 1
- SOMME DE SOUS-ENSEMBLE, 9
- STABLE, 3
- union de deux ensembles  
*notation, voir  $\cup$*
- VOYAGEUR DE COMMERCE, 6





# Bibliographie

- [Cori et al., 2010] Cori, R., Hanrot, G., Kenyon, C., and Steyaert, J.-M. (2010). Conception et analyse d'algorithmes. Cours de l'Ecole Polytechnique.
- [Garey and Johnson, 1979] Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability*. W. H. Freeman and Co.