La NP-complétude et les entiers

Johanne Cohen

PRISM/CNRS, Versailles, France.

Références

1. *Introduction to Algorithms*, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, 2009.

2. *Algorithm Design*, Jon Kleinberg, Eva Tardos, Addison-Wesley, 2006.

3. *Computers and Intractability : A Guide to the Theory of NP-Completeness*, M. R. Garey, D. S. Johnson, 1976.

Plan

Retour sur l'épisode précédent

Programmation dynamique

- Suite de Fibonacci
- Problème SOMME DE SOUS-ENSEMBLE

Faiblement/Fortement NP-complet

Annexe

Stratégie pour prouver la NP-complétude

Pour prouver la NP-complétude d'un problème A, il suffit de prouver :

1. qu'il admet un vérificateur polynomial ;
2. que $B \leq A$ avec un problème connu NP-complet B

Somme de sous-ensembles

SOMME DE SOUS-ENSEMBLE
Données : un ensemble fini d'entiers $A = \{a_1, a_2, \ldots, a_\ell\}$ et un entier $t \in \mathbb{N}$
Question : Existe-t-il un sous-ensemble $A' \subseteq A$ tel que $\sum_{a \in A'} a = t$?

Théorème

Le problème SOMME DE SOUS-ENSEMBLE est NP-complet.

Preuve :

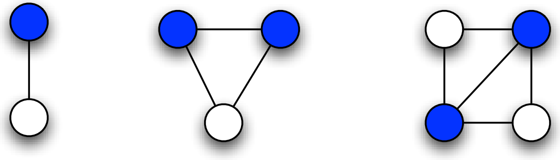
- SOMME DE SOUS-ENSEMBLE est dans NP.
- Couverture sommet $\leq$ SOMME DE SOUS-ENSEMBLE

Couverture sommet

Couverture de sommets (VC)
Données : un graphe non-orienté $G = (V, E)$ et un entier k.
Question : G contient-il une *couverture de sommets* $\mathcal{S}$ de cardinalité au plus k ?

Théorème

Le problème Couverture de sommets est NP-complet.



Les sommets en bleu font partie de la couverture sommet

VC $\leq$ SOMME DE SOUS-ENSEMBLE (1/2)

- On se donne un graphe $G = (V, E)$ et un entier k.

  Il nous faut construire un ensemble d'entiers A à partir de G.

Voir un exemple

VC $\leq$ SOMME DE SOUS-ENSEMBLE (1/2)

- On se donne un graphe $G = (V, E)$ et un entier k.

  Il nous faut construire un ensemble d'entiers A à partir de G.

  Il faut parvenir à traduire deux contraintes : un sommet peut appartenir ou ne pas appartenir à la couverture et le sous-ensemble de sommets couvrent bien toutes les arêtes.

▸ Voir un exemple

VC $\leq$ SOMME DE SOUS-ENSEMBLE (1/2)

- On se donne un graphe $G = (V, E)$ et un entier k.

  Il nous faut construire un ensemble d'entiers A à partir de G.

  Il faut parvenir à traduire deux contraintes : un sommet peut appartenir ou ne pas appartenir à la couverture et le sous-ensemble de sommets couvrent bien toutes les arêtes.

- Pour cela,
  - on fixe une numérotation des sommets et des arêtes.
  - pour chaque couple (arête,sommet), $b_{ij} = 1$ si l'arête i est incidente au sommet j, sinon $b_{ij} = 0$

Voir un exemple

VC $\leq$ SOMME DE SOUS-ENSEMBLE (1/2)

- On se donne un graphe $G = (V, E)$ et un entier k.

  Il nous faut construire un ensemble d'entiers A à partir de G.

  Il faut parvenir à traduire deux contraintes : un sommet peut appartenir ou ne pas appartenir à la couverture et le sous-ensemble de sommets couvrent bien toutes les arêtes.

- Pour cela,
  - on fixe une numérotation des sommets et des arêtes.
  - pour chaque couple (arête,sommet), $b_{ij} = 1$ si l'arête i est incidente au sommet j, sinon $b_{ij} = 0$
- On construit l'ensemble A de la façon suivante : $(b = 4)$
  - Pour chaque sommet j : $a_j = b^m + \sum_{i=0}^{m-1} b_{ij} b^i$
- On construit l'entier t :

$$t = \underbrace{k b^m}_{cardinalite\ de\ la\ couverture} + \ldots$$

▸ Voir un exemple

VC $\leq$ SOMME DE SOUS-ENSEMBLE (1/2)

- On se donne un graphe $G = (V, E)$ et un entier k.

  Il nous faut construire un ensemble d'entiers A à partir de G.

  Il faut parvenir à traduire deux contraintes : un sommet peut appartenir ou ne pas appartenir à la couverture et le sous-ensemble de sommets couvrent bien toutes les arêtes.

- Pour cela,
  - on fixe une numérotation des sommets et des arêtes.
  - pour chaque couple (arête,sommet), $b_{ij} = 1$ si l'arête i est incidente au sommet j, sinon $b_{ij} = 0$
- On construit l'ensemble A de la façon suivante : $(b = 4)$
  - Pour chaque sommet j : $a_j = b^m + \sum_{i=0}^{m-1} b_{ij} b^i$
- On construit l'entier t :

$$t = \underbrace{k b^m}_{cardinalite\ de\ la\ couverture} + \ldots$$

Voir un exemple

VC $\leq$ SOMME DE SOUS-ENSEMBLE (1/2)

- On se donne un graphe $G = (V, E)$ et un entier k.

  Il nous faut construire un ensemble d'entiers A à partir de G.

  Il faut parvenir à traduire deux contraintes : un sommet peut appartenir ou ne pas appartenir à la couverture et le sous-ensemble de sommets couvrent bien toutes les arêtes.

- Pour cela,
  - on fixe une numérotation des sommets et des arêtes.
  - pour chaque couple (arête,sommet), $b_{ij} = 1$ si l'arête i est incidente au sommet j, sinon $b_{ij} = 0$

- On construit l'ensemble A de la façon suivante : $(b = 4)$
  - Pour chaque sommet j : $a_j = b^m + \sum_{i=0}^{m-1} b_{ij} b^i$
  - Pour chaque arête i : est associé l'entier b^i

- On construit l'entier t :

$$t = \underbrace{k b^m}_{cardinalite\ \ de\ \ la\ \ couverture} + \sum_{i=0}^{m-1} \underbrace{2b^i}_{cout\ \ de\ \ l'\ \ arete\ \ i}$$

▸ Voir un exemple

VC $\leq$ SOMME DE SOUS-ENSEMBLE (2/2)

- On peut prouver :

il existe une couverture sommet du graphe G de cardinalité k

$$\Longleftrightarrow$$

il existe un sous-ensemble $A' \subseteq A$ tel que $\sum_{a \in A'} a = t$

VC $\leq$ SOMME DE SOUS-ENSEMBLE (2/2)

- On peut prouver :

  il existe une couverture sommet du graphe G de cardinalité k

  $$\Longleftrightarrow$$

  il existe un sous-ensemble $A' \subseteq A$ tel que $\sum_{a \in A'} a = t$

- La réduction se fait en temps polynomial :
  - Un entier est construit pour chaque arête et pour chaque sommet.

    Au pire des cas, il y a $\mathcal{O}(|E|)$ entiers dans l'instance.
  - Chaque entier se construit en $m + 1$ opérations.

En résumé

Théorème

Le problème SOMME DE SOUS-ENSEMBLE est NP-complet si les entiers sont codés en base 4.

En résumé

Théorème

Le problème SOMME DE SOUS-ENSEMBLE est NP-complet si les entiers sont codés en base 4.

Remarque :

En résumé

Théorème

Le problème SOMME DE SOUS-ENSEMBLE est NP-complet si les entiers sont codés en base 4.

Remarque :

- En informatique, les données (les entiers) sont codés par des 0 et des 1.
- Le problème, reste-il NP-complet si les entiers sont codés en binaire (en base 2) ?

Codage des nombres entiers

- Un entier i s'écrit en base b en utilisant des b chiffres allant de 0 à $b-1$:

  i s'écrit $c_n ... c_2 c_1 c_0$ en base b ssi $i = c_n b^n + ... + c_2 b^2 + c_1 b^1 + c_0 b^0$

- Par exemple :
  - $2406 = 2 \cdot 10^3 + 4 \cdot 10^2 + 0 \cdot 10^1 + 6 \cdot 10^0$ (en base 10)
  - $1001 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$ (en base 2)

- Un entier i entre 0 et $b^n - 1$ peut être coder en n chiffres, ou en
  - $\lceil \log_b(i+1) \rceil$ chiffres
  - $\lceil \log_2(i+1) \rceil$ bits si $b = 2$

On peut le prouver par récurrence.

En résumé

Théorème

Le problème SOMME DE SOUS-ENSEMBLE est NP-complet si les entiers sont codés en base 2 (en base *i* avec $i > 1$).

En résumé

Théorème

Le problème SOMME DE SOUS-ENSEMBLE est NP-complet si les entiers sont codés en base 2 (en base i avec $i > 1$).

Définitions :

- Un problème NP-complet est **faiblement NP-complet** si il est NP-complet même avec les entiers de l'instance codés en base 2.

En résumé

Théorème

Le problème SOMME DE SOUS-ENSEMBLE est NP-complet si les entiers sont codés en base 2 (en base i avec $i > 1$).

Définitions :

- Un problème NP-complet est **faiblement NP-complet** si il est NP-complet même avec les entiers de l'instance codés en base 2.

- Un problème NP-complet est **fortement NP-complet** si il est NP-complet avec les entiers de l'instance codés en unaire.

En résumé

Théorème

Le problème SOMME DE SOUS-ENSEMBLE est NP-complet si les entiers sont codés en base 2 (en base *i* avec $i > 1$).

Définitions :

- Un problème NP-complet est **faiblement NP-complet** si il est NP-complet même avec les entiers de l'instance codés en base 2.

- Un problème NP-complet est **fortement NP-complet** si il est NP-complet avec les entiers de l'instance codés en unaire.

Le problème SOMME DE SOUS-ENSEMBLE,
est-il fortement NP-complet ?

Remarques

- Fortement NP-complet implique faiblement NP-complet
  - On peut passer du codage unaire au codage binaire en temps polynomial.
- Mais la réciproque est fausse.

Plan

Programmation dynamique

- C'est une des plus vieilles techniques pour produire des algorithmes exacts plus efficaces que l'énumération exhaustive.

Programmation dynamique

- C'est une des plus vieilles techniques pour produire des algorithmes exacts plus efficaces que l'énumération exhaustive.

Principe (Bellman, 1949)

Composer une solution optimale du problème en combinant les solutions (optimales) de ses sous-problèmes.

Programmation dynamique

- C'est une des plus vieilles techniques pour produire des algorithmes exacts plus efficaces que l'énumération exhaustive.

Principe (Bellman, 1949)

Composer une solution optimale du problème en combinant les solutions (optimales) de ses sous-problèmes.

- En pratique :
  - Décomposer le problème en des sous-problèmes plus petits ;
  - Calculer les solutions optimales de tous ces sous-problèmes et les garder en mémoire.
  - Calculer la solution optimale à partir des solutions optimales des sous-problèmes

Plus précisément

Programmation dynamique

- Suite de Fibonacci
- Problème SOMME DE SOUS-ENSEMBLE

Suite de Fibonacci

Suite de Fibonacci
Données : Un entier $t \in \mathbb{N}$
Objectif : Calculer le n-ième terme de la suite de Fibonacci

$$(F(0) = F(1) = 1,\ F(n) = F(n-1) + F(n-2)).$$

On peut le calculer en utilisant :

Suite de Fibonacci

Suite de Fibonacci
Données : Un entier $t \in \mathbb{N}$
Objectif : Calculer le n-ième terme de la suite de Fibonacci

$$(F(0) = F(1) = 1,\ F(n) = F(n-1) + F(n-2)).$$

On peut le calculer en utilisant :
soit la récursivité
Fib(n)

1. si (n=0) ou (n=1) alors
   retourner 1 ;
2. sinon retourner
   Fib(n-1) + Fib(n-2) ;

Suite de Fibonacci

Suite de Fibonacci
Données : Un entier $t \in \mathbb{N}$
Objectif : Calculer le n-ième terme de la suite de Fibonacci

$$(F(0) = F(1) = 1,\ F(n) = F(n-1) + F(n-2)).$$

On peut le calculer en utilisant :
soit la récursivité
Fib(n)

1. si (n=0) ou (n=1) alors
   retourner 1 ;
2. sinon retourner
   Fib(n-1) + Fib(n-2) ;

nb exponentiel d'appels récursifs

Suite de Fibonacci

Suite de Fibonacci
Données : Un entier $t \in \mathbb{N}$
Objectif : Calculer le n-ième terme de la suite de Fibonacci

$$(F(0) = F(1) = 1,\ F(n) = F(n-1) + F(n-2)).$$

On peut le calculer en utilisant :

soit la récursivité

Fib(n)

1. si (n=0) ou (n=1) alors
   retourner 1 ;
2. sinon retourner
   Fib(n-1) + Fib(n-2) ;

nb exponentiel d'appels récursifs

ou soit la prog. dynamique :

Fib-Dynamique (n)

1. F[0]=1 ; F[0]=2
2. pour i allant de 3 à n, faire
   F[i]=F[i-1] + F[i-2] ;
3. retourner $F[n]$

n additions

Plus précisément

Programmation dynamique

- Suite de Fibonacci
- Problème SOMME DE SOUS-ENSEMBLE

Problème SOMME DE SOUS-ENSEMBLE

Données : un ensemble fini d'entiers $A = \{a_1, a_2, \ldots, a_\ell\}$ et un entier $t \in \mathbb{N}$
Question : Existe-t-il $A' \subseteq A$ tel que $\sum_{a \in A'} a = t$?
L'algorithme basé sur la programmation dynamique se compose de la façon suivante

Problème SOMME DE SOUS-ENSEMBLE

Données : un ensemble fini d'entiers $A = \{a_1, a_2, \ldots, a_\ell\}$ et un entier $t \in \mathbb{N}$
Question : Existe-t-il $A' \subseteq A$ tel que $\sum_{a \in A'} a = t$?
L'algorithme basé sur la programmation dynamique se compose de la façon suivante

- Décomposer le problème en des sous-problèmes plus petits ;

  Considérer les problèmes avec $A_i = \{a_1, \ldots, a_i\}$, $i = 1, \ldots, |A|$

Problème SOMME DE SOUS-ENSEMBLE

Données : un ensemble fini d'entiers $A = \{a_1, a_2, \ldots, a_\ell\}$ et un entier $t \in \mathbb{N}$
Question : Existe-t-il $A' \subseteq A$ tel que $\sum_{a \in A'} a = t$?
L'algorithme basé sur la programmation dynamique se compose de la façon suivante

- Décomposer le problème en des sous-problèmes plus petits ;

  Considérer les problèmes avec $A_i = \{a_1, \ldots, a_i\}$, $i = 1, \ldots, |A|$
- Calculer les solutions optimales de tous ces sous-problèmes.

  Calculer la liste P_i des sommes des sous ensembles de A_i.

Problème SOMME DE SOUS-ENSEMBLE

Données : un ensemble fini d'entiers $A = \{a_1, a_2, \ldots, a_\ell\}$ et un entier $t \in \mathbb{N}$

Question : Existe-t-il $A' \subseteq A$ tel que $\sum_{a \in A'} a = t$?

L'algorithme basé sur la programmation dynamique se compose de la façon suivante

- Décomposer le problème en des sous-problèmes plus petits ;

  Considérer les problèmes avec $A_i = \{a_1, \ldots, a_i\}$, $i = 1, \ldots, |A|$
- Calculer les solutions optimales de tous ces sous-problèmes.

  Calculer la liste P_i des sommes des sous ensembles de A_i.
- Calculer la solution optimale à partir des solutions optimales des sous-problèmes

  Trouver la relation entre P_i et P_{i+1}

SOMME DE SOUS-ENSEMBLE : illustration

- Instance : $A = \{1, 4, 5\}$ et $t = 8$
- P_i : la liste des sommes de tous les sous ensembles de $A_i = \{a_1, \ldots, a_i\}$.

SOMME DE SOUS-ENSEMBLE : illustration

- Instance : $A = \{1, 4, 5\}$ et $t = 8$
- P_i : la liste des sommes de tous les sous ensembles de $A_i = \{a_1, \ldots, a_i\}$.
- Calculons les P_i :

SOMME DE SOUS-ENSEMBLE : illustration

- Instance : $A = \{1, 4, 5\}$ et $t = 8$
- P_i : la liste des sommes de tous les sous ensembles de $A_i = \{a_1, \ldots, a_i\}$.
- Calculons les P_i :
  - $P_0 =< 0 >$ (par convention)

SOMME DE SOUS-ENSEMBLE : illustration

- Instance : $A = \{1, 4, 5\}$ et $t = 8$
- P_i : la liste des sommes de tous les sous ensembles de $A_i = \{a_1, \ldots, a_i\}$.
- Calculons les P_i :
  - $P_0 =< 0 >$ (par convention)
  - $P_1 =< 0, 1 >$

    listes des sommes des sous-ensembles ($\emptyset$, $\{1\}$) de $A_1 = \{a_1\}$

SOMME DE SOUS-ENSEMBLE : illustration

- Instance : $A = \{1, 4, 5\}$ et $t = 8$
- P_i : la liste des sommes de tous les sous ensembles de $A_i = \{a_1, \dots, a_i\}$.
- Calculons les P_i :
  - $P_0 =< 0 >$ (par convention)
  - $P_1 =< 0, 1 >$

    listes des sommes des sous-ensembles ($\emptyset$, $\{1\}$) de $A_1 = \{a_1\}$
  - $P_2 =< 0, 1, 4 >$

    listes des sommes des sous-ensembles
    ($\emptyset, \{1\}, \{4\}, \{1, 4\}$) de $A_2 = \{a_1, a_2\}$
  - $P_3 =< 0, 1, 4, 5, 6, 9, 10 >$

    listes des sommes des sous-ensembles de A_3

Notation

- P_i : la liste des sommes de tous les sous ensembles de $A_i = \{a_1, \ldots, a_i\}$.

- $L \oplus x$: la liste des entiers de L incrémentés par x

$$L \oplus x = \{s + x : s \in L\}$$

- On peut prouver : $$P_i = P_{i-1} \cup (P_{i-1} \oplus a_i)$$

- On note la procédure $\mathit{fusion} - \mathit{liste}(L, P)$ correspondant à la fusion de deux listes L et P d'entiers en une liste triée.

Cette procédure peut se réaliser en temps polynomial (en $\mathcal{O}(|L| + |P|)$ opérations)

Algorithme dynamique

Entrée : : un ensemble fini d'entiers $A = \{a_1, a_2, \ldots, a_\ell\}$ et un entier $t \in \mathbb{N}$
Sortie :

1. $L_0 \leftarrow < 0 >$
2. pour i à allant de 1 à $|A|$
    2.1 $L_i \leftarrow fusion - liste(L_{i-1}, L_{i-1} \oplus a_i)$
    2.2 Supprimer tous les éléments de L_i qui sont supérieurs à t
3. retourner le plus grand élément de $L_{|A|}$

Algorithme dynamique

Entrée : : un ensemble fini d'entiers $A = \{a_1, a_2, \ldots, a_\ell\}$ et un entier $t \in \mathbb{N}$
Sortie :

1. $L_0 \leftarrow < 0 >$
2. pour i à allant de 1 à $|A|$
   2.1 $L_i \leftarrow fusion - liste(L_{i-1}, L_{i-1} \oplus a_i)$
   2.2 Supprimer tous les éléments de L_i qui sont supérieurs à t
3. retourner le plus grand élément de $L_{|A|}$

Complexité : $\mathcal{O}(t \times |A| \times t)$

- A chaque itération, la liste résultant contient au plus t éléments.

En résumé

L'algorithme retourne la plus grande valeur de la somme d'un sous-ensemble de A plus petit que t, en $\mathcal{O}(t \times |A| \times t)$ opérations :

- en temps polynomial si l'entier en codé en unaire
- en temps exponentiel si l'entier en codé en binaire

car $\mathcal{O}(t) = \mathcal{O}(2^{logt})$

Plan

Problèmes fortement NP-complets

Rappel : Un problème NP-complet est **fortement NP-complet** si il est NP-complet avec les entiers de l'instance codés en unaire.

Il existe des problèmes fortement NP-complets comme :

SOMME DE SOUS-ENSEMBLE GENERALISE
Données : un ensemble fini d'entiers $A = \{a_1, a_2, \ldots, a_\ell\}$, en entier m et un entier $t \in \mathbb{N}$
Question : Existe-t-il m sous-ensembles A_1, ..., A_m deux-à-deux disjoints tel que $\sum_{a \in A_i} a = t$ pour $i = 1, \ldots, m$?

Autres problèmes fortement NP-complets.

- Couverture de sommets pondérée (WVC)
  Données : un graphe non-orienté $G = (V, E)$, une fonction de coût sur les sommets $c : V \to \mathbb{N}$ et un entier k.
  Question : G contient-il une *couverture de sommets* $\mathcal{S}$ telle que $\sum_{v \in \mathcal{S}} c(v) \leq k$?

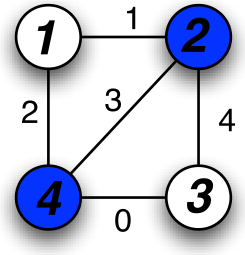
- MAX-SAT
  Données : un ensemble U de variables, un ensemble de clauses, une fonction c de coût sur les clauses et un entier k.
  Question : Existe-t-il une fonction $t : U \to \{0, 1\}$ telle que la somme des coûts des clauses satisfaites par t est supérieure ou égale à k ?

Plan

Exemple

Soit $\mathcal{I} = (G, k)$ une instance de VC :

- le graphe G



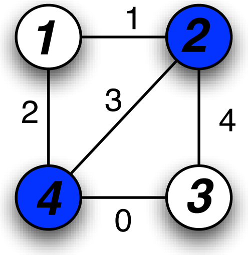
- $k = 2$

Retour à la preuve

Exemple

Soit $\mathcal{I} = (G, k)$ une instance de VC :

- le graphe G



- $k = 2$

L'ensemble A est défini de la façon suivante :

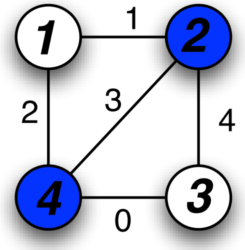
- Le sommet 1 est représenté par $a_1 = b^5 + b^1 + b^2$
- Le sommet 2 par $a_2 = b^5 + b^1 + b^3 + b^4$
- Le sommet 3 par $a_3 = b^5 + b^0 + b^4$
- Le sommet 4 par $a_4 = b^5 + b^0 + b^2 + b^3$

Retour à la preuve

Exemple

Soit $\mathcal{I} = (G, k)$ une instance de VC :

- le graphe G



- $k = 2$

L'ensemble A est défini de la façon suivante :

- Le sommet 1 est représenté par $a_1 = b^5 + b^1 + b^2$
- Le sommet 2 par $a_2 = b^5 + b^1 + b^3 + b^4$
- Le sommet 3 par $a_3 = b^5 + b^0 + b^4$
- Le sommet 4 par $a_4 = b^5 + b^0 + b^2 + b^3$

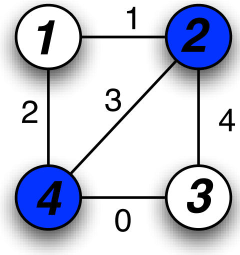
Si les sommets 2 et 4 correspondent à une couverture sommet, alors

▸ Retour à la preuve

Exemple

Soit $\mathcal{I} = (G, k)$ une instance de VC :

- le graphe G



- $k = 2$

L'ensemble A est défini de la façon suivante :

- Le sommet 1 est représenté par $a_1 = b^5 + b^1 + b^2$
- Le sommet 2 par $a_2 = b^5 + b^1 + b^3 + b^4$
- Le sommet 3 par $a_3 = b^5 + b^0 + b^4$
- Le sommet 4 par $a_4 = b^5 + b^0 + b^2 + b^3$

Si les sommets 2 et 4 correspondent à une couverture sommet, alors

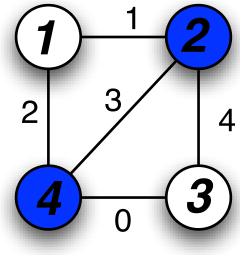
- $A' = \{a_2, a_3, \qquad\qquad \}$

Retour à la preuve

Exemple

Soit $\mathcal{I} = (G, k)$ une instance de VC :

- le graphe G



- $k = 2$

L'ensemble A est défini de la façon suivante :

- Le sommet 1 est représenté par $a_1 = b^5 + b^1 + b^2$
- Le sommet 2 par $a_2 = b^5 + b^1 + b^3 + b^4$
- Le sommet 3 par $a_3 = b^5 + b^0 + b^4$
- Le sommet 4 par $a_4 = b^5 + b^0 + b^2 + b^3$

Si les sommets 2 et 4 correspondent à une couverture sommet, alors

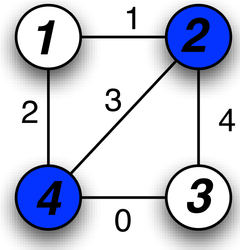
- $A' = \{a_2, a_3, \qquad \}$
- $\sum_{a \in A'} a = kb^5 + b^0 + b^1 + b^2 + 2b^3 + b^4$

Retour à la preuve

Exemple

Soit $\mathcal{I} = (G, k)$ une instance de VC :

- le graphe G



- $k = 2$

L'ensemble A est défini de la façon suivante :

- Le sommet 1 est représenté par $a_1 = b^5 + b^1 + b^2$
- Le sommet 2 par $a_2 = b^5 + b^1 + b^3 + b^4$
- Le sommet 3 par $a_3 = b^5 + b^0 + b^4$
- Le sommet 4 par $a_4 = b^5 + b^0 + b^2 + b^3$
- les éléments correspondants aux arêtes suivants : $b^0,\ b^1,\ b^2,\ b^3,\ b^4$

Si les sommets 2 et 4 correspondent à une couverture sommet, alors

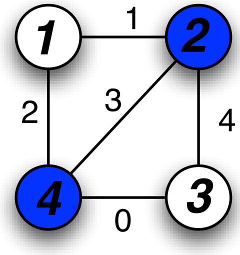
- $A' = \{a_2, a_3, b^0, b^1, b^2, b^4\}$
- $\sum_{a \in A'} a = kb^5 + 2b^0 + 2b^1 + 2b^2 + 2b^3 + 2b^4$

▸ Retour à la preuve

Exemple

Soit $\mathcal{I} = (G, k)$ une instance de VC :

- le graphe G



- $k = 2$

L'ensemble A est défini de la façon suivante :

- Le sommet 1 est représenté par $a_1 = b^5 + b^1 + b^2$
- Le sommet 2 par $a_2 = b^5 + b^1 + b^3 + b^4$
- Le sommet 3 par $a_3 = b^5 + b^0 + b^4$
- Le sommet 4 par $a_4 = b^5 + b^0 + b^2 + b^3$
- les éléments correspondants aux arêtes suivants : $b^0,\ b^1,\ b^2,\ b^3,\ b^4$

Si les sommets 2 et 4 correspondent à une couverture sommet, alors

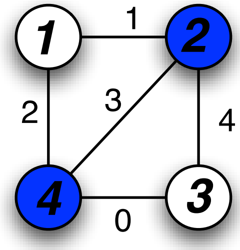
- $A' = \{a_2, a_3, b^0, b^1, b^2, b^4\}$

▸ Retour à la preuve

Exemple

Soit $\mathcal{I} = (G, k)$ une instance de VC :

- le graphe G



- $k = 2$

L'ensemble A est défini de la façon suivante :

- Le sommet 1 est représenté par $a_1 = b^5 + b^1 + b^2$
- Le sommet 2 par $a_2 = b^5 + b^1 + b^3 + b^4$
- Le sommet 3 par $a_3 = b^5 + b^0 + b^4$
- Le sommet 4 par $a_4 = b^5 + b^0 + b^2 + b^3$
- les éléments correspondants aux arêtes suivants : $b^0,\ b^1,\ b^2,\ b^3,\ b^4$

$t = kb^5 + 2b^0 + 2b^1 + 2b^2 + 2b^3 + 2b^4$

Si les sommets 2 et 4 correspondent à une couverture sommet, alors

- $A' = \{a_2, a_3, b^0, b^1, b^2, b^4\}$
- $\sum_{a \in A'} a = kb^5 + 2b^0 + 2b^1 + 2b^2 + 2b^3 + 2b^4$

Retour à la preuve