



RECHERCHE OPÉRATIONNELLE
SUD EST



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Couverture maximum de sommets dans les graphes bipartis

(PROJET DE SEMESTRE)

9 FÉVRIER 2006

AUTEUR :

Maurice GENOUD

Étudiant en mathématique, 3ème année

PROFESSEUR
Prof. D. DE WERRA

ASSISTANTE
Tinaz EKIM

Table des matières

| | |
|---|-----------|
| 1 Définitions, problèmes et complexité | 4 |
| 1.1 Définitions | 4 |
| 1.2 Notions de complexité | 5 |
| 1.3 Problèmes de couverture et de suppression de sommets | 6 |
| 1.4 BMax 2-Sat | 7 |
| 2 Approche de BMVC par le problème de suppression de sommets | 9 |
| 2.1 Notions complémentaires | 10 |
| 2.2 Propositions de l'article [MY] | 10 |
| 2.3 Insuffisance de l'article pour BMVC | 11 |
| 3 Preuve de la \mathcal{NP}-complétude de BMVC | 14 |
| 3.1 Préambule | 14 |
| 3.2 Idée de la preuve | 14 |
| 3.3 Plan de la preuve | 15 |
| 3.4 Construction du graphe G | 15 |
| 3.5 Construction du gadget H | 18 |
| 3.6 Propriétés du gadget | 19 |
| 3.7 G est biparti | 20 |
| 3.8 La réduction est polynomiale | 21 |
| 3.9 Equivalence entre BMVC dans G et BMax 2-Sat | 22 |
| 4 Améliorations possibles de la preuve | 26 |
| 4.1 Problèmes de notation | 26 |
| 4.2 Erreurs de calcul et d'indices | 27 |

Introduction

Le but de ce projet est d'étudier la complexité du problème de couverture maximum de sommets (MVC) dans les graphes bipartis (BMVC). Ces problèmes sont des variantes des problèmes de couverture de sommets (VC) et de couverture de sommets dans les graphes bipartis (BVC) respectivement.

Il a été prouvé depuis longtemps que MVC est \mathcal{NP} - *complet*. On sait aussi que BVC est un problème facile, i.e. résoluble en un temps polynomial. Cependant, il y a encore 4 ans, aucun résultat n'avait permis de conclure quoi que ce soit sur la complexité de BMVC.

On connaît aussi de nombreux résultats sur des problèmes qui pourraient nous aider à prouver la complexité de BMVC, notamment le problème de suppression de sommets. Malheureusement ces résultats s'appliquent uniquement à des restrictions de BMVC, ou alors ne s'appliquent tout simplement pas à BMVC. Nous en verrons les raisons au chapitre 2 de ce rapport.

C'est seulement en 2002 que parut la thèse de M. N. Apollonio [NA] de l'université de Rome intitulée : "Some constrained covering problems in graphs." Au chapitre 4 de cette thèse, on y trouve enfin une preuve de la \mathcal{NP} -complétude du problème BMVC. Ce résultat, loin d'être trivial et intuitif, est relativement surprenant, car, comme annoncé ci-dessus, BVC est polynomial.

Le troisième chapitre de ce rapport permettra au lecteur de comprendre l'intuition de cette preuve, sans toutefois y apporter tous les détails techniques. Puis, comme cette preuve comporte beaucoup d'erreurs d'indices, de problèmes de notations et de confusion dans la symbolique utilisée, je proposerai à la fin du rapport quelques propositions d'amélioration.

Je prie d'avance le lecteur d'excuser mes propres erreurs éventuelles et je lui souhaite beaucoup de plaisir lors de sa lecture.

Chapitre 1

Définitions, problèmes et complexité

Dans ce premier chapitre nous verrons d'abord quelques définitions utiles par la suite. Les notions de base de la théorie des graphes sont toutes supposées connues. Puis j'introduirai les notions de \mathcal{NP} – *difficile* et de \mathcal{NP} – *complet*. J'expliquerai ensuite les deux approches possibles de notre problème, ainsi que leur équivalence. Je donnerai enfin la définition du problème BMax 2-Sat. Ce chapitre est en grande partie basé sur un cours d'introduction à la recherche opérationnelle intitulé : "Théorie de la complexité", [TE].

1.1 Définitions

Un **problème** est un ensemble de questions paramétrées par une instance. L'**instance** d'un problème est une donnée composée des paramètres dont dépend le problème. Un algorithme résout un problème s'il peut répondre à toutes les questions quelle que soit l'instance.

Il existe deux types de problèmes :

- Un **problème d'optimisation** est un problème dont la réponse est une valeur optimale, i.e. une valeur minimale ou maximale.
- Un **problème de décision** est un problème dont la réponse est "oui" ou "non".

Exemple : Le problème du sac à dos.

Il s'agit de choisir un certain nombre d'objets parmi n objets de volumes respectifs v_1, \dots, v_n et d'utilités respectives u_1, \dots, u_n afin d'avoir une certaine utilité totale sans dépasser un volume donné.

La variante d'optimisation de ce problème consiste à trouver l'ensemble d'objets d'utilité maximum sans dépasser le volume V du sac à dos.

La variante de décision consiste à dire si "oui" ou "non" il existe un sous-ensemble d'objets d'utilité $\geq U$ et de volume $\leq V$.

1.2 Notions de complexité

On classe différents problèmes en plusieurs classes selon la complexité de l'algorithme qui les résout d'une façon exacte. La complexité d'un algorithme est une fonction qui associe à chaque instance d'un problème le nombre d'opérations effectuées par l'algorithme lorsqu'on lui applique l'instance donnée du problème.

On ne s'intéresse pas au nombre exact d'opérations, mais à son ordre (linéaire, polynomial, exponentiel, etc...)

Proposition 1.2.1. *Pour un problème donné, les variantes de décision et d'optimisation sont équivalentes en termes d'algorithme polynomial.*

Démonstration. 1. "Variante d'optimisation" \implies "Variante de décision" est clair.

2. La solution du problème d'optimisation est la valeur extrême (la plus grande ou la plus petite) pour laquelle la solution du problème de décision est "oui". La solution du problème d'optimisation peut donc être déterminée en répondant au problème de décision un nombre polynomial de fois. Donc "variante de décision" \implies "variante d'optimisation".

□

Il existe plusieurs classes de complexité :

1. \mathcal{P} : L'ensemble des problèmes de décision pour lesquels il existe un algorithme de résolution polynomial.
2. \mathcal{NP} : L'ensemble des problèmes de décision pour lesquels il existe un algorithme polynomial permettant de vérifier l'exactitude d'une réponse "oui".

Remarque : $\mathcal{P} \subseteq \mathcal{NP}$

Mais il reste toujours une question : $\mathcal{P} = \mathcal{NP}$? Cette question vaut 1'000'000\$ (le prix Clay). Il n'est en effet pas encore prouvé que $\mathcal{P} = \mathcal{NP}$ ou que $\mathcal{P} \neq \mathcal{NP}$. On pense cependant que $\mathcal{P} \neq \mathcal{NP}$.

On dit que P_1 est **polynomialement réductible** à P_2 , noté $P_1 \propto P_2$, s'il existe :

- une fonction polynomiale transformant chaque instance de P_1 en une instance de P_2 ,

- une fonction polynomiale transformant chaque solution de P_2 en une solution de P_1 .

Ainsi P_1 est polynomialement réductible à P_2 et si l'on connaît un algorithme polynomial résolvant P_2 , on en déduit que $P_1 \in \mathcal{P}$. En effet, on traduit les instances de P_1 en instances de P_2 , puis on résout P_2 et enfin on retraduit les solutions de P_2 en solutions de P_1 , tout cela d'ordre polynomial.

L'algorithme de résolution du sous-problème P_2 est appelé **algorithme esclave**, tandis que l'algorithme servant à résoudre P_1 (composé d'une transformation d'instance, de l'algorithme esclave et d'une transformation de solutions) est appelé **algorithme maître**.

Remarques

1. Informellement, $P_1 \propto P_2$ signifie que P_1 n'est pas plus difficile que P_2 .
2. \propto est une relation transitive : $P_1 \propto P_2 \propto P_3 \Rightarrow P_1 \propto P_3$

La classe des problèmes dits **\mathcal{NP} -complets** est l'ensemble des problèmes de décision P vérifiant :

1. $P \in \mathcal{NP}$ et
2. $\forall P' \in \mathcal{NP}, P' \propto P$

La classe des problèmes dits **\mathcal{NP} -difficiles** est l'ensemble des problèmes d'optimisation dont les versions de décision sont \mathcal{NP} -complets.

1.3 Problèmes de couverture et de suppression de sommets

1. Le problème de **couverture de sommet**, VC (Vertex Cover), consiste à trouver, dans un graphe $G = (V, E)$ donné un sous-ensemble de V de cardinalité minimum couvrant toutes les arêtes du graphe, i.e. trouver $V' \subseteq V \mid \forall u = (v_1, v_2), \{v_1, v_2\} \cap V' \neq \emptyset$ avec $|V'|$ minimum.
 - Ce problème est \mathcal{NP} -difficile en général.
 - Ce problème est polynomial pour les graphes bipartis.
2. Le problème de **couverture maximale de sommets**, MVC (Maximum Vertex Cover), est une variante de VC qui consiste à trouver le nombre maximum d'arêtes qu'il est possible de toucher avec k sommets.
 - Ce problème est \mathcal{NP} -difficile, même restreint aux graphes bipartis (c'est le sujet de ce rapport).
3. Le problème de **suppression de sommets** consiste à trouver un nombre minimum de sommets qui, supprimés, impliquent que le graphe

restant satisfait une propriété π donnée.

- La complexité de ce problème dépend de la propriété π .

MVC se ramène alors à un problème de suppression de sommets avec :

$\pi = \text{“avoir } X \text{ arêtes au maximum”}$

On a bien l'équivalence entre ces deux problèmes car, pour un graphe $G = (V, E)$ trouver le nombre maximum d'arêtes X qu'il est possible de toucher avec k sommets est équivalent à trouver le nombre minimum k de sommets qui supprimés impliquent que le sous graphe restant possède au maximum $e - X$ arêtes, où $e = |E|$.

Remarque 1

Dans MVC, on se donne k et on veut trouver X , tandis que dans le problème de suppression de sommets, on se donne X ($e - X$ en fait) et on veut trouver k . On doit donc appliquer l'algorithme de résolution du problème de suppression de sommets n^2 (correspond au nombre maximum d'arêtes de G) fois au maximum, où $n = |V|$, en posant successivement $e - X = 0, e - X = 1, e - X = 2, \dots$ jusqu'à ce qu'il donne k comme nombre minimum de sommets à supprimer.

Remarque 2

Le problème MVC restreint aux graphes bipartis se note aussi BMVC.

1.4 BMax 2-Sat

Le problème **Sat** (pour satisfiability) consiste à affecter une valeur “vrai” ou “faux” à n **variables** booléennes x_1, x_2, \dots, x_n de manière à satisfaire une instance du problème, qui est une expression du type :

$$(x_2 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee x_4) \wedge (x_1 \vee x_2)$$

Avec $\vee = \text{“ou”}$, $\wedge = \text{“et”}$ et $\bar{x}_i = \text{négation de } x_i$.

Chacune des apparitions de x_i ou \bar{x}_i est appelée **littérale** et chacune des expressions entre parenthèses est appelée **clause**.

L'instance est satisfaite si chaque clause est satisfaite, i.e. si chaque clause comporte au moins une fois la valeur “vrai”.

Par exemple l'affectation $x_1 = \text{faux}$, $x_2 = \text{vrai}$, $x_3 = \text{faux}$ et $x_4 = \text{faux}$ satisfait l'instance ci-dessus, tandis que l'affectation $x_1 = \text{faux}$, $x_2 = \text{faux}$, $x_3 = \text{vrai}$ et $x_4 = \text{faux}$ ne la satisfait pas.

2-Sat signifie que chaque clause se compose d'au plus deux littérales. Ce problème est polynomial.

Il n'est pas toujours possible de satisfaire toutes les clauses de l'instance, Le problème Max Sat consiste à trouver une affectation qui en satisfasse le maximum. Ce problème devient alors $\mathcal{NP} - \text{complet}$.

Dans le deuxième article, on se servira de Max 2-Sat avec la condition supplémentaire que les deux littérales d'une même clause ne peuvent pas être toutes les deux négatives. On nommera alors ce problème BMax 2-Sat. Ce problème est aussi $\mathcal{NP} - \text{complet}$.

Chapitre 2

Approche de BMVC par le problème de suppression de sommets

Comme le problème BMVC peut se ramener à un problème de suppression de sommets via la propriété $\pi = \text{“avoir } X \text{ arêtes au maximum”}$ (cf. paragraphe 1.3), je vais voir si les résultats connus sur les problèmes de suppression de sommets suffisent pour déterminer la classe de complexité de BMVC. Pour cela je vais étudier un article écrit en mai 1981 par M. Yannakakis intitulé : “NODE-DELETION PROBLEMS ON BIPARTITE GRAPHS”, [MY]. Comme son nom l’indique, cet article traite le problème de suppression de sommets pour différentes propriétés π . Plus précisément, on trouve dans cet article un récapitulatif de tous les résultats, et leurs démonstrations, connus à ce jour au sujet de la complexité des problèmes de suppression de sommets dans les graphes bipartis.

Dans un premier temps, je vais traduire les énoncés des propositions principales de l’article. Ces propositions, qui traitent toujours de problèmes de suppression de sommets, pourraient s’avérer utiles pour le problème MVC dans les graphes bipartis via notre propriété $\pi = \text{“avoir } X \text{ arêtes au maximum”}$, comme vu au chapitre 1.

Dans un second temps, je vais montrer pourquoi en fait cet article ne nous est d’aucune utilité pour trouver la complexité du problème BMVC, i.e. pourquoi $\pi = \text{“avoir } X \text{ arêtes au maximum”}$ ne satisfait pas les hypothèses de ces propositions.

Mais avant cela, je vais commencer par définir quelques notions que l’on retrouve dans les propositions qui suivent.

2.1 Notions complémentaires

Une propriété d'un graphe est dite **non-triviale** si elle est vraie pour un unique sommet et s'il existe des graphes où elle est fausse.

Une propriété d'un graphe est dite **intéressante** si il existe des graphes arbitrairement grands qui satisfont cette propriété.

Une propriété d'un graphe est dite **héréditaire** si lorsqu'un graphe satisfait cette propriété, alors tous les sous-graphes possibles de ce graphe la satisfont aussi.

Une propriété d'un graphe est dite **déterminée par les composantes** si lorsque toutes les composantes connexes du graphe satisfont cette propriété, alors le graphe tout entier la satisfait.

Pour un graphe G , on note $\overline{k(G)}$ le nombre de composantes connexes non-triviales (i.e. pas composé d'un unique élément) de G . Pour une propriété π , on définit alors $\overline{k(\pi)}$ comme suit :

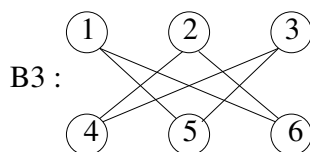
$$\overline{k(\pi)} = \sup\{\overline{k(G)} \mid G \text{ est un graphe biparti satisfaisant } \pi\}.$$

Pour un graphe G , on note $\overline{\nu(G)}$ le nombre de différents voisinages de sommets de G . Pour une propriété π , on définit alors $\overline{\nu(\pi)}$ comme suit :

$$\overline{\nu(\pi)} = \sup\{\overline{\nu(G)} \mid G \text{ est un graphe biparti satisfaisant } \pi\}.$$

Un graphe $G = (V, E)$ est **une étoile** s'il possède $|V - 1|$ arêtes reliant directement un sommet à tous les autres et aucune autre arête.

On note \underline{B}_t un graphe biparti avec les sommets x_1, x_2, \dots, x_t d'un coté et les sommets y_1, y_2, \dots, y_t de l'autre et contenant toutes les arêtes $(x_i, x_j) \forall i \neq j$ où $i, j = 1, 2, \dots, t$. Ex :



On note \underline{G}_1 l'ensemble de tous les graphes pour lesquels il existe toujours une arête qui relie directement deux arêtes quelconques de ce graphe.

2.2 Propositions de l'article [MY]

Proposition 2.2.1. *Le problème de suppression de sommets correspondant à une propriété héréditaire, non-triviale et intéressante est un problème \mathcal{NP} -complet.*

Proposition 2.2.2. *Le problème de suppression de sommets restreint aux graphes bipartis correspondant à une propriété héréditaire, déterminée par les composantes connexes, satisfaite par une unique arête et non satisfaite par tous les graphes étoilés est un problème \mathcal{NP} -complet.*

Proposition 2.2.3. *A l'exception de la propriété correspondant au problème de couverture de sommets, le problème de suppression de sommets correspondant à une propriété héréditaire, non-triviale dans les graphes bipartis et déterminée par les composantes connexes est un problème \mathcal{NP} -complet dans les graphes bipartis.*

Proposition 2.2.4. *Le problème de suppression de sommets correspondant à une propriété π telle que $k(\pi) = \infty$ est un problème \mathcal{NP} -complet.*

Proposition 2.2.5. *Le problème de suppression de sommets correspondant à une propriété héréditaire, non-triviale dans les graphes bipartis et satisfaite pour chaque couplage est un problème \mathcal{NP} -complet dans les graphes bipartis.*

Proposition 2.2.6. *Le problème de suppression de sommets correspondant à une propriété héréditaire, non-triviale dans les graphes bipartis et satisfaite par tous les graphes B_t est un problème \mathcal{NP} -complet dans les graphes bipartis.*

Proposition 2.2.7. *Le problème de suppression de sommets restreint aux graphes bipartis correspondant à une propriété π satisfaite uniquement pour les graphes de G_1 est un problème \mathcal{NP} -complet.*

Proposition 2.2.8. *Le problème de suppression de sommets restreint aux graphes bipartis correspondant à une propriété π héréditaire, satisfaite pour tous les graphes de G_1 et telle que $k(\pi) < \infty$ est un problème \mathcal{NP} -complet.*

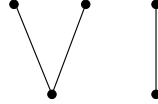
Proposition 2.2.9. *Le problème de suppression de sommets restreint aux graphes bipartis correspondant à une propriété héréditaire, non-triviale et telle que $\nu(\pi) = \infty$ est un problème \mathcal{NP} -complet alors que si $\nu(\pi) < \infty$, ce problème est polynomial.*

2.3 Insuffisance de l'article pour BMVC

Je vais maintenant montrer l'insuffisance de cet article pour prouver la \mathcal{NP} -complétude de MVC dans les graphes bipartis. Je vais donc expliquer pourquoi les hypothèses des propositions ci-dessus ne sont pas satisfaites. Dans la suite de ce chapitre, π désignera toujours la propriété : "avoir X arêtes au maximum".

1. La proposition 2.2.1 est un cas général. Elle ne nous dit donc rien sur la restriction du problème MVC aux graphes bipartis.

2. S'il est clair que π est non-triviale dans les graphes bipartis, intéressante et héréditaire, il est aussi aisé de montrer que π n'est pas déterminée par les composantes connexes. En effet, dans le graphe ci-dessous, π , avec $X = 2$ est satisfaite dans les deux composantes connexes (chaque composante a au plus $X = 2$ arêtes) mais π n'est pas satisfaite dans le graphe tout entier (qui a $3 > 2$ arêtes).



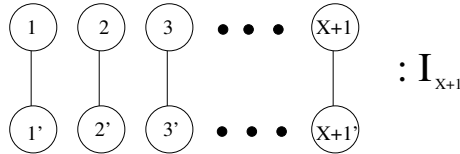
Ainsi π ne vérifie pas les hypothèses des propositions 2.2.2 et 2.2.3 .

3. Par la proposition suivante la proposition 2.2.4 ne s'applique pas à notre problème :

Proposition 2.3.1. Avec notre propriété $\pi =$ "avoir X arêtes au maximum", on a : $k(\pi) < \infty$.

Démonstration. Supposons par l'absurde que $k(\pi) = \infty \implies \sup\{k(G), G \text{ biparti avec } X \text{ arêtes au max}\} = \infty \implies \exists G \mid k(G) > X \forall X \implies G$ a plus que X arêtes. Contradiction avec le fait que G satisfait π (avoir au max X arêtes). \square

4. π n'est pas satisfaite pour chaque couplage. En effet, le graphe I_{X+1} illustré ci-dessous (le graphe biparti complément de B_{X+1}) est un couplage contenant $X + 1$ arêtes, i.e. ne satisfaisant pas π .



L'insuffisance pour notre problème de la proposition 2.2.5 en découle.

5. π n'est pas satisfaite par tous les graphes B_t . Par exemple, π n'est pas satisfaite pour B_X .

L'insuffisance pour notre problème de la proposition 2.2.6 en découle.

6. L'inutilité de la proposition 2.2.7 découle du fait que π n'est évidemment pas satisfaite uniquement pour les graphes de G_1 .

7. π n'est pas satisfaite $\forall G \in G_1$ (par exemple pas satisfaite pour B_X). L'insuffisance pour notre problème de la proposition 2.2.8 en découle.

8. Comme $\nu(G) \leq |G| \implies \nu(G) < \infty \implies \nu(\pi) < \infty$. On pourrait alors croire que le problème de suppression de sommets est polynomial avec la propriété π . Mais en analysant la preuve de cette proposition, on voit que l'algorithme nécessite en réalité un temps de l'ordre de $n^{\nu(\pi)}$, où $n = |G|$. Pour que cet algorithme soit polynomial, il faut donc non

seulement que $\nu(\pi) < \infty$, mais aussi que $\nu(\pi)$ soit constant, ce qui n'est pas le cas ici. Donc l'algorithme nécessite en réalité un temps de l'ordre de n^n , il n'est ainsi pas polynomial.

Pour notre problème, nous ne pouvons donc rien conclure de la proposition 2.2.9.

Ainsi ce premier article est totalement insuffisant pour prouver la \mathcal{NP} -complétude du problème de suppression de sommets avec la propriété π , donc totalement insuffisant pour prouver la \mathcal{NP} -complétude de BMVC. Mais ne désespérons pas et poursuivons avec le deuxième article.

Chapitre 3

Preuve de la \mathcal{NP} -complétude de BMVC

3.1 Préambule

La preuve de la \mathcal{NP} -complétude de BMVC ne fut réalisée qu'en 2002, dans la thèse de M. Nicola Apollonio [NA] de l'université de Rome. Cette thèse s'intitule : "Some constrained covering problems in graphs."

Je vais maintenant exposer le déroulement de cette preuve, ses différentes étapes et leur utilité. Le but recherché ici n'est pas de démontrer la \mathcal{NP} -complétude de BMVC de manière stricte et rigoureuse, mais seulement de manière intuitive. Je ferai donc de nombreuses références à la thèse de M. Nicola Apollonio [NA] pour les différents lemmes intervenant dans la preuve, dont j'énoncerai uniquement les résultats, et dont les démonstrations sont purement techniques, sans rien apporter d'important du point de vue conceptuel.

3.2 Idée de la preuve

Nous savons déjà que le problème BMVC appartient à la classe des problèmes \mathcal{NP} . Afin de prouver que le problème BMVC est \mathcal{NP} -complet, il suffit donc de trouver un graphe biparti G correspondant à une instance quelconque du problème BMax 2-Sat tel que le problème BMVC appliqué à G est \mathcal{NP} -complet.

Nous allons montrer qu'il est possible de résoudre un problème BMax 2-Sat quelconque en se ramenant à la résolution d'un problème BMVC. Il s'agit donc de montrer qu'il est possible de construire un graphe G , selon des règles bien précises, à partir d'une instance quelconque de BMax 2-Sat donnée, de telle manière que les problèmes BMax 2-Sat et BMVC appliqués

à G soit équivalents, i.e. tel que si l'on arrive à trouver une solution de BMVC dans G , alors on arrive à trouver une solution de BMax 2-Sat.

Explicitement, G devra être tel qu'il existe un assignement des variables du problème BMax 2-Sat satisfaisant au moins k clauses si et seulement si il est possible de couvrir au moins X arêtes de G avec un nombre maximum B_0 de sommets fixé. X et B_0 seront précisés plus tard.

Ainsi nous réduirons l'algorithme maître de résolution de BMax 2-Sat à l'algorithme esclave de résolution de BMVC, et cela en un temps polynomial. Donc si le problème BMVC était résoluble en un temps polynomial, BMax 2-Sat le serait aussi; il suffirait de traduire l'instance du problème BMax 2-Sat en une instance du problème BMVC, de résoudre ce dernier problème, et de retraduire sa solution en une solution du problème BMax 2-Sat, tout ceci en des temps polynomiaux. Or nous savons que cela est faux, i.e. que le problème BMax 2-Sat est \mathcal{NP} -complet. Par conséquent, BMVC n'est pas polynomial, donc BMVC est \mathcal{NP} -complet.

Il y a donc plusieurs assertions à vérifier.

3.3 Plan de la preuve

Voici le cheminement que nous allons suivre :

1. Construire le graphe G .
2. Construire le gadget H .
3. Énoncer quelques propriétés du gadget.
4. Prouver que G est bien biparti.
5. Montrer que la réduction du problème BMVC dans G au problème BMax 2-Sat se fait en un temps polynomial.
6. Montrer l'équivalence entre BMVC appliqué à G et BMax 2-Sat.

3.4 Construction du graphe G

On se donne maintenant une instance U de BMax 2-Sat avec n variables u_1, \dots, u_n (donc $2n$ littérales : $u_1, \dots, u_n, \bar{u}_1, \dots, \bar{u}_n$) et m clauses C_1, \dots, C_m . Rappel : dans BMax 2-Sat, une clause a soit une, soit deux littérales. Une clause est dite **pure** si elle contient uniquement deux littérales positives : $C_i = (u_j \vee u_k)$. Elle est dite **impure** sinon : $C_i = u_j$ ou $C_i = \bar{u}_j$ ou $C_i = (u_j \vee \bar{u}_k)$ ou $C_i = (\bar{u}_j \vee u_k)$. Soit p le nombre de clauses pures, J_0 leurs indices, i.e. $J_0 = \{i | C_i = \text{clause pure}\}$ et J_1 les indices des clauses impures, i.e. $J_1 = \{1, \dots, m\} - J_0$. Alors $|J_0| = p$ et $|J_1| = m - p$.

On va construire un graphe $G = (V(G), E(G))$ associé à cette instance selon les règles suivantes :

R1 Pour chaque variable u_i , on construit un graphe biparti complet G^i où une des parties notée A_i contient M sommets blancs $a_{i,1}, \dots, a_{i,M}$ associés aux littérales u_i et l'autre partie notée B^i contient M sommets noirs $b_{i,1}, \dots, b_{i,M}$ associés aux littérales \bar{u}_i . On pose $M = \Delta(H) + m + 1$, où H est le gadget (cf. prochain paragraphe) et $\Delta(H)$ correspond au degré maximum d'un sommet dans H .

R2 Pour chaque clause impure C_i avec $i \in J_1$ on construit une arête e_i comme ceci :

- si $|C_i| = 2$ et $C_i = (u_k \vee \bar{u}_l)$, $e_i = (a_{k,i}, b_{l,i})$,
 - si $|C_i| = 2$ et $C_i = (\bar{u}_k \vee u_l)$, $e_i = (b_{k,i}, a_{l,i})$,
 - si $C_i = (u_j)$, $e_i = (a_{j,i}, w_i)$ où w_i est un sommet noir pendant de G ,
 - si $C_i = (\bar{u}_j)$, $e_i = (b_{j,i}, w_i)$ où w_i est un sommet blanc pendant de G .
- L'ensemble de ces sommets pendants (feuilles) est appelé R .

R3 Pour chaque clause pure $C_i = (u_k, u_l)$ avec $i \in J_0$, on construit deux arêtes $e_{1,i} = (a_{k,i}, x_i)$ et $e_{2,i} = (a_{l,i}, y_i)$, où x_i et y_i sont des sommets noirs appelés **terminaux**. L'ensemble des terminaux est noté Z . Ces terminaux appartiennent à un sous graphe $H = H(\lambda, \delta, p)$ de G , appelé le gadget. λ et δ seront précisés plus tard. On pose de plus pour chaque clause pure $C_i : L_i = \{e_{1,i}, e_{2,i}\}$

On pose $\tilde{G} = \cup_{i=1}^n G^i$, $E_1 = \{e_i | i \in J_1\}$ et $E_0 = \{e_{1,i}, e_{2,i} | i \in J_0\}$.

On a alors $V(G) = V(\tilde{G}) \cup V(H) \cup V(R)$ et $E(G) = E(\tilde{G}) \cup E(H) \cup E_0 \cup E_1$.

Cette construction devrait être plus claire grâce à l'exemple de la page suivante.

Exemple

Soit U , l'instance de BMax 2-Sat suivante donnée avec $m = 6$ clauses et $n = 5$ variables :

$$U = (u_1 \vee \bar{u}_2) \wedge (\bar{u}_3 \vee u_1) \wedge (u_4) \wedge (\bar{u}_5) \wedge (u_2 \vee u_3) \wedge (u_1 \vee u_3)$$

Alors :

$$C_1 = (u_1 \vee \bar{u}_2) \implies e_1$$

$$C_2 = (\bar{u}_3 \vee u_1) \implies e_2$$

$$C_3 = (u_4) \implies e_3$$

$$C_4 = (\bar{u}_5) \implies e_4$$

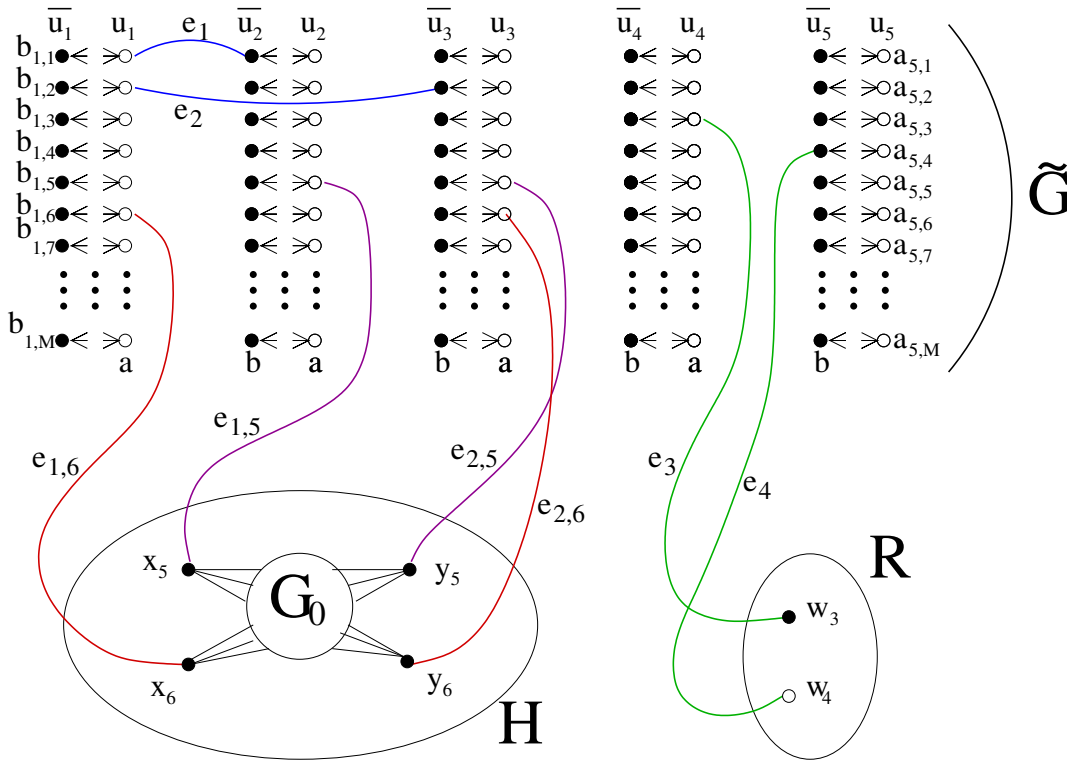
$$C_5 = (u_2 \vee u_3) \implies L_5 = \{e_{1,5}, e_{2,5}\}$$

$$C_6 = (u_1 \vee u_3) \implies L_6 = \{e_{1,6}, e_{2,6}\}$$

$$E_1 = \{e_1, e_2, e_3, e_4\}, E_1 = \{e_{1,5}, e_{2,5}, e_{1,6}, e_{2,6}\}$$

$$J_0 = \{5, 6\}, J_1 = \{1, 2, 3, 4\}$$

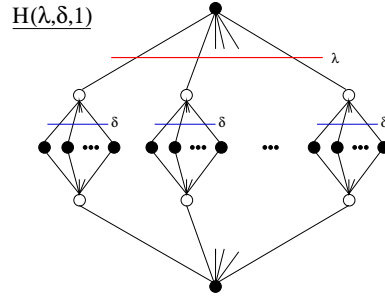
On obtient alors le graphe G suivant :



3.5 Construction du gadget H

Je vais maintenant expliquer la construction du gadget $H = H(\lambda, \delta, p)$. Nous verrons par la suite pourquoi on doit introduire ces deux paramètres λ et δ supplémentaires et comment ils sont définis. H se construit de manière différente selon que le nombre de clauses pures p vaut 1 ou est plus grand ou égal à 2.

$p = 1$ $H = H(\lambda, \delta, 1)$. Soit alors k la clause pure. On a $L_k = \{e_{1,k}, e_{2,k}\} = \{(a_{i,k}, x_k), (a_{j,k}, y_k)\}$. On note alors $x_k := x$ et $y_k := y$. On construit alors λ graphes bipartis avec 2 sommets blancs et δ sommets noirs, où chaque sommet blanc est relié à tous les sommets noirs. On relie ensuite x à un sommet blanc de chacun de ces λ graphes et y à l'autre sommet blanc de chacun de ces λ graphes. On obtient ainsi le graphe suivant :



$p \geq 2$ Nous appelons une paire de terminaux $(z, z') | z, z' \in Z$ une *bonne paire* si $z \neq z'$ et si $(z, z') \neq (x_i, y_i)$ (dans une *bonne paire*, il ne doit pas y avoir deux fois le même indice). Il y a alors $p(p-1)$ *bonnes paires* du type (x_i, x_j) , car $i = 1, \dots, p$ et $j = 1, \dots, i-1, i+1, \dots, p$. De même, il y a $p(p-1)$ *bonnes paires* du type (x_i, y_j) , (y_i, y_j) et (y_i, y_i) . Il y a donc $4p(p-1)$ *bonnes paires* au total.

On regroupe ensuite ces *bonnes paires* en $2p(p-1)$ couples $[(x_i, z); (y_i, z)]$.

Pour chacun de ces couples $[(x_i, z); (y_i, z)]$, on crée, comme dans le cas $p = 1$, λ graphes bipartis avec 2 sommets blancs et δ sommets noirs, où chaque sommet blanc est relié à tous les sommets noirs.

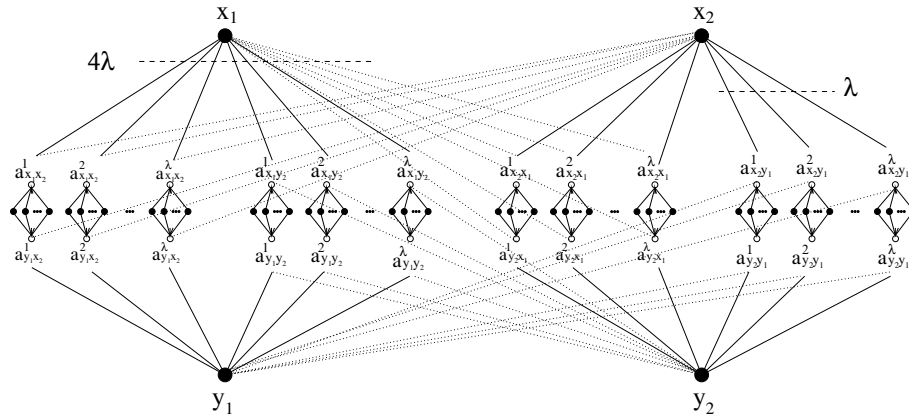
On prend alors un sommet blanc dans chacun de ces λ graphes. On nomme ces λ sommets $a_{x_i, z}^1, \dots, a_{x_i, z}^\lambda$. Les λ sommets blancs restants (1 dans chacun de ces graphes bipartis) sont alors appelés $a_{y_i, z}^1, \dots, a_{y_i, z}^\lambda$, de sorte que dans un de ces graphes bipartis, on ait le même indice j pour $a_{x_i, z}^j$ et $a_{y_i, z}^j$, $j = 1, \dots, \lambda$.

Enfin, chaque sommet noir z représentant un des terminaux est relié à tous les sommets $a_{z', z''}^j$ tels que $z = z'$ ou $z = z''$.

Remarque

Si $p = 3$, les $4 * 3 * (3 - 1) = 24$ bonnes paires sont :
 $(x_1, x_2), (x_1, x_3), (x_2, x_1), (x_2, x_3), (x_3, x_1), (x_3, x_2), (x_1, y_2), (x_1, y_3), (x_2, y_1),$
 $(x_2, y_3), (x_3, y_1), (x_3, y_2), (y_1, x_2), (y_1, x_3), (y_2, x_1), (y_2, x_3), (y_3, x_1), (y_3, x_2),$
 $(y_1, y_2), (y_1, y_3), (y_2, y_1), (y_2, y_3), (y_3, y_1)$ et (y_3, y_2) .

Cette construction devrait être plus claire grâce à l'exemple de la figure suivante, illustrant $H(\lambda, \delta, 2)$:



À ce stade du projet, nous avons donc entièrement fini la construction du graphe G . Nous devons donc vérifier qu'il possède bien toutes les caractéristiques requises. Pour cela, nous avons besoin d'en savoir plus sur le gadget H . Nous allons donc commencer par exposer quelques propriétés de H .

3.6 Propriétés du gadget

Nous allons maintenant voir pourquoi le gadget H est défini d'une manière si compliquée. H est en fait construit de cette façon afin de satisfaire la proposition suivante :

Proposition 3.6.1. Soit $\lambda = 2(m + 1)$ et $\delta = \delta_\lambda$:

$$\delta_\lambda = \begin{cases} \Delta(H) - \lambda - 1 & \text{si } p > 1 \\ \lambda/2 & \text{si } p = 1 \end{cases}$$

où $\Delta(H)$ est le degré maximum d'un sommet de H . Alors ces deux entiers positifs sont tels que pour $p \geq 2$, $2 < \delta + 2 < 4\lambda(p - 1)$ et pour $p = 1$, $2 < \delta + 1 < \lambda$. De plus, avec la construction ci-dessus, $H(\lambda, \delta, p)$ est un graphe biparti avec les propriétés suivantes :

H1 Les sommets noirs sont soit des terminaux, soit des sommets de degré deux. De plus pour chaque $z \in Z$, l'ensemble des terminaux, $d_H(z) = \Delta(H) = 4\lambda(p-1)$ si $p \geq 2$ et $d_H(z) = \Delta(H) = \lambda$ si $p = 1$.

H2 Si $p > 1$, H possède $4\lambda p(p-1)$ sommets blancs avec chacun un degré $\delta + 2$, et si $p = 1$, H possède 2λ sommets blancs avec chacun un degré $\delta + 1$.

H3 Il existe un entier t_0 , $p \leq t_0 \leq s$ (où s est le nombre de sommets qui sont soit des terminaux, soit des sommets blancs) déterminé par λ et p tel que H possède exactement 2^p t_0 -couvertures optimales et chacune d'elles contient exactement un des terminaux de chaque paire $\{x_i, x_j\}$, $j = 1, \dots, p$.

H4 Si U est la famille des t_0 -couvertures optimales de H et W est un ensemble de t_0 sommets qui n'appartient pas à U (i.e. pas optimal), alors :

$$q_{t_0}(H) - f_H(W) \geq \begin{cases} \lambda - 1 & \text{si } p > 1 \\ \lambda/2 & \text{si } p = 1 \end{cases}$$

H5 Pour $1 \leq r \leq l \leq s$, on a :

$$(l-r)\delta_\lambda \leq q_l(H) - q_r(H) \leq (l-r)\Delta(H)$$

Les propriétés $H1$ et $H2$ serviront à prouver que la réduction du problème BMax 2-Sat au problème BMVC s'effectue bien en un temps polynomial, comme nous le verrons au paragraphe 3.8 de ce même chapitre.

Les propriétés $H3$, $H4$ et $H5$ serviront quant à elles à prouver que ces deux problèmes sont effectivement équivalents. Plus précisément, ces propriétés seront nécessaires lors de la preuve de la proposition 3.9.1, au paragraphe 3.9 de ce même chapitre.

La preuve que H vérifie bien toutes ces conditions est très technique et n'apporte rien, du point de vue conceptuel, à la compréhension de la preuve qui nous intéresse. Nous allons donc accepter ces propriétés sans plus d'explications. Toutefois, le lecteur intéressé trouvera toutes ces vérifications dès la page 73 de la thèse de M. N. Appolonio [NA].

3.7 G est biparti

G est évidemment biparti, puisque dans sa construction, on a toujours relié un sommet blanc à un sommet noir et vice-versa.

3.8 La réduction est polynomiale

Montrons maintenant que la réduction du problème BMax 2-Sat au problème BMVC appliqué au graphe G se fait en un temps polynomial. Il s'agit donc de montrer qu'on construit G à partir d'une instance du problème BMax 2-Sat avec n variables et m clauses en un nombre d'opérations polynomial en n et m . On doit ainsi prouver que le nombre de sommets et le nombre d'arêtes de G sont bornés par un polynôme en n et m :

- **Sommets** : $|G| = |\tilde{G}| + |H| + |R| \leq 2nM + |H| + m$: dépend de $|H|$.
- **Arêtes** : Le nombre d'arêtes de G , sans compter les arêtes de H , est

$$nM^2 + k_1 + 2k_0 \leq n(\Delta(H) + m + 1)^2 + 2m$$

car $k_0 \leq k_0 + k_1 = k \leq m$ et par définition de M . Le nombre maximum d'arêtes dans H est $\Delta(H) * |H|$ (nombre de sommets * degré maximum d'un sommet). Donc le nombre d'arêtes de G dépend de $\Delta(H)$ et $|H|$.

Il reste donc à vérifier que $|H|$ et $\Delta(H)$ sont bornés par un polynôme en n et m . Nous traitons différemment ces vérifications selon que $p = 1$ ou que $p > 1$:

1. $p > 1$:

- $|H|$: Par la propriété H1 de la proposition 3.6.1, les sommets noirs de H sont les terminaux et les sommets de degré 2. Or il y a moins de $2m$ terminaux et, par construction de H il y a $\lambda * \delta * b$ sommets de degré 2, où b est le nombre de bonnes paires. Rappel :

$$\begin{aligned} b &= \#\{(x_i, x_j), (x_i, y_j), (y_i, y_j), (y_i, x_j), i, j = 1, \dots, p \quad i \neq j\} \\ &= 4p(p - 1) \end{aligned}$$

Donc, en posant n_r le nombre de sommets noirs, et comme $\delta < 4\lambda(p - 1) - 2$ on a :

$$n_r \leq \lambda * \delta * b + 2m \leq (4\lambda(p - 1) - 2) * \lambda * 4p(p - 1) + 2m$$

qui est borné par un polynôme en m puisque $p \leq m$ et $\lambda = 2(m + 1)$. Par la propriété H2 de la proposition 3.6.1, H contient $4\lambda p(p - 1)$ sommets blancs.

Donc le nombre de sommets blancs est aussi borné par un polynôme en m puisque $p \leq m$ et $\lambda = 2(m + 1)$.

- $\Delta(H)$: Par les propriétés H1 et H2 de la proposition 3.6.1,

$$\Delta(H) = 4\lambda(p - 1)$$

donc, pour les mêmes raisons qui ci-dessus, $\Delta(H)$ est aussi borné par un polynôme en m .

2. $p = 1$: Par les propriétés $H1$ et $H2$ de la proposition 3.6.1,

$$|H| = \lambda * \delta + 2\lambda + 2 \text{ et } \Delta(H) = \lambda$$

avec $\delta < \lambda - 1$. Donc $|H|$ et $\Delta(H)$ sont aussi bornés par un polynôme en m puisque $\lambda = 2(m + 1)$.

On a donc prouvé que l'on peut se ramener d'une instance du problème BMax 2-Sat à une instance du problème BMVC en un temps polynomial en n et m . Il ne nous reste donc plus qu'à vérifier que les problèmes BMax 2-Sat et BMVC appliqués au graphe G sont équivalents.

3.9 Equivalence entre BMVC dans G et BMax 2-Sat

Dans ce paragraphe, je vais montrer pourquoi le problème BMax 2-Sat et le problème BMVC appliqué au graphe G construit comme ci-dessus sont équivalents. Il s'agit donc de prouver que si l'on arrive à trouver une solution de BMVC dans G , alors on arrive à trouver une solution de BMax 2-Sat, et inversement, i.e. qu'il est possible de couvrir au moins X arêtes de G avec un nombre maximum B_0 de sommets fixés si et seulement s'il existe un assignement des variables du problème BMax 2-Sat satisfaisant au moins k clauses.

Les définitions suivantes et la proposition 3.9.1 seront utiles pour la proposition 3.9.2, qui nous donnera l'équivalence.

Définition

Un ensemble de sommets X est une (k, l) -couverture si X possède au maximum k sommets s'il est possible de toucher au minimum l arêtes avec les k sommets de X .

On note $q_k(G)$ le nombre maximum d'arêtes couvertes par k sommets dans un graphe G .

$f_G(X)$ désigne le nombre d'arêtes touchées par X dans G .

Proposition 3.9.1. *Soit G , le graphe défini par les règles **R1**, **R2** et **R3** à partir d'un problème BMax 2-Sat avec m clauses et n variables. Alors il existe un entier t_0 uniquement déterminé par n et m tel que, en posant $t = nM + t_0$ et $B_0 = nM^2 + p + q_{t_0}(H)$, U est une (t, B_0) -couverture si et seulement si :*

1. $U \cap G^i$ est soit A^i , soit B^i , pour $i = 1, \dots, n$

2. $|U \cap \{x_i, y_i\}| = 1$ pour $i = 1, \dots, p$ (exactement une terminale dans U pour chaque clause pure).
3. La restriction de la (t, B_0) -couverture au gadget H est une t_0 -couverture maximum de H , i.e. qu'avec t_0 sommets, on ne peut pas couvrir plus d'arêtes dans H que ne peut en couvrir U .

La démonstration de cette proposition est à nouveau très technique. C'est ici que sont employées les propriétés $H3$, $H4$ et $H5$ vérifiées par le gadget H . De plus ce résultat n'apporte pas de nouveau concept pour la preuve de la complexité du problème BMVC. Nous allons donc l'admettre. La démonstration complète se trouve cependant à la page 70 de [NA].

Par contre, la proposition suivante est très importante, puisqu'elle nous prouvera qu'un problème BMax 2-Sat peut se réduire à un problème BMVC. Sa démonstration nous donne de plus explicitement l'équivalence entre ces deux problèmes.

Proposition 3.9.2. *Soit G , le graphe défini par les règles **R1**, **R2** et **R3** du paragraphe 3.4 à partir d'un problème BMax 2-Sat avec m clauses et n variables. Alors il existe un assignement de 0 et 1 des n variables satisfaisant au moins k clauses si et seulement si il est possible de trouver une (t, B_0+k) -couverture dans G , avec t et B_0 définis comme dans la proposition ci-dessus.*

Démonstration. “ \implies ” : On suppose d'abord qu'il existe un assignement de 0 = faux et 1 = vrai satisfaisant au moins k clauses, k_0 clauses de J_0 et k_1 clauses de J_1 . Posons alors :

$$U_i = \begin{cases} A^i & \text{si la variable } u_i = 1 \\ B^i & \text{si la variable } u_i = 0 \end{cases} \quad \text{et } \tilde{U} = \cup_{i=1}^n U^i$$

Grâce à cet assignement, \tilde{U} couvre k_1 arêtes de E_1 et au moins k_0 arêtes de E_0 . En effet, si une clause $C = (c_1, c_2)$, où $c_i = u_i$ ou \bar{u}_i , est satisfaite, alors c_1 (ou/et c_2) est égal à 1, et donc c_1 (ou/et c_2 respectivement) appartient à \tilde{U} .

Posons encore $U' = \{z_1, \dots, z_p\} \cup U''$ où :

1. z_i est l'extrémité dans le gadget de l'arête de L_i non couverte par \tilde{U} si L_i contient exactement une arête non couverte.
2. z_i est l'extrémité dans le gadget d'une des arêtes de L_i (choisie arbitrairement) si \tilde{U} couvre aucune ou les deux arêtes de L_i .
3. $U'' \in V(H)$ est un ensemble avec $t_0 - p$ sommets blancs tel que $f_H(\{z_1, \dots, z_p\} \cup U'') = q_{t_0}(H)$.

Posons enfin $U = \tilde{U} \cup U'$.

Alors U est une $(t, B_0 + k)$ -couverture dans G . En effet :

1. $|U| = |\tilde{U}| + |\{z_1, \dots, z_p\}| + |U''| = nM + p + (t_0 - p) = nM + t_0 = t$
par définition de t_0 .
2. $f_G(\tilde{U}) = nM^2 + k_1 + k_0 + p'$ où p' est le nombre de clauses $(1, 1)$,
i.e. le nombre de clauses comptées deux fois.
 $f_G(U') = q_{t_0}(H) + p$, on ajoute les p arêtes couvertes par les
 $\{z_1, \dots, z_p\}$.
 $f_G(U) \leq f_G(\tilde{U}) + f_G(U')$, \leq car une arête correspondant à une
clause $(1, 1)$ est comptée deux fois dans la partie de droite. Donc

$$\begin{aligned}
f_G(U) &= f_G(\tilde{U}) + f_G(U') - p' \\
&= (nM^2 + k_1 + k_0 + p') + (q_{t_0}(H) + p) - p' \\
&= (nM^2 + q_{t_0}(H) + p) + (k_1 + k_0) \\
&= B_0 + k
\end{aligned}$$

“ \Leftarrow ” : Soit U , une $(t, B_0 + k)$ -couverture dans G . Alors par la proposition 3.9.1, comme U couvre au moins B_0 arêtes, $U \cap V(G^i)$ est soit A^i , soit B^i , pour $i = 1, \dots, n$ et $|U \cap \{x_i, y_i\}| = 1$ (*) pour $i = 1, \dots, p$. De plus, toujours par la proposition 3.9.1, la restriction de la (t, B_0) -couverture au gadget H est une t_0 -couverture maximum de H , i.e. le nombre d'arêtes touchées par U entièrement contenues dans H est $q_{t_0}(H)$. Posons alors :

$$u_i = \begin{cases} 1 & \text{si } U \cap V(G^i) = A^i \\ 0 & \text{si } U \cap V(G^i) = B^i \end{cases}$$

Avec cette affectation, toutes les clauses correspondant aux arêtes représentant ces dernières et touchées par U contiennent un 1 (en effet, soit une des extrémités est \bar{u}_i avec $u_i = 0$, soit une des extrémités est u_i avec $u_i = 1$). Donc toute les arêtes représentant des clauses et touchées par U représentent des clauses satisfaites. Il reste donc à vérifier que ces arêtes sont au nombre de k .

Ces arêtes sont les $f_G(U)$ arêtes de G touchées par U , moins les M^2 arêtes de chaque G^i , $i = 1, \dots, n$ touchées par U (puisque soit A^i , soit B^i est inclu dans U pour tout i), moins les $q_{t_0}(H)$ arêtes touchées par U ayant leurs deux extrémités dans H , moins enfin les p arêtes couvertes par les terminaux contenus dans U (au nombre de p par *). En résumé, le nombre de clauses satisfaites est :

$$f_G(U) - nM^2 - q_{t_0}(H) - p$$

Or, par définition de B_0 :

$$f_G(U) - nM^2 - q_{t_0}(H) - p = f_G(U) - B_0$$

De plus, par hypothèse, U couvre $B_0 + k$ arêtes au minimum, i.e. :
 $f_G(U) \geq B_0 + k$, ou encore $k \leq f_G(U) - B_0$.
Ainsi le nombre de clauses satisfaites est :

$$f_G(U) - nM^2 - q_{t_0}(H) - p = f_G(U) - B_0 \geq k$$

Donc au moins k clauses sont satisfaites.

□

Voilà donc l'équivalence entre le problème BMax 2-Sat et le problème BMVC appliqué au graphe G prouvée.

De plus, cette dernière implication (" \Leftarrow ") donne la façon de retrouver une solution du problème BMax 2-Sat (i.e. une affectation des variables satisfaisant au moins k clauses). Ainsi la transformation d'une solution du problème BMVC appliqué à G en une solution du problème BMax 2-Sat s'effectue en un temps polynomial.

Cela termine la preuve de la \mathcal{NP} -complétude de BMVC.

Chapitre 4

Améliorations possibles de la preuve

Il me paraît difficile d'améliorer la structure et le déroulement de la preuve de la \mathcal{NP} -complétude de BMVC. Il y a par contre dans la thèse de Nicola Appolonio [NA] quelques erreurs de notation, confusions d'indices et fautes de calculs. Cela rend la lecture de la preuve relativement longue et compliquée. Je vais donc dans ce chapitre proposer, hormis quelques notations plus adéquates, une liste des erreurs que j'ai pu relever. J'espère que cela facilitera la lecture de la preuve en question pour les prochaines personnes s'y intéressant.

Tous les points ci-dessous se réfèrent à la thèse de Nicola Appolonio : [NA].

4.1 Problèmes de notation

1. Les variables du problème BMax 2-Sat sont parfois notées $U = \{u_i\}$ et parfois $x = \{x_i\}$. Nicola Appolonio saute d'une notation à l'autre sans raisons apparentes.
2. Les clauses du problème BMax 2-Sat sont parfois notées $C = \{u_i, u_j\}$ et parfois $C = (u_i, u_j)$. Ces deux notations représentent la même chose. $\{u_i, u_j\}$ ne représente pas un ensemble, mais bien un couple. J'ai donc adopté la notation $C = (u_i, u_j)$.
3. Les parties G^i du graphe G sont parfois utilisées pour désigner $V(G^i)$, les sommets de G^i , bien que la distinction est souvent faite (la notation $V(G^i)$ est utilisée).
4. Le gadget a lui aussi deux notations possibles H et G^0 qui sont totalement équivalentes. G^0 représente parfois $H - Z$ où Z est l'ensemble des terminaux, mais pas toujours, d'où quelques confusions et difficultés de lecture. Par exemple, au fond de la page 67 de [NA], $V(G^0)$ représente $V(H - Z)$. Dans ce rapport, j'ai gardé la notation H pour

le gadget et G^0 pour $H - Z$.

5. Plusieurs notations sont aussi utilisées pour le degré maximum d'un sommet de H : $\Delta = \Delta^0 = \Delta(H) = \Delta(G^0)$.
6. Pour la construction du gadget, l'ensemble des λ sommets pour chaque bonne paire est noté Z , comme l'ensemble des terminaux, mais ces deux ensembles n'ont rien à voir.

4.2 Erreurs de calcul et d'indices

1. Pour la construction de G ([NA] p. 65), dans la règle R1, les sommets noirs (blancs respectivement) sont $a_{i,j}$ ($b_{i,j}$ respectivement), et pas $a_{i,1}$ ($b_{i,1}$ respectivement).
2. Pour la construction de G ([NA] p. 65), dans la règle R2, $e_j = (a_{i,j}, b_{k,j})$ et pas $e_j = (a_{i,j}, b_{i,j})$, et plus loin $e_j = (a_{i,j}, w_j)$ ($e_j = (b_{i,j}, w_j)$ respectivement) et pas $e_j = (a_{i,1}, w_j)$ ($e_j = (b_{i,1}, w_j)$ respectivement). De plus il manque ici la règle suivante :

$$c_j = (\bar{u}_i, u_k) \implies e_j = (b_{i,j}, a_{k,j})$$

3. Pour la construction de G ([NA] p. 65), dans la formule (4.6.1), $E_1 = \{e_j : j \in J_1\}$ et pas $E_0 = \{e_j : j \in J_1\}$.
4. Il y a plusieurs erreurs dans la preuve de l'équivalence entre les problèmes BMVC et BMax 2-Sat, [NA] p. 68 :
 - (a) À la 3ème ligne, $f_G(X) \leq B_0 + (k_0 + k_1)$, et pas " $=$ ". Cette erreur se retrouve à la fin de la preuve, on a en effet $k \leq f_G(X) - B_0$ et pas $k = f_G(X) - B_0$. Mais la conclusion de la preuve n'en est qu'à plus forte raison vraie.
 - (b) À la 5ème ligne, l'indice j va de 1 à m , mais il s'agit des terminaux, j doit donc aller de 1 à p .
 - (c) À la 6ème ligne, l'auteur a oublié $q_{t_0}(H)$, les arêtes couvertes par U ayant leurs deux extrémités dans H . Il oublie de nouveau $q_{t_0}(H)$ à la dernière ligne de la preuve, lorsqu'il soustrait B_0 de $f_G(U)$ (qui remplace $f_G(X)$, encore un exemple de confusion dans les notations). Ces deux erreurs se compensent, le résultat n'est donc point affecté.
 - (d) Dans la définition de E_1 à la ligne 7, il manque les arêtes aboutissant dans l'ensemble R des clauses à une seule littérale. Cela n'affecte pas non plus la preuve, puisque ces arêtes seront comptées dans k_1 dans la fin de la preuve.

Toujours dans cette même ligne, le deuxième indice i devrait être un indice j . On devrait avoir en fin de compte :

$$E'_1 = \cup_{i \neq j; i, j=1}^n E_G(V(G^i), V(G^j)) + E_G(V(\tilde{G}), V(R))$$

5. Dans la propriété $H4$ de la page 69 de [NA] (formule 4.6.5), $\dots = \lambda/2$ si $p = 1$, et pas si $p = 2$.
6. Toujours à la même page, on dit que l'ordre de G (son nombre de sommets) est $nM + n_0$. En fait l'ordre de G est $2nM + n_0 + |R|$.
7. Dans la formule (4.7.2) de la page 73 de [NA], $f_2 = 2\lambda + (\lambda - 1)\delta$, et juste dessus, $q_\theta(H) = \max\{f_0, f_1, f_2\}$. De plus dans la formule (4.7.3), $f_1 - f_0 + 1 = \lambda - \delta$.

Cette liste n'est évidemment pas exhaustive, il est possible que d'autres erreurs aient échappé à mon attention.

Conclusion

Même si nous savons depuis longtemps que le problème de couverture de sommets dans les graphes bipartis (BVC) se résout en un temps polynomial, nous ne savions pas quelle était la complexité du problème de couverture maximum de sommets restreint aux graphes bipartis (BMVC). Intuitivement, comme ce problème est proche du problème BVC, nous aurions pu supposer que la complexité de BMVC était polynomiale.

Cependant, bien que nous n'ayons pas pu nous ramener aux nombreux résultats connus sur les problèmes de suppression de sommets, nous avons réussi à prouver que BMVC est \mathcal{NP} -complet, en nous servant du problème BMax 2-Sat, que l'on savait \mathcal{NP} -complet.

Bien que complète, la preuve en question comporte beaucoup d'erreurs. On pourrait y apporter de nombreuses améliorations, ou même la réécrire dans sa totalité. Peut-être pour un futur projet ...

Il ne me reste plus qu'à remercier le Prof. D. de Werra qui a accepté que je fasse ce projet dans sa chaire et l'assistante Tinaz Ekim, qui m'a suivi tout au long du projet. Je remercie aussi toutes les personnes qui m'ont aidé pour la rédaction, ainsi que toutes les personnes qui ont lu ce projet pour leur attention.

Bibliographie

[NA]

APPOLONIO, NICOLA (2002) *“The Max Bipartite Vertex Cover Problem”*, chapitre 4 de la thèse de M. Nicola Appolonio intitulée *“Some Constrained Covering Problems in Graphs.”* (Université de Rome, “La Sapienza”)

[MY]

YANNAKAKIS, MIHALIS (1981) *“Node-deletion problems on bipartite graphs”*, Siam Journal of Computing, 10,2, Bell Laboratories, Murray Hill (New Jersey, USA)

[TE]

EKIM, TINAZ (2005) *“Théorie de la complexité”*, cours d'introduction à la recherche opérationnelle. Chair de Recherche Opérationnelle Sud-Est (FSB-IMA-ROSE, EPFL)