



**HAL**  
open science

# Réduction de l'espace de recherche du Problème de la Somme Coloration Minimum d'un graphe

Clément Lecat

► **To cite this version:**

Clément Lecat. Réduction de l'espace de recherche du Problème de la Somme Coloration Minimum d'un graphe. Autre [cs.OH]. Université de Picardie Jules Verne, 2017. Français. NNT : 2017AMIE0033 . tel-03692885

**HAL Id: tel-03692885**

**<https://theses.hal.science/tel-03692885>**

Submitted on 10 Jun 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**Thèse de Doctorat**  
*Mention Informatique*  
*Spécialité Recherche opérationnelle et optimisation*

présentée à *l'Ecole Doctorale en Sciences Technologie et Santé (ED 585)*

**de l'Université de Picardie Jules Verne**

par

**Clément LECAT**

pour obtenir le grade de Docteur de l'Université de Picardie Jules Verne

***Réduction de l'espace de recherche du Problème de la  
Somme Coloration Minimum d'un graphe***

Soutenue le 27/11/2017, après avis des rapporteurs, devant le jury d'examen :

|  |                   |
|--|-------------------|
| <b>M. J.K HAO, Professeur des Universités</b>                        | <b>Rapporteur</b> |
| <b>M<sup>me</sup> C. SOLNON, Professeur des Universités</b>          | <b>Rapporteur</b> |
| <b>M<sup>me</sup> L. BRISOUX-DEVENDEVILLE, Maître de Conférences</b> | <b>Examineur</b>  |
| <b>M. F. MANYÀ, Directeur de Recherche</b>                           | <b>Examineur</b>  |
| <b>M. A. MOUKRIM, Professeur des Universités</b>                     | <b>Examineur</b>  |
| <b>M. B. MAZURE, Professeur des Universités</b>                      | <b>Examineur</b>  |
| <b>M. C.M LI, Professeur des Universités</b>                         | <b>Directeur</b>  |
| <b>M<sup>me</sup> C. LUCET, Maître de Conférences</b>                | <b>Directeur</b>  |



# Remerciements

Ces trois années furent pour moi une expérience inoubliable me permettant de vivre des moments et des sentiments que seule une thèse peut vous faire vivre. Cela passe de la frustration à la joie, de la fatigue à la plénitude, etc. Durant ces années, j'ai pu faire fait énormément de rencontres qui m'ont marqué. Je ne pourrais malheureusement pas citer tout le monde. Je demande à ceux qui n'apparaissent pas dans les lignes qui vont suivre de ne pas m'en vouloir, le cœur y est !

Mes premiers remerciements sont adressés à mes directeurs de thèse Chu-Min et Corinne qui m'ont porté leur confiance et m'ont donné la chance de réaliser ces travaux. À leur côté, j'ai énormément appris, aussi bien scientifiquement qu'humainement. Je les remercie de m'avoir donné la possibilité de participer à différentes aventures « extra-thèse » telles que la semaine du cerveau, qui m'ont permis de m'épanouir. Je dois également saluer leur patience et leur disponibilité. Ils ont toujours su libérer de leur temps pour répondre à mes interrogations et pour me guider. Merci à vous.

Je tenais ensuite à exprimer ma profonde gratitude à Jin-Kao Hao et Christine Solnon d'avoir accepté de rapporter ces travaux de thèse en un temps si restreint. Un grand merci également à Laure Brisoux-Devendeville, Felip Manyà, Aziz Moukrim et Bertrand Mazure d'avoir accepté d'être membres du jury de cette thèse. Il m'est cependant nécessaire d'avoir une pensée particulière pour Laure et Bertrand : Laure avec qui j'ai pu énormément échanger et Bertrand pour m'avoir fait découvrir le monde passionnant de la recherche durant mes années de master.

L'un des facteurs qui a également permis le bon déroulement de cette thèse est le cadre dans lequel elle s'est déroulée. Au sein du MIS, je me suis pleinement épanoui du fait de l'accueil chaleureux et de la bonne ambiance y régnant. Je remercie l'ensemble des permanents, des membres de l'équipe GOC, plus particulièrement Céline et Gilles. Un grand merci à Yu, avec qui j'ai pu avoir de longues conversations philosophiques sur l'IA, et qui m'a également fait découvrir de nouvelles saveurs venues de Chine. Un grand merci aux actuels et anciens membres de l'équipe administrative, notamment Juliette et Valérie, "the best" pour reprendre une citation. Et enfin, merci à l'ensemble des doctorants, ingénieurs et stagiaires que j'ai pu côtoyer, Doha, Nathan, Nour Eddine, Nicolas, Saïd, Thawsif, Thibault, Zaynab, etc. Grosse pensée aux anciens du bureau 318, Monsieur Baptiste, Rachid, etc, et aux actuelles actuels du bureau devenu 418, mon frère Haïtien Richie, Anass, Monica, Simon et Adrien. Gros clin d'oeil à Olivier avec qui j'ai pu visiter la belle ville de Montreuil-sur-Mer et observer la vue magnifique des remparts.

Puis il y a les amis, ceux avec qui je partage pas mal de bons moments depuis maintenant une paire d'années : Damien, Marc, Nicolas A., Nicolas S., Sélim, Thomas... sans oublier Amélie et Jérôme.

Bien entendu, je suis obligé de garder une place particulière pour mes deux compères Jordan et Romuald qui m'ont toujours suivi, même si dans certains cas, je ne me serais peut-être pas suivi moi-même, et avec qui j'ai passé de très bons moments. Merci les amis !

Je garde les quelques lignes qu'il me reste pour ma famille qui m'a toujours soutenu. Malgré mon manque de présence, ils ne m'en ont jamais tenu rigueur et je dois les en remercier. Djedi, mamie li, mamie let, papi, Lucette, parrain, marraine et tous les autres... merci à vous, je vous embrasse de tout mon cœur.

Maman, papa, popo, vous avez toujours été là pour moi. Vous m'avez guidé et laissé libre de faire mes propres choix. Vos conseils et votre amour ont été durant ces années importants pour moi. Je suis devenu celui que je suis en partie grâce à vous et je vous en remercie. J'espère vous rendre fier de moi.

Pour terminer, il me reste une personne à remercier, une personne qui me donne envie chaque jour de me surpasser. Je ne lui dis pas assez, et pourtant, je lui dois beaucoup. Elle a su me soutenir au quotidien, dans les moments pas faciles et pour beaucoup d'autre chose. Merci à toi, Déborah. Merci pour tout ce que tu fais ma bien-aimée.

Si par le plus grand des hasards, quelqu'un vient à lire entièrement ces lignes, je l'en remercie sincèrement, car une thèse ce n'est pas qu'un seul homme, mais c'est un tout, auquel il est nécessaire de rendre hommage.





# Table des matières

|  |            |
|--|------------|
| <b>Remerciements</b>   | <b>i</b>   |
|  | <b>iii</b> |
| <b>Table des matières</b>  | <b>v</b>   |
| <b>Liste des figures</b>   | <b>vii</b> |
| <b>Liste des tableaux</b>  | <b>ix</b>  |
| <b>Introduction</b>  | <b>1</b>   |
| <b>1 De la Complexité au Problème de la Somme Coloration Minimum d'un Graphe</b>                             | <b>3</b>   |
| 1.1 Classes de complexité  | 3          |
| 1.2 De la logique propositionnelle au problème SAT   | 7          |
| 1.3 Introduction aux graphes   | 10         |
| 1.4 Problèmes de coloration  | 16         |
| 1.5 Conclusion   | 22         |
| <b>2 Résolution du Problème de la Somme Coloration Minimum d'un Graphe</b>                                   | <b>23</b>  |
| 2.1 Approche non exhaustive pour la résolution de MSCP   | 24         |
| 2.2 Approche complète pour MSCP  | 31         |
| 2.3 Benchmark  | 37         |
| 2.4 Conclusions  | 40         |
| <b>3 Nouvelles Bornes pour le Problème de la Somme Coloration Minimum d'un Graphe</b>                        | <b>41</b>  |
| 3.1 Coloration majeure et motif  | 42         |
| 3.2 Nouvelles bornes pour la force d'un graphe   | 47         |
| 3.3 Expérimentations de $UB_A$ et de $UB_S$  | 54         |
| 3.4 $LBM_\Sigma$ une nouvelle borne pour la somme coloration   | 58         |
| 3.5 Conclusions  | 62         |
| <b>4 Un Nouveau Paradigme de Résolution du Problème de la Somme Coloration Minimum d'un Graphe</b>           | <b>63</b>  |
| 4.1 Encodage vers MaxSAT   | 63         |
| 4.2 Encodage vers MinSAT   | 67         |
| 4.3 Diminution du nombre de variables  | 70         |
| 4.4 Étude des performances des encodages MaxSAT et MinSAT  | 81         |
| 4.5 Conclusions  | 85         |
| <b>5 Deux Approches de type Branch-and-Bound pour le Problème de la Somme Coloration Minimum d'un Graphe</b> | <b>87</b>  |
| 5.1 Branch-and-Bound   | 88         |
| 5.2 BBMSCP   | 89         |
| 5.3 3LMSCP   | 93         |

|                            |            |
|----------------------------|------------|
| 5.4 Conclusions .....      | 97         |
| <b>Conclusion Générale</b> | <b>99</b>  |
| <b>Index</b>               | <b>101</b> |

# Liste des figures

|      |   |    |
|------|---|----|
| 1.1  | Représentation du contenu d'un ruban d'une machine de Turing  | 4  |
| 1.2  | Diagramme représentant une structuration possible des classes de complexité où $P \neq NP$ .  | 7  |
| 1.3  | Tables de vérité de la logique propositionnelle   | 9  |
| 1.4  | Représentation d'un graphe $G = (V, E)$ .   | 10 |
| 1.5  | Graphe complet.   | 11 |
| 1.6  | Graphe partiel du graphe Figure 1.4.  | 11 |
| 1.7  | Sous-graphe du graphe Figure 1.4 induit par $V' = \{a, b, c, d\}$ .   | 11 |
| 1.8  | Graphe complémentaire du graphe Figure 1.4.   | 12 |
| 1.9  | Un arbre.   | 13 |
| 1.10 | Graphe biparti.   | 14 |
| 1.11 | 3-arbre.  | 14 |
| 1.12 | 3-arbre partiel du graphe Figure 1.11.  | 14 |
| 1.13 | Graphe d'intervalles.   | 14 |
| 1.14 | Graphe planaire.  | 15 |
| 1.15 | Graphe $P_4$ .  | 15 |
| 1.16 | Graphes $P_4$ -sparse.  | 15 |
| 1.17 | Split-graphe.   | 15 |
| 1.18 | Une coloration valide du graphe de la Figure 1.4.   | 16 |
| 1.19 | Une solution optimale pour <i>GCP</i> .   | 18 |
| 1.20 | Graphe parfait.   | 19 |
| 1.21 | Une solution optimale pour le problème <i>OCCP</i> .  | 19 |
| 1.22 | Une solution optimale pour le problème <i>pSMC</i> et <i>npSMC</i> .  | 21 |
| 1.23 | Une solution optimale pour le problème <i>MSCP</i> .  | 21 |
| 2.1  | Paysage d'une méthode de recherche locale.  | 28 |
| 2.2  | Modélisation d'un graphe $G = (V, E)$ .   | 32 |
| 2.3  | Un réseau de contraintes pour le problème <i>MSCP</i> .   | 34 |
| 2.4  | Coût associé aux différentes fonctions de coût de l'Exemple 24.   | 36 |
| 3.1  | Une solution optimale de <i>GCP</i> , $X = \{\{a, b, c, e\}_1, \{d, f, g, h\}_2\}$ où $\Sigma(X) = 12$ , et une solution optimale de <i>MSCP</i> , $X' = \{\{a, b, c, f, g, h\}_1, \{d\}_2, \{e\}_3\}$ où $\Sigma(X') = 11$ . | 42 |
| 3.2  | Nuage de points comparant les bornes $UB_{hmt}$ et $UB_A$ .   | 56 |
| 3.3  | Nuage de points comparant les bornes $UB_{hmt}$ et $UB_S$ .   | 57 |
| 3.4  | Partionnement d'un graphe $G = (V, E)$ en cliques avec $\Sigma(G) \geq 10$ .  | 59 |
| 3.5  | Nuage de points comparant les bornes $LB_\Sigma$ et $LBM_\Sigma$ .  | 61 |
| 4.1  | Graphe $G = (V, E)$ .   | 66 |
| 4.2  | Nuage de points comparant le taux de réduction des graphes de type Allocation dont la densité moyenne est égale à 0.149.  | 75 |
| 4.3  | Nuage de points comparant le taux de réduction des instances de type DSJC dont la densité moyenne est égale à 0.499.  | 76 |

|      |  |    |
|------|--|----|
| 4.4  | Nuage de points comparant le taux de réduction des instances de type Flat dont la densité moyenne est égale à 0.486. . . . .   | 76 |
| 4.5  | Nuage de points comparant le taux de réduction des instances de type Leighton dont la densité moyenne est égale à 0.108. . . . .   | 77 |
| 4.6  | Nuage de points comparant le taux de réduction des instances de type Mycielski dont la densité moyenne est égale à 0.232. . . . .  | 77 |
| 4.7  | Nuage de points comparant le taux de réduction des instances de type Queen dont la densité moyenne est égale à 0.311. . . . .  | 78 |
| 4.8  | Nuage de points comparant le taux de réduction des instances de type Miles dont la densité moyenne est égale à 0.292. . . . .  | 78 |
| 4.9  | Nuage de points comparant le taux de réduction entre un encodage classique ( $\Delta(G)+1$ ) avec un encodage respectant la propriété de Minot (Propriété 24) et la densité d'un graphe. . . . . | 80 |
| 4.10 | Nuage de points comparant le taux de réduction entre un encodage classique ( $\Delta(G)+1$ ) avec un encodage augmenté (Propriété 25) et la densité d'un graphe. . . . .                         | 80 |
| 4.11 | Nuage de points comparant le taux de réduction entre un encodage respectant la propriété de Minot (Propriété 24) avec un encodage augmenté (Propriété 25) et la densité d'un graphe. . . . .     | 81 |
| 5.1  | Graphe $G = (V, E)$ . . . . .  | 88 |
| 5.2  | Arbre de recherche associé à une méthode énumérative pour un problème de 3-coloration. . . . .   | 88 |
| 5.3  | Modélisation des couleurs disponibles d'un graphe $G = (V, E)$ pour $k = 3$ tel que $c(B) = 2$ . . . . .   | 90 |
| 5.4  | Modélisation d'une décomposition en cliques de la Figure 5.3 après coloration du sommet $b$ suivant une approche <i>CliquePart</i> . . . . .   | 92 |
| 5.5  | Arbre de recherche associé à 3LMSCP pour un problème de 3-coloration (Figure 5.1). . . . .   | 94 |

# Liste des tableaux

|     |  |    |
|-----|--|----|
| 2.1 | Graphes de Mycielski . . . . .   | 37 |
| 2.2 | Graphes SGB . . . . .  | 38 |
| 2.3 | Graphes DSJC . . . . .   | 38 |
| 2.4 | Graphes de Leighton . . . . .  | 39 |
| 2.5 | Graphes d'allocation . . . . .   | 39 |
| 2.6 | Graphes plats . . . . .  | 40 |
| 3.1 | Cardinalité de $\phi(n)$ . . . . .   | 44 |
| 3.2 | $\phi(8)$ . . . . .  | 46 |
| 3.3 | Comparaison de notre borne algébrique ( $UB_A$ ) et notre borne algorithmique ( $UB_S$ ) face à la borne proposée dans <a href="#">Hajiabolhassan et al. (2000)</a> ( $UB_{hmt}$ ) . . . . .                               | 55 |
| 3.4 | Comparaison de notre borne algébrique ( $LBM\Sigma$ ) avec les meilleures bornes inférieures connues de la littérature pour la somme chromatique. . . . .  | 61 |
| 4.1 | Comparaison de la taille des instances générées. . . . .   | 74 |
| 4.2 | Temps de résolution en seconde des différents encodages sur les instances <i>COLOR</i> ( <b>COLOR</b> ) et <i>DIMACS</i> ( <b>Dimacs</b> ). . . . .  | 82 |
| 4.3 | Moyenne des temps de résolution en seconde des différents encodages sur un ensemble d'instances aléatoires. . . . .  | 83 |
| 4.4 | Comparaison des temps de résolution en seconde des différents encodages du problème <i>weighted partial MaxSAT</i> face aux meilleurs temps connus de la littérature pour un jeu d'instances. . . . .                      | 85 |
| 5.1 | Comparaison des temps de résolution en secondes du solveur <i>BBMSCP</i> et du solveur <i>ISAC</i> pour les instances <i>COLOR</i> ( <b>COLOR</b> ) et <i>DIMACS</i> ( <b>Dimacs</b> ). . . . .                            | 92 |
| 5.2 | Comparaison du temps moyen de résolution en secondes du solveur <i>BBMSCP</i> et du solveur <i>ISAC</i> pour un ensemble d'instances aléatoires. . . . .   | 93 |
| 5.3 | Comparaison des temps de résolution en secondes du solveur <i>3LMSCP</i> , du solveur <i>BBMSCP</i> et du solveur <i>ISAC</i> pour les instances <i>COLOR</i> ( <b>COLOR</b> ) et <i>DIMACS</i> ( <b>Dimacs</b> ). . . . . | 95 |
| 5.4 | Comparaison du temps moyen de résolution en secondes du solveur <i>3LMSCP</i> , du solveur <i>BBMSCP</i> et du solveur <i>ISAC</i> pour un ensemble d'instances aléatoires. . . . .  | 97 |



# Introduction

En quelques décennies, l'informatique s'est développée et a intégré notre société au point où un lendemain sans numérique est devenu difficilement envisageable. Que se soit dans les domaines de la santé, des transports, de la finance et bien d'autres encore, l'informatique s'est imposée. Cela nécessite une évolution constante des outils technologiques afin de pouvoir gérer des problèmes de plus en plus complexes. La théorie des graphes a permis de modéliser certains de ces problèmes et de leur apporter des solutions par le développement d'algorithmes spécifiques, continuellement améliorés depuis plusieurs décennies par les travaux de nombreux chercheurs. Le problème de coloration de graphe sur lequel s'appuie l'étude de cette thèse est un problème intensivement étudié dans la littérature et à l'origine de nombreux problèmes dérivés comme celui de la somme coloration minimum.

Le Problème de la Somme Coloration Minimum (*MSCP*) d'un graphe est un problème d'optimisation combinatoire introduit dans [Kubicka et Schwenk \(1989\)](#). Contrairement à un problème de décision dont l'objectif est de déterminer si une solution existe, l'objectif d'un problème d'optimisation est de déterminer la meilleure solution selon certains critères, parmi l'ensemble des solutions possibles du problème. Pour *MSCP*, l'objectif est de déterminer une coloration valide d'un graphe minimisant la somme des poids associés aux couleurs utilisées. Le nombre minimum de couleurs dans une solution optimale de *MSCP* est appelé la force du graphe, et la somme des poids associés aux couleurs utilisées dans une solution optimale de *MSCP* est appelée la somme chromatique du graphe. En pratique, *MSCP* permet notamment de répondre au problème de gestion de flux (population, réseau, transport, *etc.*) pour laquelle *MSCP* a la particularité d'apporter une solution améliorant la qualité de service en minimisant le délai moyen d'attente. Par exemple, considérons une couleur comme étant le temps d'attente à l'accès d'une ressource pour un groupe de personne. Minimiser le temps d'attente maximum ne garantit pas l'obtention d'un temps d'attente moyen minimum. En effet, défavoriser un petit groupe de personne en augmentant leur temps d'attente peut permettre de réduire l'attente moyenne et par conséquent satisfaire un plus grand nombre de personnes.

De nombreuses études ont été menées autour de *MSCP*. Certains de ces travaux ont étudié *MSCP* sur des familles spécifiques de graphes afin d'en extraire un ensemble de bornes permettant une réduction de l'espace des solutions ou de proposer des méthodes de résolution efficaces ([Kubicka et Schwenk \(1989\)](#), [Halldórsson et al. \(2001\)](#), *etc.*). À notre connaissance, pour un graphe arbitraire, la plus grande partie des méthodes développées permettant une résolution exacte du problème repose sur une modélisation de *MSCP* en un autre paradigme de recherche tel que la programmation par contrainte ([Minot et al. \(2016a\)](#)), la programmation linéaire ([Sen et al. \(1992\)](#)), *etc.* Cependant du fait de la complexité de *MSCP*, la majorité des travaux réalisés pour *MSCP* sont des approches incomplètes à base de recherches locales ([Benlic et Hao \(2012\)](#), [Helmar et Chiarandini \(2011\)](#), *etc.*), d'algorithmes évolutionnaires ([Moukrim et al. \(2014\)](#), [Jin et al. \(2014\)](#), *etc.*), pour l'obtention d'une solution de bonne qualité en un temps raisonnable.

Les travaux rapportés dans ce manuscrit ont pour but de développer de nouveaux outils aidant à la résolution de *MSCP*. Ils se scindent en deux catégories. La première s'intéresse à la structuration d'une coloration d'un graphe afin d'extraire de l'information et d'introduire de nouvelles

bornes permettant la réduction de l'espace des solutions de *MSCP*. Pour cela, nous nous sommes intéressés à la notion de motifs introduit dans [Bonomo et Valencia-Pabon \(2009, 2014\)](#) pour la famille des graphes  $P_4$  sparses. Après avoir adapté leur approche et introduit un ensemble de propriétés sur les motifs, nous avons élaboré deux nouvelles bornes supérieures de la force d'un graphe, appelées  $UB_A$  et  $UB_S$ . La borne  $UB_A$  est une borne algébrique alors que  $UB_S$  est une borne algorithmique prenant en considération la notion de stable maximum d'un graphe. Les résultats expérimentaux ont montré que ces approches réduisaient significativement la force du graphe en comparaison des meilleures bornes connues de la littérature. Les motifs ont également permis l'introduction d'une borne inférieure algébrique ( $LBM_\Sigma$ ) de la somme chromatique qui pour quelques familles de graphes, améliore significativement les résultats de la littérature. De plus, cette approche prouve l'optimalité de la meilleure solution connue pour 4 instances de la littérature. Ces travaux ont mené à un ensemble de publications [Lecat et al. \(2017a,b,c,e\)](#) et ont été récompensés par le prix Jeune Chercheur lors de la 18ème conférence de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (*ROADEF 2017*).

La seconde partie de nos travaux s'intéresse à la résolution exacte de *MSCP*. La première approche que nous avons introduite et étudiée, est une modélisation de *MSCP* en un problème *weighted partial MaxSAT* et en un problème *weighted partial MinSAT* ([Lecat et al. \(2015a,b\)](#)). Les formalismes *MaxSAT/MinSAT* sont bien connus et extrêmement étudiés. Dans le cas du formalisme *MaxSAT*, une compétition récompensant les meilleurs solveurs de résolution est organisée chaque année. L'intérêt de notre approche est d'apporter une modélisation de *MSCP* vers ces problèmes afin d'exploiter la puissance de ces solveurs pour la résolution du problème. Pour cela, nous avons considéré et renforcé la propriété proposée dans [Minot et al. \(2017\)](#) en utilisant notre borne de la force  $UB_S$  afin de proposer deux encodages de *MSCP* en un problème *weighted partial MaxSAT* et deux encodages de *MSCP* en un problème *weighted partial MinSAT*. Les résultats expérimentaux ont montré que le solveur *ISAC* ([Ansótegui et al. \(2016\)](#)) était particulièrement adapté à la résolution des instances *weighted partial MaxSAT*.

La deuxième approche que nous avons proposée se base sur une méthode de résolution de type *Branch-and-Bound*. Nos contributions consistent en l'élaboration de deux solveurs appelés *BBMSCP* et *3LMSCP* ([Lecat et al. \(2015a,b\)](#)). Le solveur *BBMSCP* est une méthode énumérative pour laquelle nous avons développé les bornes  $LB_{B_\Sigma}$  et  $LBP_{B_\Sigma}$ . La borne  $LB_{B_\Sigma}$  considère les plus petites couleurs disponibles pour les sommets non colorés afin de déterminer une borne inférieure de la somme chromatique. Pour la borne  $LBP_{B_\Sigma}$ , nous considérons un partitionnement du sous-graphe composé des sommets non colorés en une partition de cliques afin de déterminer une borne inférieure de la somme chromatique. Ces bornes permettent d'élaguer l'espace de recherche durant la résolution de *MSCP*. Contrairement au solveur *BBMSCP*, le solveur *3LMSCP* exploite un ensemble de colorations restreintes, appelées colorations majeures. La notion de coloration majeure induit une abstraction de l'espace de recherche en supprimant l'ensemble des colorations symétriques d'une coloration.

Ce manuscrit est organisé en cinq chapitres. Le premier chapitre introduit *SAT* et *MSCP*, et rappelle quelques notions de complexité afin de situer *MSCP*. Le deuxième chapitre s'intéresse aux différentes méthodes existantes dans la littérature pour la résolution de *MSCP*. Ce chapitre introduit également l'ensemble de benchmarks *DIMACS* et *COLOR* considérés pour nos expérimentations. Le troisième chapitre présente la notion de motif et introduit les bornes  $UB_A$ ,  $UB_S$  et  $LBM_\Sigma$ . Le quatrième chapitre présente les différents encodages de *MSCP* en un problème *weighted partial MaxSAT* et en un problème *weighted partial MinSAT*, et étudie l'amélioration de ces encodages par  $UB_S$ . Le cinquième chapitre présente les solveurs *BBMSCP* et *3LMSCP*. Nous terminons ce manuscrit par un ensemble de conclusions et perspectives.

# Chapitre 1

## De la Complexité au Problème de la Somme Coloration Minimum d'un Graphe

### Sommaire

---

|  |           |
|--|-----------|
| <b>1.1 Classes de complexité</b> . . . . .                                     | <b>3</b>  |
| 1.1.1 Machine de Turing . . . . .  | 3         |
| 1.1.2 Définition de P et de NP par une machine de Turing . . . . .             | 4         |
| 1.1.3 NP-complet et NP-difficile . . . . .                                     | 6         |
| <b>1.2 De la logique propositionnelle au problème SAT</b> . . . . .            | <b>7</b>  |
| 1.2.1 Logique propositionnelle . . . . .                                       | 7         |
| 1.2.2 Le problème SAT . . . . .  | 9         |
| <b>1.3 Introduction aux graphes</b> . . . . .                                  | <b>10</b> |
| 1.3.1 Préliminaires . . . . .  | 10        |
| 1.3.2 Typologie . . . . .  | 13        |
| <b>1.4 Problèmes de coloration</b> . . . . .                                   | <b>16</b> |
| 1.4.1 Coloration de graphe . . . . .   | 16        |
| 1.4.2 Problème de coloration d'un graphe . . . . .                             | 18        |
| 1.4.3 Problème de partition chromatique de coût optimal . . . . .              | 18        |
| 1.4.4 Problème de multi-coloration et sum multi-coloration de graphe . . . . . | 19        |
| 1.4.5 Problème de la somme coloration minimum d'un graphe . . . . .            | 21        |
| <b>1.5 Conclusion</b> . . . . .  | <b>22</b> |

---

Ce chapitre a pour objectif de présenter le Problème de la Somme Coloration Minimum d'un graphe (*MSCP*) et un ensemble de notions utilisé tout au long de ce manuscrit. Nous commençons par introduire quelques notions de complexité. Ensuite, nous présentons rapidement le problème *SAT* et nous rappelons quelques notions associées aux graphes. Avant de conclure ce chapitre, nous présentons différents problèmes liés à la coloration de graphe.

### 1.1 Classes de complexité

#### 1.1.1 Machine de Turing

Hilbert et Ackermann proposent en 1928 dans [Hilbert et Ackermann \(1950\)](#), le problème de décidabilité ("*Entscheidungsproblem*"). Ce dernier consiste à déterminer l'existence d'une méthode algorithmique permettant de vérifier s'il existe une instanciation des variables d'une formule de la logique du premier ordre la rendant vraie. Pour répondre à ce problème, Alan Turing proposa dans [Turing \(1937\)](#) une machine abstraite, connue sous le nom de *machine de Turing*. Une machine de

Turing est constituée d'un *ruban* (Figure 1.1) représentant une mémoire infinie, sur lequel une tête de lecture/écriture se déplace en suivant un ensemble fini de règles.

|     |   |   |  |   |   |  |   |  |     |
|-----|---|---|--|---|---|--|---|--|-----|
| ... | a | c |  | b | l |  | d |  | ... |
|-----|---|---|--|---|---|--|---|--|-----|

FIGURE 1.1 – Représentation du contenu d'un ruban d'une machine de Turing

Une machine de Turing se décrit formellement comme suit :

**Définition 1 *Machine de Turing***

Une machine de Turing est un quadruplet  $M=(K,\Sigma,\delta,s_0)$  avec :

- $K$  un ensemble fini d'états ;
- $\Sigma$  un ensemble fini de symboles appelé alphabet ;
- $\delta : K \times \Sigma \rightarrow (K \cup \{\text{"alt"}, \text{"yes"}, \text{"no"}\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\}$  une fonction de transition, où *alt* est un état d'arrêt, *yes* est un état d'acceptation, *no* est un état de rejet,  $\leftarrow$  ( $\rightarrow$ ) est une instruction de déplacement de la tête de lecture/écriture vers la gauche (droite) et  $-$  est une instruction de non-déplacement de la tête de lecture/écriture ;
- $s_0 \in K$  l'état initial.

**Remarque 1** Nous considérons que le caractère blanc  $\sqcup \in \Sigma$ .

Bien qu'étant une approche rudimentaire, il est possible de réaliser, à l'aide d'une machine de Turing, un ensemble d'algorithmes permettant de résoudre de nombreux problèmes tels que la détermination de palindrome, le décalage d'un bit, etc. Les machines de Turing sont une composante essentielle de l'informatique théorique puisqu'elles sont la base des théories de complexité.

Avant d'introduire quelques classes de complexité, il est nécessaire de faire la distinction entre une *machine de Turing déterministe* et une *machine de Turing non-déterministe*.

**Définition 2 *Machine de Turing déterministe***

Une machine de Turing est dite *déterministe* si  $\forall (s, e) \in K \times \Sigma, \exists ! d \in \delta$ .

**Définition 3 *Machine de Turing non-déterministe***

Une machine de Turing est dite *non-déterministe* si  $\forall (s, e) \in K \times \Sigma, \exists d \in \delta$ .

Bien que formellement proche, la différenciation entre une machine de Turing déterministe et une machine de Turing non-déterministe est due à la non-unicité de la fonction de transition de  $\delta$ , pour tout couple  $K \times \Sigma$ , dans le cas d'une machine de Turing non-déterministe. Un choix est donc laissé à la machine dans l'optique qu'un état *yes* soit rencontré. Pour prouver l'existence d'une solution vérifiant une formule, il est simplement nécessaire de rencontrer un état *yes*. Cependant, pour prouver l'inexistence d'une telle solution, il est nécessaire que toutes les exécutions mènent à l'état *no*.

Nous allons à présent introduire quelques notions de complexité.

**1.1.2 Définition de P et de NP par une machine de Turing**

Les problèmes traités dans ce manuscrit sont des problèmes de *décision* ou d'*optimisation*.

**Définition 4 *Problème de décision***

Un problème de décision  $\Pi = (D_\Pi, Y_\Pi)$  est un problème constitué d'un ensemble d'instances  $D_\Pi$  pour lequel il existe un sous-ensemble  $Y_\Pi \subset D_\Pi$  d'instances positives.

**Définition 5 *Problème décidable (indécidable)***

Un problème de décision est dit *décidable (indécidable)* si et seulement si il est possible (impossible) de déterminer une machine de Turing retournant un état "yes" ou un état "no" en un temps fini pour ce problème.

**Définition 6 Problème d'optimisation**

Un problème d'optimisation est un quadruplet  $\Pi = (D_\Pi, S_\Pi, m_\Pi, \text{obj}_\Pi)$  tel que :

- $D_\Pi$  représente un ensemble d'instances ;
- étant donnée  $I \in D_\Pi$ ,  $S_\Pi(I)$  représente l'ensemble des solutions réalisables de  $I$  ;
- étant donnée  $I \in D_\Pi$  et  $x \in S_\Pi(I)$ ,  $m_\Pi(I, x)$  est une fonction d'évaluation assignant une valeur à la solution  $x$  pour l'instance  $I$  ;
- $\text{obj}_\Pi \in \{\min, \max\}$  indique la minimisation ou la maximisation du problème.

Un problème d'optimisation  $\Pi$  consiste à déterminer pour une instance  $I \in D_\Pi$  une solution  $x' \in S_\Pi(I)$  telle que :

- si  $\text{obj}_\Pi = \min$  :  $\forall x \in S_\Pi(I) m_\Pi(I, x') \leq m_\Pi(I, x)$  ;
- si  $\text{obj}_\Pi = \max$  :  $\forall x \in S_\Pi(I) m_\Pi(I, x') \geq m_\Pi(I, x)$ .

Avant de présenter les définitions des classes  $P$  et  $NP$ , nous allons dans un premier temps nous intéresser à la notion de *complexité temporelle*. La complexité temporelle d'une machine de Turing est représentée par une fonction  $t : \mathbb{N} \rightarrow \mathbb{N}$ , où  $t(n)$  représente le nombre d'étapes de calculs nécessaires à une machine de Turing pour donner un résultat. De cela, nous pouvons définir formellement la classe de complexité  $TIME$ .

**Définition 7  $O(t(n))$**

La complexité temporelle d'un algorithme  $f$  est dit en  $O(t(n))$  s'il existe une constante  $c \in \mathbb{R}^{*+}$ , tel que  $\forall n \in \mathbb{N}^*$ ,  $f(n) \leq c \times t(n)$ .

**Définition 8  $\Omega(t(n))$**

La complexité temporelle d'un algorithme  $f$  est dit en  $\Omega(t(n))$  s'il existe une constante  $c \in \mathbb{R}^{*+}$ , tel que  $\forall n \in \mathbb{N}^*$ ,  $f(n) \geq c \times t(n)$ .

**Définition 9  $TIME(t(n))$**

La classe de complexité temporelle  $TIME(t(n))$  représente la classe des problèmes décidables par une machine de Turing déterministe en temps  $t(n)$ .

**Définition 10 Temps polynomial**

Un problème est dit décidable en un temps polynomial si sa complexité est  $O(t(n))$  où  $t(n) = n^c$  et  $c \in \mathbb{N}$ .

Ainsi, la Définition 9 nous permet d'introduire la classe de complexité  $P$ .

**Définition 11  $P$**

La classe de complexité polynomiale ( $P$ ) représente l'ensemble des problèmes décidables en un temps polynomial par une machine de Turing déterministe. Soit :

$$P = \bigcup_{c \in \mathbb{N}} TIME(n^c)$$

L'ensemble des problèmes issu de la classe  $P$  admet un algorithme de détermination en temps polynomial. L'intérêt pratique de ces problèmes est que le temps d'attente pour l'obtention d'une solution reste sur une échelle de temps "raisonnable". Cependant, tout problème de décision et d'optimisation n'appartient pas nécessairement à la classe  $P$ . En effet, certains problèmes sont issus de la classe de complexité  $NP$ .

**Définition 12  $NTIME(t(n))$**

La classe de complexité temporelle  $NTIME(t(n))$  représente la classe des problèmes décidables par une machine de Turing non-déterministe en temps  $t(n)$ .

**Définition 13 NP**

La classe de complexité non-déterministe polynomiale (NP) représente l'ensemble des problèmes décidables en un temps polynomial par une machine de Turing non-déterministe. Soit :

$$NP = \bigcup_{c \in \mathbb{N}} \text{NTIME}(n^c)$$

Nous remarquons que tous les problèmes de la classe  $P$  appartiennent également à la classe  $NP$ . En effet, d'après les Définitions 2 et 3, une machine de Turing déterministe peut être considérée comme un cas particulier de machine de Turing non-déterministe. Par conséquent, si nous nous intéressons aux relations d'ordre existantes entre les classes de complexité, nous en déduisons que  $P \subseteq NP$ . Cependant,  $P \stackrel{?}{\supseteq} NP$ . Cette question reste à ce jour une des interrogations majeures de l'informatique.

Nous allons, dans la sous-section suivante, introduire les classes de complexité des problèmes  $NP$ -complet et  $NP$ -difficile.

**1.1.3 NP-complet et NP-difficile**

Parmi l'ensemble des problèmes de décision de la classe  $NP$ , il existe un ensemble de problèmes considéré comme plus dur. Si un algorithme vient à résoudre un seul de ces problèmes en un temps polynomial, alors tout problème de la classe  $NP$  admet un algorithme de résolution en temps polynomial. Ainsi, cela revient à prouver que  $P \supseteq NP$  ( $P=NP$ ).

Avant de pouvoir introduire la classe de complexité  $NP$ -complet, il est nécessaire de définir la notion de *réduction polynomiale*.

**Définition 14 Réduction polynomiale**

Soient  $\Pi_1$  et  $\Pi_2$  deux problèmes de décision. Une réduction polynomiale de  $\Pi_1$  vers  $\Pi_2$ , notée  $\Pi_1 \propto \Pi_2$ , est une fonction calculable en temps polynomial  $f$  telle que  $\pi$  est une solution de  $\Pi_1$  si et seulement si  $f(\pi)$  est une solution de  $\Pi_2$ .

Par conséquent tout algorithme résolvant le problème de décision  $\Pi_2$  résoud également le problème de décision  $\Pi_1$ .

**Définition 15 NP-complet**

Un problème de décision  $\Pi_2$  de  $NP$  est dit  $NP$ -complet si pour tous problèmes de décision  $\Pi_1$  de  $NP$ ,  $\Pi_1 \propto \Pi_2$ .

Le premier problème de décision à avoir été démontré  $NP$ -complet fut le problème de satisfiabilité ( $SAT$ ), que nous introduirons plus tard dans ce manuscrit (Section 1.2.2), dans Cook (1971).

**Théorème 1** Soit  $\Pi$  un problème de décision. Si  $\Pi \in NP$  alors  $\Pi \propto SAT$ .

À la suite de ces travaux, Karp proposa dans Karp (1972) un ensemble de problèmes  $NP$ -complet qu'il démontra en considérant le théorème suivant :

**Théorème 2** Soit  $\Pi$  un problème de décision. Si  $\Pi \in NP$  et  $SAT \propto \Pi$  alors  $\Pi \in NP$ -complet.

Pour terminer cette section sur la complexité, nous allons introduire la classe de complexité  $NP$ -difficile. Cette classe de complexité est importante pour ce manuscrit puisque c'est à cette classe qu'appartient le problème de la somme coloration minimum d'un graphe. Contrairement au problème  $NP$ -complet, un problème appartenant à la classe  $NP$ -difficile n'appartient pas nécessairement à la classe  $NP$ . Formellement, ce dernier est défini comme suit :

**Définition 16 NP-difficile**

Un problème  $\Pi_1$  est dit  $NP$ -difficile s'il existe un problème  $\Pi_2 \in NP$ -complet tel que  $\Pi_2 \propto \Pi_1$ .

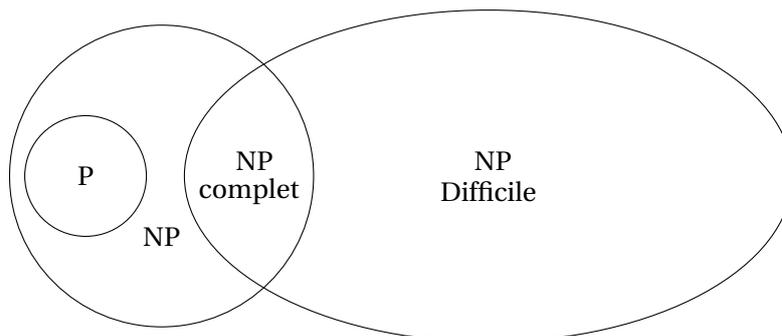


FIGURE 1.2 – Diagramme représentant une structuration possible des classes de complexité où  $P \neq NP$ .

Une des représentations possibles de la structuration des classes de complexité présentée dans ce manuscrit est donnée par la Figure 1.2.

Nous avons abordé dans cette section quelques notions de complexité. Pour plus de précision, le lecteur peut se reporter à l'ouvrage *computational complexity* de Papadimitriou (2003) et à l'ouvrage *computers and intractability* de Garey et Johnson (1979).

Nous allons dans la section suivante introduire le problème fondateur de la *NP*-complétude : le problème SAT.

## 1.2 De la logique propositionnelle au problème SAT

Le problème SAT est un problème de décision important en informatique théorique. Il a été le premier problème démontrant l'existence de la *NP*-complétude (Cook (1971)). Le formalisme utilisé pour la modélisation de ce problème est celui de la logique propositionnelle.

### 1.2.1 Logique propositionnelle

Au fil de notre histoire, la logique a évolué. L'essence même de la logique est de permettre à l'homme d'extraire un raisonnement, de tirer un ensemble de conclusions, à partir d'un ensemble de règles d'inférence. La logique syllogistique introduite par Aristote fut l'une des premières logiques formelles. Bien que rudimentaire, elle permet de dessiner les bases de la logique mathématique, en posant le fait que toute logique formelle possède une syntaxe et une sémantique pour pouvoir raisonner. La logique propositionnelle que nous allons présenter est l'une de ces logiques. Elle permet la représentation d'un ensemble de connaissances à l'aide de *propositions*.

#### Définition 17 Proposition

*Une proposition est une expression pouvant être vraie ou fausse.*

Pour pouvoir présenter la logique propositionnelle, il est nécessaire d'étudier la syntaxe et la sémantique liées à la logique propositionnelle.

#### Syntaxe

Afin de rester en adéquation avec le jargon théorique, nous utiliserons dans la suite de ce manuscrit le terme de formule pour exprimer une proposition. Avant de définir les règles nous permettant de construire une formule "bien formée" de la logique propositionnelle, nous allons définir l'abécédaire de cette dernière.

#### Définition 18 Alphabet de la logique propositionnelle

*L'alphabet de la logique propositionnelle est constituée :*

- de deux symboles  $\top$  et  $\perp$  ;
- d'un ensemble de variables propositionnelles  $\mathcal{P}_s = \{a, b, c, \dots\}$  ;

- d'un ensemble de connecteurs logiques :  $\{\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow\}$ ;
- des parenthèses ouvrante ( et fermante ).

Une *formule de la logique propositionnelle* est donc composée d'un ensemble de variables propositionnelles, de connecteurs logiques, etc. Cependant, comme pour un langage, il est nécessaire de suivre un ensemble de règles afin de construire une formule cohérente.

**Définition 19 Formule de la logique propositionnelle**

*L'ensemble des formules bien formées de la logique propositionnelle est le plus petit ensemble de formules tel que :*

- $p \in \{\mathcal{P}_s \cup \{\top, \perp\}\}$  est une formule de la logique propositionnelle ;
- si  $A$  est une formule de la logique propositionnelle alors  $(\neg A)$  est une formule de la logique propositionnelle ;
- si  $A$  et  $B$  sont deux formules de la logique propositionnelle alors  $(A \vee B)$ ,  $(A \wedge B)$ ,  $(A \Rightarrow B)$  et  $(A \Leftrightarrow B)$  sont des formules de la logique propositionnelle.

À l'aide des Définitions 18 et 19, il nous est maintenant possible de construire une formule. Nous allons à présent étudier la sémantique afin de pouvoir donner un sens, raisonner et analyser ces formules.

**Sémantique**

La sémantique de la logique propositionnelle se base sur l'algèbre de Boole. À chaque variable de  $\mathcal{P}_s$  est associée une valeur appartenant à l'ensemble Booléen  $\mathbb{B}$  prenant ses valeurs dans l'ensemble  $\{\text{vrai, faux}\}$ ,  $\{V, F\}$ , ou encore  $\{0, 1\}$ , appelée valeur de vérité. L'ensemble des valeurs de vérité associées à chaque variable d'une formule est défini par la fonction d'*interprétation*.

**Définition 20 Interprétation**

*Une interprétation est une fonction  $\mathcal{I} : \mathcal{P}_s \rightarrow \{0, 1\}$  qui associe à chaque variable propositionnelle une valeur de vérité.*

Afin de pouvoir interpréter une formule nous définissons ci-dessous le *niveau d'une formule*.

**Définition 21 Niveau d'une formule**

*Soit  $N_n$  avec  $n \in \mathbb{N}$ , le niveau d'une formule.  $N_n$  se définit comme suit :*

- $N_0 \in \mathcal{P}_s$  ;
- $N_1 \in \{\neg \mathcal{I}(p) \mid p \in N_0\} \cup \{\mathcal{I}(p) \wedge \mathcal{I}(q) \mid p \in N_0 \text{ et } q \in N_{i < 1}\} \cup \{\mathcal{I}(p) \vee \mathcal{I}(q) \mid p \in N_0 \text{ et } q \in N_{i < 1}\} \cup \{\mathcal{I}(p) \Rightarrow \mathcal{I}(q) \mid p \in N_0 \text{ et } q \in N_{i < 1}\} \cup \{\mathcal{I}(p) \Leftrightarrow \mathcal{I}(q) \mid p \in N_0 \text{ et } q \in N_{i < 1}\}$  ;
- ... ;
- $N_{n+1} \in \{\neg \mathcal{I}(p) \mid p \in N_n\} \cup \{\mathcal{I}(p) \wedge \mathcal{I}(q) \mid p \in N_n \text{ et } q \in N_{i < n+1}\} \cup \{\mathcal{I}(p) \vee \mathcal{I}(q) \mid p \in N_n \text{ et } q \in N_{i < n+1}\} \cup \{\mathcal{I}(p) \Rightarrow \mathcal{I}(q) \mid p \in N_n \text{ et } q \in N_{i < n+1}\} \cup \{\mathcal{I}(p) \Leftrightarrow \mathcal{I}(q) \mid p \in N_n \text{ et } q \in N_{i < n+1}\}$  ;

Ainsi, l'interprétation d'une formule se réduit à l'interprétation de chaque sous-niveau de formules en commençant par  $N_0$  en se basant sur les tables de vérité (Figure 1.3).

L'*interprétation d'une formule* se définit formellement comme suit :

**Définition 22 Interprétation d'une formule**

*Soit  $p, q \in \mathcal{P}_f$ . L'interprétation d'une formule  $F$  suit les règles suivantes :*

- $\mathcal{I}(\top) = 1$  ;
- $\mathcal{I}(\perp) = 0$  ;

|          |   |             |   |   |             |   |   |               |   |   |                   |   |   |
|----------|---|-------------|---|---|-------------|---|---|---------------|---|---|-------------------|---|---|
| $\neg$   |   | $\vee$      | 0 | 1 | $\wedge$    | 0 | 1 | $\Rightarrow$ | 0 | 1 | $\Leftrightarrow$ | 0 | 1 |
| 0        | 1 | 0           | 0 | 1 | 0           | 0 | 0 | 0             | 1 | 0 | 0                 | 1 | 0 |
| 1        | 0 | 1           | 1 | 1 | 1           | 0 | 1 | 1             | 1 | 1 | 1                 | 0 | 1 |
| Négation |   | Conjonction |   |   | Disjonction |   |   | Implication   |   |   | Équivalence       |   |   |

FIGURE 1.3 – Tables de vérité de la logique propositionnelle

- $\mathcal{I}(p \vee q) = \mathcal{I}(p) \vee \mathcal{I}(q)$ ;
- $\mathcal{I}(p \wedge q) = \mathcal{I}(p) \wedge \mathcal{I}(q)$ ;
- $\mathcal{I}(p \Rightarrow q) = \mathcal{I}(\neg p) \vee \mathcal{I}(q)$ ;
- $\mathcal{I}(p \Leftrightarrow q) = \mathcal{I}(p \Rightarrow q) \wedge \mathcal{I}(q \Rightarrow p)$ .

**Exemple 1** Soit  $F$  une formule avec  $a, b, c, d \in \mathcal{P}_s$  et  $\mathcal{I}(a)=0$ ,  $\mathcal{I}(b)=1$ ,  $\mathcal{I}(c)=0$  et  $\mathcal{I}(d)=1$ .

- $N_0 : F = ((a \vee b) \Rightarrow (b \vee bc)) \wedge d$ ;
- $N_1 : F = ((0 \vee 1) \Rightarrow (1 \vee 0)) \wedge 1$ ;
- $N_2 : F = (1 \Rightarrow 1) \wedge 1$ ;
- $N_3 : F = 1 \wedge 1$ ;
- $N_4 : F = 1$ .

Une formule  $F$  est satisfiable s'il existe une interprétation  $\mathcal{I}$  pour laquelle  $\mathcal{I}(F) = 1$ . Une telle interprétation est appelée *modèle* de  $F$ . Si pour toute interprétation  $\mathcal{I}$  la formule  $F$  n'admet aucun modèle alors cette dernière est dite *insatisfiable*.

## 1.2.2 Le problème SAT

Pour toute formule de la logique propositionnelle, il existe une multitude de formules équivalentes.

**Exemple 2** Soit  $a, b, c \in \mathcal{P}_s$ ,  $(a \vee b) \wedge (c) = (a \vee b) \wedge (c \wedge \top) = (a \wedge c) \vee (b \wedge c) = \dots$

Cependant, il est possible d'extraire pour certaines formules un pattern récurrent. Dans ce manuscrit, nous nous intéressons à l'un de ces patterns : *la forme normale conjonctive (CNF)*. Avant de définir la notion de *CNF*, il est nécessaire d'introduire les termes *littéral* et *clause*.

### Définition 23 Littéral

Un littéral est une variable propositionnelle ou sa négation.

**Exemple 3** Soit  $p \in \mathcal{P}_s$ ,  $p$  et  $\neg p$  sont des littéraux.

### Définition 24 Clause

Une clause est une disjonction de littéraux de la forme  $\bigvee_{i \in \mathbb{N}} l_i$  où chaque  $l_i$  est un littéral.

**Exemple 4** Soit  $c, l, r \in \mathcal{P}_s$ .  $r \vee \neg c \vee l$  est une clause.

### Définition 25 CNF

Une formule  $F$  est une forme normale conjonctive (CNF) si  $F = \bigwedge_{i \in \mathbb{N}} c_i$  telle que chaque  $c_i$  est une clause.

**Exemple 5** Soit  $a, b, c, d \in \mathcal{P}_s$ .  $(a \vee c \vee \neg d) \wedge (b \vee \neg c \vee d) \wedge (c \vee d)$  est une CNF

Ainsi, le problème SAT se définit formellement comme suit :

**Définition 26 Problème SAT**

- Instance : Soit une formule  $F$  telle que  $F$  est une CNF;
- Question : Existe-t-il une affectation des variables de  $F$  satisfaisant chacune des clauses ?

**Exemple 6** Soit  $F=(a \vee c \vee \neg d) \wedge (b \vee \neg c \vee d) \wedge (c \vee d)$ . Une solution au problème est :  $\mathcal{S}=\{a, b, c\}$ .

Nous allons à présent nous intéresser à la notion de graphe et à différents problèmes de coloration de graphe.

### 1.3 Introduction aux graphes

#### 1.3.1 Préliminaires

Un *graphe* est un objet mathématique (Figure 1.4) comportant un ensemble de points appelés *sommets* et un ensemble de liens entre certains de ces sommets appelés *arêtes*. Formellement, nous définissons un graphe de la façon suivante :

**Définition 27 Graphe**

Un graphe  $G$  est un couple  $(V, E)$ , noté  $G = (V, E)$ , tel que :

- $V$  représente l'ensemble des sommets de  $G$  ;
- $E \subseteq \{(u, v) | u, v \in V^2\}$ .

**Définition 28 Arête**

Soit  $(G = V, E)$  un graphe. Une arête est un couple  $(u, v) \in E$ .

**Remarque 2** Les graphes présentés dans ce manuscrit sont considérés comme étant non-orientés, c'est à dire si un couple de sommets  $(u, v) \in E$  alors nous considérons implicitement que le couple  $(v, u) \in E$ . De plus, ces derniers sont connexes. En d'autres termes,  $\forall u, v \in V, \exists \{u_1, u_2, \dots, u_n\} \subset V$  tel que  $\{(u, u_1), (u_1, u_2), \dots, (u_n, v)\} \subseteq E$ .

**Exemple 7** Soit un graphe  $G = (V, E)$  avec  $V = \{a, b, c, d, e, f\}$  et  $E = \{(a, b), (a, c), (a, d), (b, d), (b, f), (c, e), (d, e), (e, f)\}$ . La Figure 1.4 est une représentation de  $G$ .

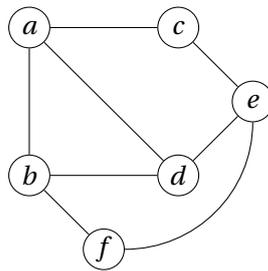


FIGURE 1.4 – Représentation d'un graphe  $G = (V, E)$ .

Un sommet est relié à un ensemble de sommets d'un graphe par l'intermédiaire d'arêtes. Ainsi, nous définissons la notion de *voisinage d'un sommet* comme suit :

**Définition 29 Voisinage d'un sommet**

Le voisinage d'un sommet  $v$  de  $G$ , noté  $\mathcal{N}(v)$ , est l'ensemble des sommets de  $G$  tel que :

$$\mathcal{N}(v) = \{u | (u, v) \in E\}$$

**Remarque 3** Deux sommets  $u$  et  $v$  sont dits adjacents ou voisins si  $u \in \mathcal{N}(v)$ .

**Définition 30 Degré d'un sommet**

Le degré d'un sommet  $v$  de  $G$ , noté  $d(v)$ , est la cardinalité du voisinage de  $v$  :

$$d(v) = |\mathcal{N}(v)|$$

**Définition 31 Degré d'un graphe**

Le degré d'un graphe  $G$ , noté  $\Delta(G)$ , est la valeur maximale des degrés des sommets du graphe  $G$  :

$$\Delta(G) = \max\{d(v) \mid v \in V\}$$

**Exemple 8** Soit le graphe  $G$  de la figure 1.4,  $\mathcal{N}(C) = \{A, E\}$ ,  $d(C) = 2$  et  $\Delta(G) = d(D) = 3$ .

**Définition 32 Chaîne**

Soit un graphe connexe  $G = (V, E)$  tel que  $V = \{u_1, u_2, \dots, u_n\}$ . Le graphe  $G$  est une chaîne, si et seulement si  $d(u_1) = d(u_n) = 1$  et  $\forall v \in V \setminus \{u_1, u_n\}, d(v) = 2$ .

Un graphe est dit *complet* si tous les sommets de ce dernier sont reliés entre eux. Un graphe est dit *partiel* si une ou plusieurs de ses arêtes ont été supprimées. Pour terminer, un *sous-graphe* est un graphe induit par un sous-ensemble de sommets. Les notions de graphe complet, de sous-graphe et de graphe partiel se définissent donc formellement de la façon suivante :

**Définition 33 Graphe complet**

Un graphe  $G$  est complet si  $\forall (u, v) \in V^2, (u, v) \in E$ .

**Définition 34 Graphe partiel**

Un graphe  $G' = (V, E')$  est un graphe partiel de  $G = (V, E)$  si et seulement si  $E' \subset E$ .

**Définition 35 Sous-graphe**

Un graphe  $G' = (V', E')$  est un sous-graphe de  $G = (V, E)$  induit par le sous-ensemble  $V'$  si et seulement si  $V' \subset V$  et  $E' = \{(u, v) \in E \mid u \in V' \text{ et } v \in V'\}$ .

**Exemple 9** Soit le graphe  $G$  de la Figure 1.4. Les graphes des Figures 1.5, 1.6 et 1.7 représentent respectivement un graphe complet, un graphe partiel et un sous-graphe de  $G$ .

**Remarque 4** Le graphe  $G$  de la Figure 1.4 est un graphe partiel du graphe complet de la Figure 1.5.

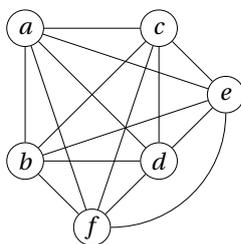


FIGURE 1.5 – Graphe complet.

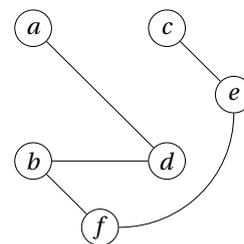


FIGURE 1.6 – Graphe partiel du graphe Figure 1.4.

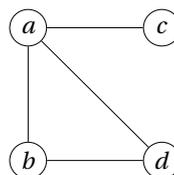


FIGURE 1.7 – Sous-graphe du graphe Figure 1.4 induit par  $V' = \{a, b, c, d\}$ .

Nous allons à présent introduire la notion de *graphe complémentaire*.

**Définition 36 Graphe complémentaire**

Le graphe  $\bar{G} = (V, \bar{E})$  est le graphe complémentaire de  $G = (V, E)$ , si et seulement si  $E \cap \bar{E} = \emptyset$  et  $(V, E \cup \bar{E})$  est un graphe complet.

**Exemple 10** Soit le graphe  $G$  de la Figure 1.4. Le graphe complémentaire de  $G$  est représenté par la Figure 1.8.

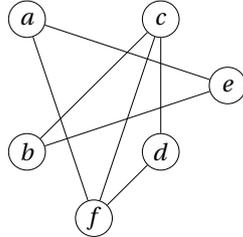


FIGURE 1.8 – Graphe complémentaire du graphe Figure 1.4.

Il est possible à partir d'un graphe  $G$  d'extraire des ensembles de sommets induisant des sous-graphes possédant une structure particulière. Ainsi, nous présentons la notion de *clique* et de *stable*. Dans le cas d'une clique, nous recherchons un sous-ensemble de sommets dans  $V$  formant un graphe complet. Pour un stable, nous recherchons un sous-ensemble de sommets dans  $V$  tel que tous ces sommets sont non adjacents deux à deux. Formellement, ces notions se définissent de la façon suivante.

**Définition 37 Clique**

Soit  $G = (V, E)$  un graphe. Une clique  $C$  de  $G$  est un sous-ensemble de sommets de  $V$  tel que  $\forall u, v \in C, (u, v) \in E$ .

**Définition 38 Stable**

Soit  $G = (V, E)$  un graphe. Un stable  $S$  est un sous-ensemble de sommets de  $V$  tel que  $\forall u, v \in S, (u, v) \notin E$ .

**Exemple 11** Soit le graphe  $G$  de la figure 1.4,  $C = \{a, b, d\}$  est une clique de  $G$  et  $S = \{b, c\}$  est un stable de  $G$ .

**Remarque 5** Soit  $G$  un graphe. Un stable (une clique) de  $G$  est une clique (un stable) de  $\bar{G}$ .

L'intérêt des notions de clique et de stable est qu'elles permettent la mise en évidence de certaines données dans un graphe. En effet, dans le cas des graphes sociaux, une clique ou un stable permet de mettre en évidence des groupes possédant des propriétés spécifiques communes pour la clique ou distinctes pour le stable. Il peut donc être pertinent de déterminer une clique ou un stable maximal ou maximum afin d'identifier des groupes de communautés devant faire l'objet de traitements spéciaux. Cependant, avant d'introduire les problématiques liées à la recherche de *clique maximum* ou de *stable maximum* d'un graphe, nous faisons la distinction formelle entre les notions de clique maximale et de clique maximum.

**Définition 39 Clique maximale**

Soit  $G = (V, E)$  un graphe. Une clique  $C$  de  $G$  est dite maximale si  $\nexists C' \subseteq V$ , avec  $C'$  une clique de  $G$ , telle que  $C \subset C'$ .

**Définition 40 Clique maximum**

Soit  $G = (V, E)$  un graphe. Une clique  $C$  de  $G$  est dite maximum si  $\forall C' \subseteq V$ , telle que  $C'$  est une clique de  $G$ , alors  $|C'| \leq |C|$ .

**Remarque 6** Les définitions de stable maximal et stable maximum sont similaires à celles de clique maximale et clique maximum.

Déterminer une clique maximum ou un stable maximum d'un graphe sont des problèmes d'optimisation combinatoire NP-difficiles qui se rapportent aux problèmes de décision prouvés NP-complet dans **Karp (1972)** suivant :

**Définition 41 Problème  $k$ -clique**

- Instance : Soit  $G = (V, E)$  un graphe et  $k$  un entier positif tel que  $k \leq |V|$  ;
- Question : Existe-t-il dans  $G$  une clique de cardinalité égale ou supérieure à  $k$  ?

**Définition 42 Problème  $k$ -stable**

- Instance : Soit  $G = (V, E)$  un graphe et  $k$  un entier positif tel que  $k \leq |V|$  ;
- Question : Existe-t-il dans  $G$  un stable de cardinalité égale ou supérieure à  $k$  ?

La cardinalité de la clique maximum d'un graphe  $G$  est notée  $\omega(G)$ . tandis que la cardinalité du stable maximum est notée  $\alpha(G)$ . Nous remarquons que  $G$  peut avoir plusieurs cliques maximums et plusieurs stables maximums.

Nous allons à présent nous intéresser à différentes familles de graphes possédant chacune des spécificités structurelles.

**1.3.2 Typologie**

Les propriétés structurelles de certaines familles de graphes peuvent permettre de diminuer le temps de recherche, voire de diminuer la complexité temporelle de quelques problèmes sur ces familles. Nous allons présenter rapidement quelques familles de graphes pour lesquelles des résultats relatifs à *MSCP* ont été démontrés dans la littérature.

**Définition 43 Arbre (Figure 1.9)**

Un arbre est un graphe  $G = (V, E)$  connexe tel que  $\forall u_1, u_2, u_3, \dots, u_k \in V, \exists \{(u_1, u_2), (u_2, u_3), \dots, (u_{k-1}, u_k), (u_k, u_1)\} \subseteq E$ .

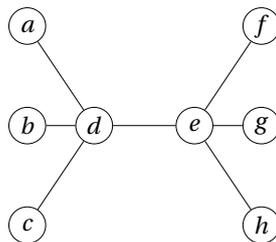


FIGURE 1.9 – Un arbre.

**Définition 44 Graphe biparti (Figure 1.10)**

Soit un graphe  $G = (V, E)$  tel que  $V = A \cup B$  et  $A \cap B = \emptyset$ . Le graphe  $G$  est dit biparti si et seulement si  $\forall (u, v) \in E, u \in A$  et  $v \in B$ .

**Définition 45  $k$ -arbre**

Un graphe  $G = (V, E)$  est un  $k$ -arbre si  $G$  est un graphe complet tel que  $|V| = k$ , ou si  $G$  a un sommet  $v$  de degré  $k$  tel que le sous-graphe induit par  $\mathcal{N}(v)$  est complet et le sous-graphe induit par  $V \setminus \{v\}$  est un  $k$ -arbre

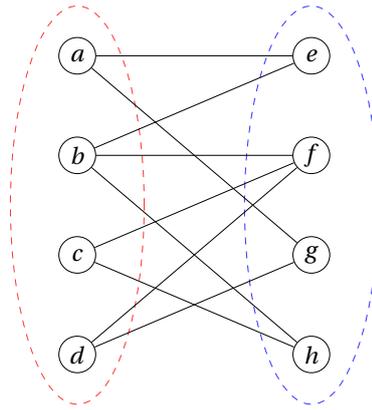


FIGURE 1.10 – Graphe biparti.

**Définition 46 *k*-arbre partiel**

Un graphe  $G'$  est un *k*-arbre partiel si et seulement si  $G'$  est un sous-graphe de  $G$  tel que  $G$  est un *k*-arbre.

**Exemple 12** Les Définitions 45 et 46 sont illustrées respectivement par la Figure 1.11 et la Figure 1.12.

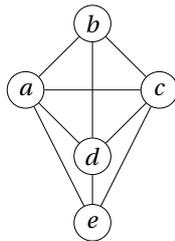


FIGURE 1.11 – 3-arbre.

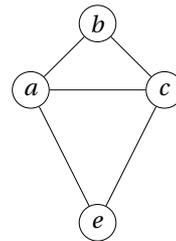


FIGURE 1.12 – 3-arbre partiel du graphe Figure 1.11.

**Définition 47 Graphe d'intervalles** (Figure 1.13)

Soit  $I = \{I_1, I_2, \dots, I_n\}$  un ensemble d'intervalles sur  $\mathbb{R}$ . Un graphe d'intervalles  $G = (V, E)$  est un graphe auquel est associé à chaque sommet  $v \in V$  un intervalle  $I_v \in I$  tel que  $\forall (u, v) \in E, I_u \cap I_v \neq \emptyset$ .

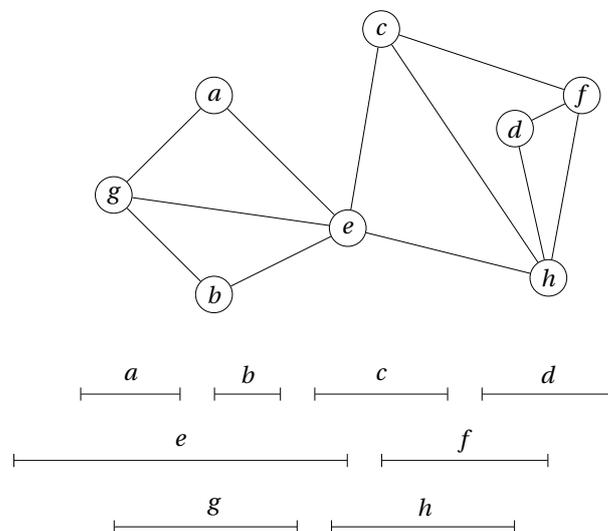


FIGURE 1.13 – Graphe d'intervalles.

**Définition 48 Graphe planaire** (Figure 1.14)

Un graphe  $G$  est dit planaire s'il existe une représentation topologique telle qu'aucune des arêtes de  $G$  ne se croisent.

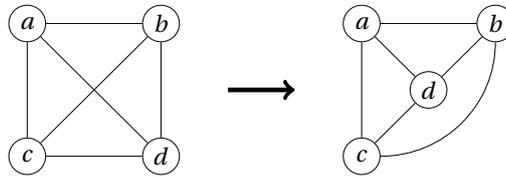


FIGURE 1.14 – Graphe planaire.

**Définition 49  $P_k$**  (Figure 1.15)

Un graphe  $G=(V,E)$  est  $P_k$  si  $G$  est une chaîne telle que  $|V|=k$ .

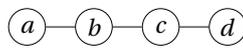


FIGURE 1.15 – Graphe  $P_4$ .

**Définition 50  $P_k$ -sparse** (1.16)

Un graphe  $G=(V,E)$  est  $P_k$ -sparse si  $\forall V' \subseteq V$ , tel que  $|V'|=k+1$ , le sous-graphe induit par  $V'$  contient au plus  $P_k$ .

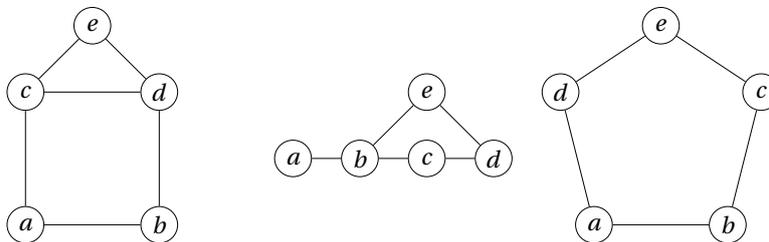


FIGURE 1.16 – Graphes  $P_4$ -sparse.

**Définition 51 Split-graphe** (Figure 1.17)

Un graphe  $G=(V,E)$  est un split-graphe si  $\exists I, C \subseteq V$  avec  $I \cup C = V$  et  $I \cap C = \emptyset$ , tel que  $C$  est une clique et  $I$  un stable.

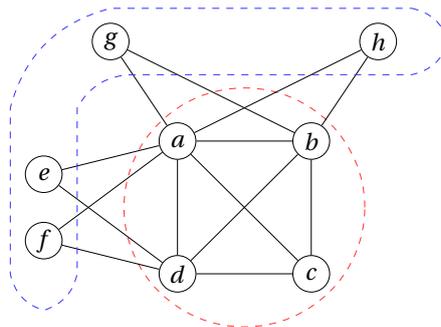


FIGURE 1.17 – Split-graphe.

**Définition 52**  *$k_C$ -split-graphe,  $k_I$ -split-graphe,  $k$ -split-graphe*

Soit  $G = (V, E)$  un graphe,  $I$  et  $C$  deux ensembles de sommets tels que  $I \cup C = V$  et  $I \cap C = \emptyset$ .

- Le graphe  $G$  est un  $k_C$ -split-graphe si  $\exists u \in C$  tel que  $d(u) - (|C| - 1) = k$  et  $\forall v \in C, d(v) - (|C| - 1) \leq k$ ;
- Le graphe  $G$  est un  $k_I$ -split-graphe si  $\exists u \in I$  tel que  $d(u) = k$  et  $\forall v \in I, d(v) \leq k$ ;
- Le graphe  $G$  est un  $k$ -split-graphe s'il est  $k_C$ -split-graphe ou  $k_I$ -split-graphe.

## 1.4 Problèmes de coloration

Les problèmes de coloration ont un vaste champ applicatif. En effet, ils permettent la gestion de flux tel que celui du trafic aérien ou ferroviaire, l'allocation de ressources telles que les fréquences réseaux, l'ordonnancement de processus en conflit de ressource et bien d'autre encore ( Bar-Noy *et al.* (1998); Barnier et Brisset (2004); Chaitin (1982); Chow et Hennessy (1990); Hale (1980); Kroon *et al.* (1996); Laurent et Hao (2009); Leighton (1979); Supowit (1987); Woo *et al.* (1991)). Dans cette section, après avoir introduit la notion de coloration de graphe, nous allons nous intéresser à différents problèmes de coloration. Pour certain de ces problèmes, un coût est associé aux couleurs afin de déterminer une coloration minimisant la somme des coûts, pour d'autres, l'objectif est de minimiser le nombre de couleurs, etc.

### 1.4.1 Coloration de graphe

La coloration de graphe est un problème en théorie des graphes dont les premiers travaux remontent à la fin du 19<sup>ème</sup> avec le théorème des quatre couleurs. La conjecture de ce théorème était de démontrer que seules quatre couleurs suffisent à colorier une carte tout en respectant le fait que deux états limitrophes ne puissent pas avoir une même couleur. Résoudre ce problème revient à colorer le graphe, où un état est représenté par un sommet, une frontière entre deux états est représentée par une arête et une couleur est représentée par un entier. La contrainte est que deux sommets voisins ne peuvent avoir une même couleur. Formellement, la notion de *coloration* se définit comme suit :

**Définition 53** *Coloration*

Une coloration d'un graphe  $G = (V, E)$  est une fonction  $c : V \rightarrow \{1, 2, \dots, k\}$  qui associe à chaque sommet  $v \in V$  une couleur  $c(v)$ .

**Définition 54** *Coloration valide*

Une coloration d'un graphe  $G = (V, E)$  est dite valide si  $\forall (u, v) \in E, c(u) \neq c(v)$ .

**Exemple 13** Soit le graphe  $G$  de la Figure 1.4. Une coloration valide de ce graphe est représentée par la Figure 1.18.

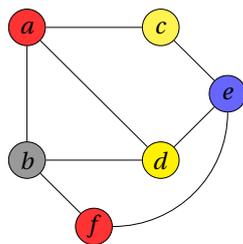


FIGURE 1.18 – Une coloration valide du graphe de la Figure 1.4.

Une coloration valide d'un graphe peut être considérée comme une partition des sommets de  $G$  en stables où chaque stable représente un ensemble de sommets colorés avec une même couleur. Chaque stable forme ce que nous appelons une *classe couleur*.

**Définition 55 Classe couleur**

Soit  $G = (V, E)$  un graphe et  $c : V \rightarrow \{1, 2, \dots, k\}$  une coloration valide de  $G$ . Une classe couleur  $X_n$  est un sous-ensemble de  $V$  tel que  $\forall v \in X_n, c(v) = n$ . Une coloration  $X$  est donc un ensemble de classes couleurs dont la cardinalité  $|X|$  est le nombre de couleurs utilisées.

**Exemple 14** Soit  $G = (V, E)$  le graphe de la Figure 1.4 et la coloration valide associée à ce graphe (Figure 1.18). Nous associons à chaque couleur un entier : rouge  $\rightarrow 1$ , jaune  $\rightarrow 2$ , bleu  $\rightarrow 3$  et gris  $\rightarrow 4$ . La coloration valide s'écrit donc  $X = \{\{a, f\}_1, \{c, d\}_2, \{e\}_3, \{b\}_4\}$ .

Durant la recherche d'une coloration valide, une coloration  $X$  n'est pas nécessairement définie sur l'ensemble des sommets d'un graphe. Une telle coloration est dite *partielle* et se définit comme suit :

**Définition 56 Coloration partielle**

Soit  $G = (V, E)$  un graphe. Une coloration  $X = \{X_1, X_2, \dots, X_k\}$  de  $G$  est partielle si  $|\bigcup_{i=1}^k X_i| \leq |V|$ .

Une coloration partielle induit qu'il existe un ensemble de sommets non colorés dans le graphe. Ainsi, nous introduisons les notions  $\mathcal{N}^+(v)$ ,  $\mathcal{N}^-(v)$  et  $Disp(v)$  qui correspondent respectivement à l'ensemble des sommets voisins colorés de  $v$ , l'ensemble des sommets voisins non colorés de  $v$  et l'ensemble des couleurs disponibles pour le sommet  $v$ .

**Définition 57  $\mathcal{N}^+(v)$**

Soit  $G = (V, E)$  un graphe,  $v$  un sommet de  $G$  et  $X$  une coloration partielle. L'ensemble des sommets voisins colorés de  $v$ , noté  $\mathcal{N}^+(v)$ , se définit comme suit :

$$\mathcal{N}^+(v) = \{u \in \mathcal{N}(v) \mid \exists! X_k \in X, u \in X_k\}$$

**Définition 58  $\mathcal{N}^-(v)$**

Soit  $G = (V, E)$  un graphe,  $v$  un sommet de  $G$  et  $X$  une coloration partielle. L'ensemble des sommets voisins non colorés de  $v$ , noté  $\mathcal{N}^-(v)$ , se définit comme suit :

$$\mathcal{N}^-(v) = \{u \in \mathcal{N}(v) \mid \forall X_k \in X, u \notin X_k\}$$

**Remarque 7**  $\mathcal{N}(v) = \mathcal{N}^+(v) \cup \mathcal{N}^-(v)$ .

**Définition 59  $Disp(v)$**

Soit  $G = (V, E)$  un graphe,  $v$  un sommet non coloré de  $G$  et  $X$  une coloration partielle. L'ensemble des couleurs disponibles pour un sommet  $v$ , noté  $Disp(v)$ , se définit comme suit :

$$Disp(v) = \{k \in \{1, \dots, |X|\} \mid \forall u \in \mathcal{N}^+(v), c(u) \neq k\}$$

Il est toujours possible de déterminer une coloration valide. En effet, il suffit de déterminer un nombre de couleurs  $k$  équivalent au nombre de sommets du graphe. Cependant, si nous fixons un  $k$  inférieur au nombre de sommets du graphe, est-il possible de déterminer une coloration valide? Ce problème est un problème de décision appelé problème de  $k$ -coloration. Le problème de  $k$ -coloration a été démontré *NP-complet* dans **Karp (1972)**. Il se définit formellement de la façon suivante :

**Définition 60 Problème de  $k$ -coloration**

— Instance : Soit un graphe  $G = (V, E)$  et un entier  $k$  ;

— Question : Existe-t-il une coloration  $X$  valide telle que  $|X| \leq k$  ?

Dans le cas où  $k$  est inférieur ou égale à 2, le problème de  $k$ -coloration est un problème de décision appartenant à la classe de complexité P. Cependant pour tout  $k$  supérieur ou égale à 3, le problème de  $k$ -coloration est un problème NP-complet [Karp \(1972\)](#). La preuve de la NP-complétude repose sur le principe que le problème 3-SAT se réduit polynomialement à un problème de 3-coloration et que le problème de 3-coloration se réduit polynomialement à tout problème de  $k$ -coloration tel que  $k$  est supérieur à 3 couleurs.

### 1.4.2 Problème de coloration d'un graphe

Le problème de coloration d'un graphe (*GCP*), également appelé le problème du nombre chromatique d'un graphe, est le problème d'optimisation associé au problème de  $k$ -coloration. L'objectif de *GCP* est de déterminer le nombre de couleurs minimum nécessaire à l'obtention d'une coloration valide d'un graphe. Ce nombre de couleurs minimum est appelé *nombre chromatique*.

#### Définition 61 *GCP*

Soit  $G$  un graphe. Le problème de coloration d'un graphe (*GCP*) consiste à déterminer le nombre chromatique d'un graphe  $G$ , noté  $\chi(G)$ , tel que :

$$\chi(G) = \min\{|X| \mid X \text{ est une coloration valide de } G\}$$

**Exemple 15** Soit le graphe  $G$  de la Figure 1.4. Le nombre de couleur nécessaire à l'obtention d'une solution optimale pour le problème *GCP* d'une coloration est de 3 ( $\chi(G) = 3$ ). Une coloration solution de *GCP* est donnée par la Figure 1.19. avec

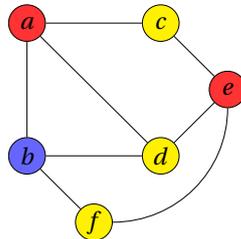


FIGURE 1.19 – Une solution optimale pour *GCP*.

*GCP* étant une extension du problème de  $k$ -coloration, il est donc NP-difficile [Karp \(1972\)](#). Il est intéressant de remarquer que les graphes planaires sont 4-colorables [Appel et Haken \(1989\)](#); [Robertson et al. \(1996\)](#). Cependant, la détermination du nombre chromatique de ces graphes reste un problème NP-difficile.

Nous remarquons que la notion de nombre chromatique a permis d'introduire la famille de *graphe parfait*.

#### Définition 62 *Graphe parfait*

Un graphe  $G$  est dit parfait si pour tout sous-graphe induit  $G'$  de  $G$ ,  $\omega(G') = \chi(G')$ .

**Exemple 16** La Définition 62 est illustrée par la Figure 1.20.

### 1.4.3 Problème de partition chromatique de coût optimal

Le problème de partition chromatique de coût optimal (*OCCP*) est un problème de coloration introduit dans [Supowit \(1987\)](#). Différemment à un problème de coloration classique, le problème associe un poids à chaque couleur. L'objectif du problème *OCCP* est donc de déterminer une coloration valide minimisant la somme des poids des couleurs utilisées.

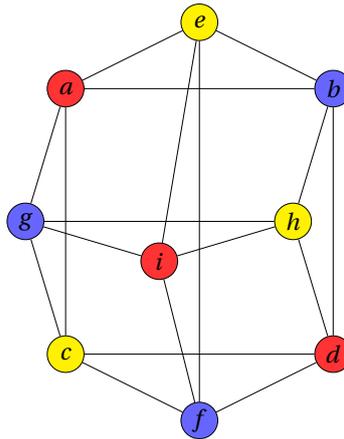


FIGURE 1.20 – Graphe parfait.

**Définition 63 OCCP**

Soit  $G$  un graphe et à chaque couleur  $c_1, c_2, \dots, c_k$  est associée un poids  $w_1, w_2, \dots, w_k$  tel que le poids de la couleur  $c_i$  est  $w_i, \forall i \in \{1, \dots, k\}$ . L'objectif du problème de partition chromatique de coût optimal d'un graphe (OCCP) est de déterminer une coloration telle que :

$$\Sigma_{\text{OCCP}}(G) = \min \left\{ \sum_{i=1}^{|X|} w_i \times |X_i| \mid X \text{ est une coloration valide de } G \right\}$$

**Exemple 17** Soit le graphe  $G$  de la Figure 1.4. Nous associons le poids 1 à la couleur rouge, 4 à la couleur jaune et 5 à la couleur bleue. Une solution optimale pour le problème OCCP est représenté par la Figure 1.21 ( $\Sigma_{\text{OCCP}}(G) = 16$ ).

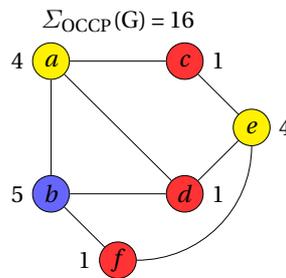


FIGURE 1.21 – Une solution optimale pour le problème OCCP.

Dans le cas général, pour un graphe arbitraire, le problème OCCP est un problème NP-difficile. En effet, comme nous le montrerons par la suite, cela résulte du fait que le problème OCCP est une généralisation du problème de la somme coloration minimum d'un graphe, lui-même étant NP-difficile. Différents travaux ont également étudié la complexité du problème OCCP sur certaines familles de graphes telles que les graphes d'intervalles, bipartis, ... [Jansen \(1997\)](#); [Kroon et al. \(1996\)](#).

**1.4.4 Problème de multi-coloration et sum multi-coloration de graphe**

Un problème de multi-coloration associe à chaque sommet d'un graphe, un ensemble de couleurs de cardinalité fixe. Cependant, nous remarquons que deux ensembles de couleurs associés à deux sommets voisins sont disjoints. Avant de définir formellement la multi-coloration, il est nécessaire de s'intéresser à une *instance de la multi-coloration*.

**Définition 64 Instance de la multi-coloration**

Une instance de la multi-coloration est un couple  $(G, l)$  où  $G = (V, E)$  est un graphe et  $l$  une fonction  $l : V \rightarrow \mathbb{N}$  qui associe à chaque sommet  $v \in V$  une longueur  $l(v)$ .

**Définition 65 Longueur maximale**

La longueur maximale d'une instance de la multi-coloration  $(G, l)$ , notée  $l_{\max}$ , se définit comme étant :

$$l_{\max} = \max\{l(v) \mid \forall v \in V\}$$

**Définition 66 Multi-coloration**

Une multi-coloration d'un graphe  $G = (V, E, l)$  est une fonction  $F : V \rightarrow 2^{\mathbb{N}}$  qui assigne à chaque sommet  $v \in V$  un ensemble  $F(v)$  de couleurs tel que  $|F(v)| = l(v)$  et  $\forall (u, v) \in E, F(u) \cap F(v) = \emptyset$ .

Nous allons à présent définir la plus petite couleur d'un sommet  $v$  du graphe, notée  $F_{\min}(v)$ , et la plus grande couleur, notée  $F_{\max}(v)$ .

**Définition 67  $F_{\min}(v), F_{\max}(v)$**

Soit  $(G, l)$  une instance de la multi-coloration et  $v \in V$ .  $F_{\min}(v)$  et  $F_{\max}(v)$  se définissent comme suit :

- $F_{\min}(v) = \min\{i \mid i \in F(v)\}$  ;
- $F_{\max}(v) = \max\{i \mid i \in F(v)\}$

Le problème de somme multi-coloration d'un graphe est un problème d'optimisation NP-difficile introduit dans [Bar-Noy et al. \(2000\)](#). Cependant, il est nécessaire de faire la distinction entre le problème utilisant uniquement la fonction de multi-coloration (Définition 66), appelé *somme multi-coloration préemptive (pSMC)*, et le problème de *somme multi-coloration non-préemptive (npSMC)* qui intègre en plus de la contrainte de voisinage, une contrainte de contiguïté entre les couleurs d'un sommet.

**Définition 68 pSMC**

Le problème de somme multi-coloration préemptive, notée pSMC, a pour objectif de déterminer la somme multi-coloration chromatique, notée  $\Sigma_{pSMC(G)}$ , telle que :

$$\Sigma_{pSMC(G)} = \min\left\{\sum_{i=1}^{|\mathbb{N}|} F_{\max}(v) \mid F(v) \text{ appartient à une multi-coloration valide}\right\}$$

**Définition 69 npSMC**

Le problème de somme multi-coloration non-préemptive, notée npSMC, a pour objectif de déterminer la somme multi-coloration chromatique, notée  $\Sigma_{npSMC(G)}$ , telle que :

$$\Sigma_{npSMC(G)} = \min\left\{\sum_{i=1}^{|\mathbb{N}|} F_{\max}(v) \mid F(v) \text{ appartient à une multi-coloration valide}\right\}$$

Tout en respectant la contrainte suivante :

$$F_{\max}(v) = F_{\min}(v) + (l(v) - 1)$$

**Exemple 18** Soit le graphe  $G$  de la Figure 1.4 tel que  $\forall v \in V, l(v) = 2$ . La solution optimale pour le problème pSMC et npSMC est donnée par la Figure 1.22 ( $\Sigma_{pSMC(G)} = \Sigma_{npSMC(G)} = 20$ ).

Nous remarquons qu'une solution optimale au problème pSMC n'est pas nécessairement une solution optimale au problème npSMC.

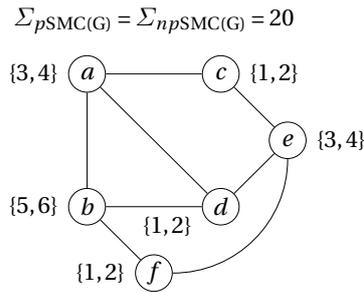


FIGURE 1.22 – Une solution optimale pour le problème  $pSMC$  et  $npSMC$ .

### 1.4.5 Problème de la somme coloration minimum d'un graphe

Le problème de la somme coloration minimum d'un graphe ( $MSCP$ ), est un problème de coloration introduit dans [Kubicka et Schwenk \(1989\)](#). Tout comme le problème  $OCCP$ , un poids est associé à chaque couleur. Contrairement au problème  $OCCP$ ,  $MSCP$  associe un poids équivalent à la valeur de la couleur. Comme précisé dans la Définition 53, une couleur est représentée par un entier. L'objectif est de minimiser la somme des poids des couleurs. Une solution optimale pour  $MSCP$  est appelée la *somme chromatique* du graphe.

#### Définition 70 $MSCP$

Soit  $G$  un graphe. Le problème de la somme coloration minimum ( $MSCP$ ) d'un graphe  $G$  consiste à déterminer la somme chromatique de  $G$ , notée  $\Sigma(G)$ , telle que :

$$\Sigma(G) = \min \left\{ \sum_{i=1}^{|X|} i \times |X_i| \mid X \text{ est une coloration valide de } G \right\}$$

La somme coloration associée à une coloration valide  $X$  est notée  $\Sigma(X)$ .

**Exemple 19** Soit le graphe  $G$  de la Figure 1.4. Nous associons le poids 1 à la couleur rouge, 2 à la couleur jaune et 3 à la couleur bleue. Une solution optimale pour  $MCP$  est donnée par la Figure 1.23 ( $\Sigma(G) = 10$ ).

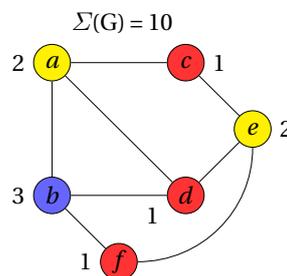


FIGURE 1.23 – Une solution optimale pour le problème  $MSCP$ .

Nous allons à présent définir formellement la *force d'un graphe* qui est le plus petit nombre de couleurs nécessaire à l'obtention d'une solution optimale pour  $MSCP$ .

#### Définition 71 Force d'un graphe

Soit  $G = (V, E)$  un graphe. La force d'un graphe, notée  $s(G)$  se définit comme suit :

$$s(G) = \min \{ |X| \mid X \text{ est une solution optimale} \}$$

Nous remarquons que suivant la Définition 63, *MSCP* peut être considéré comme un cas particulier du problème *OCCP*. De plus, *MSCP* peut également être considéré comme un problème *npSMC* et *pSMC* dont la cardinalité de l'ensemble des couleurs à assigner à chaque sommet est égale à 1.

Pour un graphe arbitraire, *MSCP* a été prouvé *NP*-difficile dans Kubicka et Schwenk (1989). Cependant, pour certaines familles de graphes tels que les arbres ou les graphes bipartis de degré inférieur ou égal à 3, la détermination de la somme chromatique se fait en temps polynomial Kroon *et al.* (1996); Giaro *et al.* (2002). Pour la famille des split-graphes, il est démontré dans Salavatipour (2003) que la résolution de *MSCP* est *NP*-difficile. Cependant, la résolution de *MSCP* pour la famille des  $k_C$ -split-graphes est en temps polynomial (Salavatipour (2003)).

## 1.5 Conclusion

À travers ce chapitre nous avons introduit différentes notations et rappelé les différentes notions nécessaires à la bonne lecture de ce manuscrit. Concrètement, nous avons dans un premier temps fait une brève présentation des classes de complexités pour pouvoir situer le problème de somme coloration par rapport aux autres problèmes de décision et d'optimisation. Puis, nous avons introduit le problème de décision SAT qui fut le premier problème à avoir été démontré *NP*-complet. Nous avons également, dans ce chapitre, rappelé quelques définitions concernant la théorie des graphes et présenté quelques familles de graphe. Puis nous avons terminé en introduisant la notion de coloration et le problème phare de ce manuscrit : le problème de la somme coloration minimum d'une graphe.

## Chapitre 2

# Résolution du Problème de la Somme Coloration Minimum d'un Graphe

### Sommaire

---

|   |           |
|---|-----------|
| <b>2.1 Approche non exhaustive pour la résolution de MSCP</b> . . . . . | <b>24</b> |
| 2.1.1 Algorithmes gloutons . . . . .                                    | 24        |
| 2.1.2 Recherches locales . . . . .                                      | 26        |
| 2.1.3 Algorithmes évolutionnaires . . . . .                             | 29        |
| <b>2.2 Approche complète pour MSCP</b> . . . . .                        | <b>31</b> |
| 2.2.1 Programme linéaire en nombres entiers . . . . .                   | 31        |
| 2.2.2 Programmation par contraintes . . . . .                           | 33        |
| <b>2.3 Benchmark</b> . . . . .  | <b>37</b> |
| 2.3.1 Graphe de Mycielski . . . . .                                     | 37        |
| 2.3.2 Graphe SGB . . . . .  | 37        |
| 2.3.3 Graphe DSJC . . . . .   | 38        |
| 2.3.4 Graphe de Leighton . . . . .                                      | 38        |
| 2.3.5 Graphe d'allocation . . . . .                                     | 39        |
| 2.3.6 Graphe flat . . . . .   | 40        |
| <b>2.4 Conclusions</b> . . . . .  | <b>40</b> |

---

Le problème de la somme coloration minimum (*MSCP*) d'un graphe est un problème d'optimisation hautement combinatoire. Les différentes méthodes de résolution dédiées à ce problème sont classifiables en deux catégories. La première catégorie regroupe les approches dites *exhaustives* ou *exactes* et la seconde catégorie regroupe les approches dites *non exhaustives* ou *incomplètes*. Une méthode exhaustive permet d'extraire l'optimum d'un problème. Dans le cas des méthodes non exhaustives, la coloration obtenue est la meilleure parmi un sous-ensemble de colorations considérées, mais ne peut être prouvée optimum pour l'ensemble des colorations possibles. Cependant, l'intérêt majeur de ces méthodes est leur temps de résolution. Parmi ces approches, nous trouvons des méthodes gloutonnes telles que *EXSCOL* Wu et Hao (2012b), *MD-SAT* Li et al. (2009), *MRLF* Li et al. (2009), des méthodes à base de recherche locale telle que *BLS* Benlic et Hao (2012), *MDS(5)+LS* Helmar et Chiarandini (2011), *IDCA* Sghiouer (2011) et *TS* Bouziri et Jouini (2010), et des algorithmes évolutionnaires tels que *MA-MSCP* Sghiouer (2011); Moukrim et al. (2014), *MASC* Jin et al. (2014), *GPX-FF* Kokosiński et Kwarcianny (2007) et *HESA* Jin et Hao (2016). Dans ce chapitre, nous n'abordons pas l'ensemble de ces méthodes de résolution. Nous nous intéressons essentiellement aux principes généraux. Cependant, certaines de ces méthodes sont présentées afin d'illustrer le fonctionnement de ces algorithmes pour la résolution de *MSCP*.

En ce qui concerne les méthodes exhaustives, il n'existe pas, à notre connaissance, d'approche de résolution dédiée à *MSCP* pour des graphes arbitraires. Un ensemble d'auteurs a étudié l'obtention d'un optimum de *MSCP* à travers différents paradigmes tels que la programmation par contraintes, la programmation linéaire en nombres entiers, etc. Compte tenu de la complexité de

*MSCP*, ces paradigmes ont une efficacité limitée pour les graphes arbitraires. Cependant, pour certaines familles de graphes des méthodes de résolution exhaustives efficaces ont été développées.

Ce chapitre est organisé comme suit. La section 1 présente différentes approches non exhaustives pour *MSCP*. La section 2 introduit la modélisation de *MSCP* selon différents paradigmes de résolution. La section 3 présente différentes familles d'instances sur lesquelles nous basons nos benchmarks ainsi que différentes caractéristiques connues de ces graphes. Pour terminer, la section 4 conclut ce chapitre.

## 2.1 Approche non exhaustive pour la résolution de *MSCP*

Pour extraire une solution, les méthodes de résolution de problèmes d'optimisation combinatoire, explorent l'ensemble des configurations possibles. Cet ensemble de configurations est appelé *espace de recherche* ou *espace des solutions*. Dans le cas de *MSCP*, nous définissons l'espace de recherche de la façon suivante :

### Définition 72 *Espace de recherche de *MSCP**

Soit  $G$  un graphe. L'espace de recherche de *MSCP*, noté  $\Omega$ , se définit comme suit :

$$\Omega = \{X \mid X \text{ est une coloration valide de } G\} \cup \{X \mid X \text{ est une coloration non valide de } G\}$$

La cardinalité de l'espace de recherche d'un problème d'optimisation est dans la majorité des cas de taille importante. Pour le cas de *MSCP*, si nous considérons un graphe  $G=(V,E)$ , la taille de l'espace de recherche est de  $k^{|V|}$ , où  $k$  est le nombre de couleurs initialement considéré. Par conséquent, l'énumération de l'ensemble des colorations est impossible même pour des graphes de petite taille. L'intérêt des méthodes non exhaustives est qu'elles renforcent la recherche dans un sous-ensemble de l'espace de recherche afin d'extraire la meilleure solution possible pour ce sous-ensemble en un temps raisonnable.

Dans la partie suivante, nous présentons le principe de fonctionnement d'une heuristique gloutonne.

### 2.1.1 Algorithmes gloutons

Les heuristiques gloutonnes sont des approches de construction de solutions que nous pourrions qualifier de "one-shot". Leur objectif est de construire une solution unique incrémentalement sans possibilité de retour sur une décision prise au cours de la recherche. L'étude approfondie du problème est donc nécessaire afin d'optimiser l'impact des décisions prises au cours de la construction de la solution. De plus, l'étude des différentes familles d'instances permet également d'orienter la construction afin d'affiner la qualité de la solution finale.

Les algorithmes gloutons n'ont pas seulement un intérêt théorique. En pratique, du fait de l'unicité de la solution, le temps de calcul et le coût d'occupation mémoire dû à la construction de la solution sont raisonnables. Pour mieux appréhender les concepts liés au fonctionnement des méthodes gloutonnes, il est intéressant de considérer la vision proposée dans [Béjian](#) pour les définir. Pour Béjian, ces heuristiques sont aveuglées par leur "appétit démesuré". De ce fait, la capacité de raisonnement de ces méthodes se limite à l'instant présent. Ainsi, cela se traduit par une précipitation vers le choix donnant le meilleur résultat sur le moment, sans préoccupation des conséquences de ce choix sur l'avenir. Au final, ces méthodes ne se remettent donc jamais en question puisque tout choix effectué est définitif. Par conséquent, cette notion de choix est donc primordiale. Cela se traduit dans un algorithme glouton par les concepts de *fonction objectif* et *fonction de sélection*. La fonction objectif est spécifique au problème d'optimisation. La fonction objectif associe un score à chaque solution afin d'en évaluer la qualité et extraire la solution ayant le plus faible (haut) score si l'objectif est de minimiser (maximiser) la solution objectif. La fonction de sélection est la stratégie développée pour déterminer le candidat à affecter suivant un

ensemble de critères choisis pour qualifier les candidats. L'Algorithme 1 illustre le schéma de fonctionnement général d'un algorithme glouton. La fonction de sélection et la fonction objectif sont respectivement illustrées aux Lignes 5 et 6.

---

**Algorithme 1 :** Méthode de fonctionnement d'un algorithme glouton.

---

**Entrée :**  $I$  une instance du problème.  
**Sortie :**  $S$  une solution du problème.

```

1 début
2   Décomposition de  $I$  en une liste de candidats  $L$ ;
3    $S \leftarrow \emptyset$ ;
4   tant que  $L \neq \emptyset$  faire
5     Choisir un candidat  $l$  de  $L$ ;
6     Attribuer une valeur à  $l$ ;
7      $S \leftarrow S \cup \{l\}$ ;
8      $L \leftarrow L \setminus \{l\}$ ;
9   retourner  $S$ ;
```

---

### MDSAT et MRLF

*MDSAT* et *MRLF* sont deux heuristiques gloutonnes introduites dans Li *et al.* (2009) dédiées à la résolution de *MSCP*. Ces deux approches sont basées sur les célèbres algorithmes de coloration de graphe *DSATUR* proposé dans Brélaz (1979) et *RLF* proposé dans Leighton (1979). La méthode de fonctionnement de *MDSAT* et *MRLF* est similaire. L'Algorithme 2 synthétise leur schéma de fonctionnement pour *MSCP*.

---

**Algorithme 2 :** Méthode de fonctionnement MRLF(G) et MDSAT(G).

---

**Entrée :**  $G = (V, E)$  un graphe.  
**Sortie :**  $\Sigma$  une somme coloration associée à une coloration valide.

```

1 début
2    $\Sigma \leftarrow 0$ ;
3   tant que  $V \neq \emptyset$  faire
4     Choisir un sommet  $v$  de  $V$ ;
5      $\Sigma \leftarrow \Sigma + \min\{k \mid k \in \text{Disp}(v)\}$ ;
6      $V \leftarrow V \setminus \{v\}$ ;
7   retourner  $\Sigma$ ;
```

---

La différenciation entre ces deux méthodes se fait lors de la sélection du prochain sommet à colorer (Ligne 4). Avant de définir les critères de sélection de *MDSAT* (Définition 77) et *MRLF* (Définition 78), nous allons introduire quelques notations sur les critères de voisinage d'une solution.

Nous rappelons que pour un sommet  $v$  d'un graphe  $G$ ,  $\mathcal{N}^-(v)$  représente l'ensemble des sommets voisins non colorés de  $v$  (Définition 58) et  $\text{Disp}(v)$  l'ensemble des couleurs disponibles de  $v$  (Définition 59).

#### Définition 73 $d_{N_{dsat}^-}$

Soit  $G = (V, E)$  un graphe,  $v$  un sommet du graphe et  $k$  la plus petite couleur disponible de  $v$ .

$$d_{N_{dsat}^-}(v) = |\{u \in \mathcal{N}^-(v) \mid k = \min(\text{Disp}(v)) \text{ et } k \in \text{Disp}(u)\}|$$

#### Définition 74 $d_{N_{dsatNC}^-}$

Soit  $G = (V, E)$  un graphe,  $v$  un sommet du graphe et  $k$  la plus petite couleur disponible de  $v$ .

$$d_{N_{dsatNC}^-}(v) = |\{u \in \mathcal{N}^-(v) \mid k = \min(\text{Disp}(v)) \text{ et } k \notin \text{Disp}(u)\}|$$

**Définition 75**  $d_{N_{cminC}^-}$

Soit  $G = (V, E)$  un graphe,  $v$  un sommet du graphe et  $k$  la plus petite couleur disponible de  $v$ .

$$d_{N_{cminC}^-}(v) = |\{u \in \mathcal{N}^-(v) \mid k = \min(\text{Disp}(v)) \text{ et } k = \min(\text{Disp}(u))\}|$$

**Définition 76**  $d_{N_{cminNC}^-}$

Soit  $G = (V, E)$  un graphe,  $v$  un sommet du graphe et  $k$  la plus petite couleur disponible de  $v$ .

$$d_{N_{cminNC}^-}(v) = |\{u \in \mathcal{N}^-(v) \mid k = \min(\text{Disp}(v)) \text{ et } k \neq \min(\text{Disp}(u))\}|$$

**Définition 77** *MDSAT*

Soit  $G = (V, E)$  un graphe et  $V' \subseteq V$  l'ensemble des sommets de  $G$  non colorés. Le critère de sélection d'un sommet de *MDSAT* est défini comme suit :

$$\text{MDSAT}(G) = \{v \in V' \mid v = \min(d_{N_{dsatC}^-} / d_{N_{dsatNC}^-})\}$$

**Définition 78** *MRLF*

Soit  $G = (V, E)$  un graphe et  $V' \subseteq V$  l'ensemble des sommets de  $G$  non colorés. Le critère de sélection d'un sommet de *MRLF* est défini comme suit :

$$\text{MRLF}(G) = \{v \in V' \mid v = \min(d_{N_{cminC}^-} / d_{N_{cminNC}^-})\}$$

La philosophie des règles de sélection de *MDSAT* (Définition 77) et *MRLF* (Définition 78) est globalement similaire. Seulement, dans le cas de *MDSAT*, [Li et al. \(2009\)](#) considèrent l'impact de la coloration du sommet sur le degré de saturation des sommets voisins non colorés alors que dans le cas de *MRLF*, [Li et al. \(2009\)](#) considèrent l'impact de la coloration sur la plus petite couleur disponible des sommets voisins non colorés.

**EXCOL**

L'algorithme glouton *EXCOL* a été introduit dans [Wu et Hao \(2012b\)](#). Le schéma de fonctionnement de *EXCOL* est inspiré de celui *EXTRACOL* [Wu et Hao \(2012a\)](#). Pour construire une coloration, cette approche se base sur l'extraction d'un ensemble de stables de même cardinalité dans le but d'obtenir un ensemble de classes couleurs. L'Algorithme 3 synthétise le schéma de fonctionnement de *EXCOL*. La règle de sélection de *EXCOL* est définie par la fonction *extractiondesstables* (Ligne 6). Cette dernière permet l'extraction d'un ensemble disjoint de stables de cardinalités équivalentes. La fonction *extractiondesstables* est modélisée par l'Algorithme 4. Cette dernière commence par extraire un stable du graphe de cardinalité la plus importante possible (Ligne 6). Ensuite, un ensemble de stables de cardinalités équivalentes est recherché dans le graphe (Ligne 8-14). Pour terminer, le plus grand ensemble disjoint de stables parmi l'ensemble des stables sauvegardés durant la recherche est extrait (Ligne 15).

Wu s'est basé sur l'heuristique *ATS* présentée dans [Wu et Hao \(2013a\)](#) pour l'extraction de stables.

**2.1.2 Recherches locales**

La recherche locale est une approche de résolution consistant à parcourir le voisinage d'une solution initiale en se déplaçant de solution en solution. L'intérêt de cette approche est qu'elle permet une intensification de la recherche sur un sous-ensemble de l'espace de recherche. La notion de *voisinage* est un élément essentiel de la recherche locale.

**Définition 79** *Voisinage*

Le voisinage d'une solution  $S$  est une fonction :

$$\begin{aligned} \mathcal{V} : \Omega &\rightarrow 2^\Omega \\ S &\rightarrow \mathcal{V}(S) \end{aligned}$$

Où  $\mathcal{V}(S)$  est un ensemble de solutions de  $\Omega$  atteignable à partir d'une transformation donnée.

---

**Algorithme 3 :** Méthode de fonctionnement de EXCOL( $G, try_{\max}$ ).

---

**Entrées :**  $G = (V, E)$  un graphe,  $try_{\max}$  le nombre maximal de tentatives de construction d'un stable et  $stable_{\max}$  le nombre maximal de stables.

**Sortie :**  $\Sigma$  une somme coloration associée à une coloration valide.

```

1  début
2  |  $\Sigma \leftarrow 0$ ;
3  |  $I \leftarrow \emptyset$ ;
4  |  $k \leftarrow 1$ ;
5  | tant que  $V \neq \emptyset$  faire
6  | |  $I \leftarrow \text{extractiondestables}(G, try_{\max}, stable_{\max})$ ;
7  | | pour chaque  $I'$  de  $I$  faire
8  | | |  $\Sigma \leftarrow \Sigma + k \times |I'|$ ;
9  | | |  $V \leftarrow V \setminus I'$ ;
10 | | |  $k \leftarrow k + 1$ ;
11 | retourner  $\Sigma$ ;

```

---



---

**Algorithme 4 :**  $\text{extractiondestables}(G, try_{\max}, stable_{\max})$ .

---

**Entrées :**  $G = (V, E)$  un graphe,  $try_{\max}$  le nombre maximal de tentatives de construction d'un stable et  $stable_{\max}$  le nombre maximal de stables.

**Sortie :**  $I$  un ensemble de stables.

```

1  début
2  |  $I \leftarrow \emptyset$ ;
3  |  $I_{\text{set}} \leftarrow \emptyset$ ;
4  |  $I_{\text{inter}} \leftarrow \emptyset$ ;
5  |  $try \leftarrow 0$ ;
6  | extraction d'un stable de  $G$  de cardinalité aussi grande que possible dans  $I_{\max}$ ;
7  |  $I_{\text{set}} \leftarrow I_{\text{set}} \cup \{I_{\max}\}$ ;
8  | tant que  $try \leq try_{\max}$  et  $|I| \leq stable_{\max}$  faire
9  | | extraction d'un stable de  $G$  de cardinalité  $|I_{\max}|$  dans  $I_{\text{inter}}$ ;
10 | | si  $I_{\text{inter}} \notin I_{\text{set}}$  alors
11 | | |  $I_{\text{set}} \leftarrow I_{\text{set}} \cup \{I_{\text{inter}}\}$ ;
12 | | |  $try \leftarrow 0$ ;
13 | | sinon
14 | | |  $try \leftarrow try + 1$ ;
15 |  $I \leftarrow \max\{I' \subseteq I_{\text{set}} \mid \forall I_u, I_v \in I_{\text{set}}, I_u \cap I_v = \emptyset\}$ ;
16 | retourner  $I$ ;

```

---

L'objectif d'un problème d'optimisation est de trouver une solution optimale. Durant l'exploration de l'espace de recherche, une méthode de recherche doit à partir d'une solution courante, juger de la qualité des solutions voisines. Pour cela, la *fonction d'évaluation* permet de quantifier la qualité d'une solution. Dans le cas de *MSCP*, nous définissons la fonction d'évaluation comme suit :

**Définition 80 fonction d'évaluation**

Soit  $c$  une coloration. La fonction d'évaluation  $f$  est une fonction :

$$\begin{aligned} f: \Omega &\rightarrow \mathbb{N} \\ c &\rightarrow \Sigma(c) \end{aligned}$$

À partir d'une solution courante, il n'est pas toujours possible de trouver une configuration voisine permettant d'améliorer la solution courante. Dans ce cas de figure, la solution courante est appelée *optimum local*. Si cet optimum est une solution optimale du problème alors il est appelé *optimum global*. Cependant, nous remarquons qu'une méthode de recherche locale ne peut prouver l'optimalité de la solution du fait qu'elle ne considère qu'un sous-ensemble de solutions de l'espace de recherche. Dans le cas de *MSCP*, les notions d'optimum local et d'optimum global se définissent formellement comme suit :

**Définition 81 Optimum local**

Une coloration  $c$  est un *optimum local* si  $\forall c' \in \mathcal{V}(c), \Sigma(c) \leq \Sigma(c')$

**Définition 82 Optimum global**

Une coloration  $c$  est un *optimum global* si  $\forall c' \in \Omega, \Sigma(c) \leq \Sigma(c')$

La première phase d'une méthode de recherche locale consiste à générer un chemin reliant une solution initiale à un optimum local. Cette phase est appelée *phase de descente*. Cependant, quand un optimum est atteint, il est nécessaire de s'en extraire. Cette deuxième phase est appelée *phase de dégradation*. La dégradation consiste à sélectionner une solution voisine n'améliorant pas la solution courante. La dégradation est une phase importante dans la recherche locale, car elle nécessite de dégrader suffisamment la solution courante afin de ne pas retomber dans un optimum local déjà visité. Pour cela, de nombreuses stratégies d'échappement ont été étudiées dans la littérature telles que le recuit simulé [Kirkpatrick et al. \(1983\)](#), la recherche tabou [Glover \(1989\)](#), etc. Les phases de descente et de dégradation sont illustrées par la Figure 2.1 qui schématise un paysage de recherche dans lequel évolue un algorithme de recherche locale.

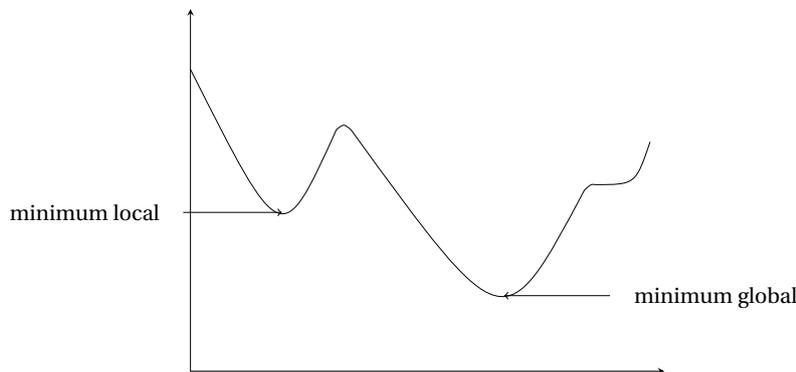


FIGURE 2.1 – Paysage d'une méthode de recherche locale.

L'Algorithme 5 schématise le fonctionnement d'une méthode de recherche locale.

---

**Algorithme 5** : Méthode de fonctionnement d'un algorithme de recherche locale.

---

**Entrée** :  $I$  une instance du problème.  
**Sortie** :  $s$  une solution du problème.

```

1  début
2  | Générer une solution initiale  $s' \in \Omega_1$  ;
3  |  $s \leftarrow s'$  ;
4  | tant que condition d'arrêt non vérifiée faire
5  |   | si une descente est possible pour  $s'$  alors
6  |   |   | Remplacer  $s'$  par une solution de  $\mathcal{V}(s)$  permettant la descente ;
7  |   |   |  $s \leftarrow s'$ 
8  |   | sinon
9  |   |   | Remplacer  $s'$  par une solution suivant un critère d'échappement ;
10 | retourner  $s$  ;

```

---

## BLS

L'algorithme de recherche locale *BLS* (Breakout Local Search) est une approche proposée dans Benlic et Hao (2012) pour la résolution de *MSCP*. Le schéma de fonctionnement de *BLS* est donné par l'Algorithme 6. Pendant la phase de descente, l'algorithme *BLS* propose deux fonctions de voisinage selon l'état de la coloration courante. Dans le cas où cette dernière est valide, la procédure analyse les échanges entre les classes couleurs pouvant permettre une amélioration de la solution courante (Ligne 9). Dans le cas où la solution courante n'est pas valide, une recherche est effectuée afin de réparer les différents conflits existants. Lorsqu'un conflit est détecté, un ensemble de mouvements est effectué pour atteindre une coloration valide. À la fin de cette procédure, une coloration valide est obtenue et la fonction de voisinage précédemment décrite est appliquée (Ligne 9). Pour s'extraire d'un optimum local, *BLS* introduit un degré de diversification. Le degré de diversification permet une perturbation de la solution courante selon une certaine intensité. Dans le cas d'un degré de diversification faible, les modifications de la solution sont faibles. Cela a pour conséquence une stagnation de la recherche. À l'inverse, un degré de diversification élevé est comparable à la génération d'une nouvelle solution aléatoire. Dans le cas de *BLS*, le degré de diversification est faible si la solution évolue (Ligne 17), augmenté si la solution est similaire au dernier optimum local visité (Ligne 15) et élevé si la solution n'évolue pas après un nombre d'itérations prédéfinis (Ligne 13).

### 2.1.3 Algorithmes évolutionnaires

L'algorithme évolutionnaire est une approche de recherche de résolution qui trouve son inspiration dans le darwinisme. Sa méthode de fonctionnement, modélisée dans l'Algorithme 7, repose sur l'évolution d'une solution à partir d'un ensemble de solutions par des mécanismes de sélection (Ligne 4), de reproduction (Ligne 5), de mutation (Ligne 6) et d'évolution (Ligne 7). Contrairement à une approche de type recherche locale, une approche évolutionnaire permet une plus grande diversification de l'espace de recherche à travers les différentes solutions formant sa population.

#### MA-MSCP

*MA-MSCP* est un algorithme évolutionnaire proposé dans Moukrim *et al.* (2014) combinant une approche génétique avec une recherche locale. L'objectif de la méthode de recherche locale proposée est de faire muter l'individu obtenu après croisement. La méthode de recherche locale proposée combine trois types de voisinages différents intervenant de façon aléatoire tout au long de la recherche. Ce type de mutation permet une intensification de la recherche.

---

**Algorithme 6 :** Méthode de fonctionnement de  $BLS(G, degree_{init}, degree_{max})$ .

---

**Entrées :**  $G = (V, E)$  un graphe,  $degree_{init}$  un degré de perturbation initial et  $degree_{max}$  un degré de perturbation maximal.

**Sortie :**  $c$  une coloration.

```

1  début
2  Générer une coloration initiale  $c' \in \Omega_1$  ;
3   $c \leftarrow c'$  ;
4   $c_{lastoptimum} \leftarrow c'$  ;
5   $degree \leftarrow degree_{init}$  ;
6  tant que condition d'arrêt non vérifiée faire
7      si  $c'$  est une coloration non valide alors
8          Appliquer recherche local avec réparation sur  $c'$  ;
9          Appliquer recherche local classique sur  $c'$  ;
10     si  $\Sigma(c') < \Sigma(c)$  alors
11          $c \leftarrow c'$  ;
12     si la recherche n'évolue plus après un certain nombre d'itération alors
13          $degree \leftarrow degree_{max}$  ;
14     sinon si  $c_{lastoptimum} = c'$  alors
15          $degree \leftarrow degree + 1$  ;
16     sinon
17          $degree \leftarrow degree_{init}$  ;
18      $c_{lastoptimum} \leftarrow c'$  ;
19     Perturber la solution  $c'$  avec le facteur de perturbation  $degree$  ;
20 retourner  $c$  ;

```

---



---

**Algorithme 7 :** Méthode de fonctionnement d'un algorithme évolutionnaire.

---

**Entrée :**  $I$  une instance du problème.

**Sortie :**  $s$  une solution du problème.

```

1  début
2  Générer un ensemble solution  $S \subseteq \Omega_1$  ;
3  tant que condition d'arrêt non vérifiée faire
4      Sélectionner deux individus  $s_1$  et  $s_2$  de  $S$  ;
5      Croiser  $s_1$  et  $s_2$  afin d'obtenir un ensemble de solutions formant la descendance  $S_f$  ;
6      Muter les individus de  $S_f$  selon une probabilité ;
7      Remplacement d'un ensemble de la population de  $S$  par un ensemble de  $S_f$  ;
8  retourner  $s$  la meilleure solution de  $S$  ;

```

---

L'algorithme 8 modélise le fonctionnement de *MA-MSCP*. Dans le cas de *MA-MSCP*, la génération de la population initiale est faite aléatoirement (Ligne 2). Cependant, sur une petite partie de la population, une heuristique de type destruction/construction est utilisée pour la génération. La phase de croisement (Ligne 5) implémentée s'inspire de celle proposée dans Galinier et Hao (1999) pour le problème de coloration de graphe. Nous remarquons également le croisement utilisé pour *MA-MSCP*.

---

**Algorithme 8 :** Méthode de fonctionnement de *MA-MSCP*(G).

---

**Entrée :**  $G = (V, E)$  un graphe.  
**Sortie :**  $c$  une coloration.

```

1  début
2  | Générer un ensemble de colorations  $C \subseteq \Omega_1$  ;
3  | tant que condition d'arrêt non vérifiée faire
4  | | Sélectionner deux colorations  $c_1$  et  $c_2$  de  $C$  ;
5  | | Croiser  $c_1$  et  $c_2$  afin d'obtenir une coloration  $c'$  ;
6  | | si probabilité de mutation active alors
7  | | | Appliquer une recherche locale sur  $c'$  ;
8  | | si  $\Sigma(c')$  améliore la plus mauvaise somme coloration de  $C$  alors
9  | | | Insérer  $c'$  dans  $C$  ;
10 | | | Rejeter la coloration de  $C$  ayant plus mauvaise somme coloration ;
11 | retourner  $c$  la meilleure coloration de  $C$  ;

```

---

## 2.2 Approche complète pour MSCP

À notre connaissance, il n'existe pas de solveur de résolution exacte dédié à *MSCP* pour des graphes arbitraires. Cependant, des travaux ont étudié la modélisation de *MSCP* dans divers paradigmes de recherche afin de permettre l'utilisation de méthodes de résolution complètes dédiées à ces différents paradigmes.

### 2.2.1 Programme linéaire en nombres entiers

Le problème de programmation linéaire en nombres entiers *PLNE* est un problème d'optimisation combinatoire *NP*-difficile. L'objectif d'un problème *PLNE* est de déterminer les valeurs d'un ensemble de variables soumis à un ensemble de contraintes afin de maximiser ou de minimiser une fonction objectif.

**Définition 83 Programme linéaire en nombre entier**

Un programme linéaire en nombre entier est défini par :

- un ensemble  $V$  de variables telles que  $\forall v \in V, v \in \mathbb{Z}$  ;
- un ensemble  $C$  de contraintes ;
- une fonction objectif  $f : z = \sum_{i=1}^{|V|} w_i v_i$  où  $w_i \in W$  tel que  $W$  est un ensemble de coûts.

La modélisation de *MSCP* sous forme d'un *PLNE* est proposée dans Sen et al. (1992) de la façon suivante :

**Définition 84 Modélisation de MSCP en PLNE**

Soit  $G = (V, E)$  un graphe et  $k$  un nombre de couleurs fixé. *MSCP* se modélise en *PLNE* comme suit :

- une variable booléenne  $v_{ij}$  est défini pour chaque sommet  $i$  et chaque couleur  $j$ , tel que  $v_{ij} = 1$  si et seulement si le sommet  $i$  est coloré avec la couleur  $j$  ;
- à chaque sommet  $i$  doit être assigné une et une seule couleur  $j : \forall i \in \{1, \dots, |V|\}, \sum_{j=1}^k v_{ij} = 1$  ;
- deux sommets adjacents  $i$  et  $u$  ne peuvent être colorés avec une même couleur  $j : \forall (i, u) \in E$  et  $\forall j \in \{1, \dots, k\}, v_{ij} + v_{uj} \leq 1$  ;
- minimiser :  $\Sigma(G) = \sum_{i=1}^{|V|} \sum_{j=1}^k j \times v_{ij}$ .

**Exemple 20** Soit le graphe  $G$  de la Figure 2.2 et 3 le nombre de couleurs considérées. Le PLNE pour la résolution de MSCP est le suivant :

$$\text{Minimiser : } \Sigma(G) = \sum_{j=1}^3 j \times v_{aj} + j \times v_{bj} + j \times v_{cj} + j \times v_{dj}$$

$$\text{En fonction de : } \left\{ \begin{array}{l} v_{a1} + v_{a2} + v_{a3} + v_{a4} = 1 \\ v_{b1} + v_{b2} + v_{b3} + v_{b4} = 1 \\ v_{c1} + v_{c2} + v_{c3} + v_{c4} = 1 \\ v_{d1} + v_{d2} + v_{d3} + v_{d4} = 1 \\ v_{a1} + v_{b1} \leq 1 \\ v_{a2} + v_{b2} \leq 1 \\ v_{a3} + v_{b3} \leq 1 \\ v_{a1} + v_{c1} \leq 1 \\ v_{a2} + v_{c2} \leq 1 \\ v_{a3} + v_{c3} \leq 1 \\ v_{a1} + v_{d1} \leq 1 \\ v_{a2} + v_{d2} \leq 1 \\ v_{a3} + v_{d3} \leq 1 \\ v_{b1} + v_{c1} \leq 1 \\ v_{b2} + v_{c2} \leq 1 \\ v_{b3} + v_{c3} \leq 1 \\ v_{b1} + v_{d1} \leq 1 \\ v_{b2} + v_{d2} \leq 1 \\ v_{b3} + v_{d3} \leq 1 \\ v_{ij} \in \{0, 1\}, \forall i \in V \text{ et } \forall j \in \{1, \dots, 3\} \end{array} \right.$$

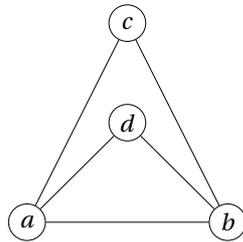


FIGURE 2.2 – Modélisation d'un graphe  $G = (V, E)$ .

Cette modélisation fut proposée dans Wang *et al.* (2013).

Dans Minot *et al.* (2016b, 2017) une amélioration du modèle PLNE (Définition 85) est proposée en considérant une décomposition du graphe en un ensemble de cliques, afin d'incorporer une contrainte d'exclusion mutuelle entre les sommets d'une clique.

**Définition 85 Modélisation améliorée de MSCP en PLNE**

Soit  $G = (V, E)$  un graphe,  $C_{clique}$  une décomposition de  $G$  en cliques et  $k$  un nombre de couleurs fixé. MSCP se modélise en PLNE comme suit :

- une variable booléenne  $v_{ij}$  est défini pour chaque sommet  $i$  et chaque couleur  $j$ , tel que  $v_{ij} = 1$  si et seulement si le sommet  $i$  est coloré avec la couleur  $j$  ;
- à chaque sommet  $i$  doit être assigné une et une seule couleur  $j : \forall i \in \{1, \dots, |V|\}, \sum_{j=1}^k v_{ij} = 1$  ;
- tous les sommets appartenant à une même clique de  $C_{clique}$  ne peuvent être colorés avec une même couleur :  $\forall i \in C$  tel que  $C \in C_{clique}, \sum_{j=1}^k v_{ij} \leq 1$  ;
- deux sommets adjacents  $i$  et  $u$  n'appartenant pas à une même clique de  $C_{clique}$  ne peuvent être colorés avec une même couleur  $j : \forall (i, u) \in E$  tel que  $\nexists C \in C_{clique}$  où  $\{i, u\} \subseteq C$  et  $\forall j \in \{1, \dots, k\}, v_{ij} + v_{uj} \leq 1$  ;
- minimiser :  $\Sigma(G) = \sum_{i=1}^{|V|} \sum_{j=1}^k j \times v_{ij}$ .

**Exemple 21** Soit le graphe  $G$  de la Figure 2.2,  $C_{clique} = \{a, b, c\}$  une décomposition de  $G$  en cliques et 3 le nombre de couleurs considéré. Le PLNE pour la résolution de MSCP est le suivant :

$$\text{Minimiser : } \Sigma(G) = \sum_{j=1}^3 j \times v_{aj} + j \times v_{bj} + j \times v_{cj} + j \times v_{dj}$$

$$\text{En fonction de : } \left\{ \begin{array}{l} v_{a1} + v_{a2} + v_{a3} + v_{a4} = 1 \\ v_{b1} + v_{b2} + v_{b3} + v_{b4} = 1 \\ v_{c1} + v_{c2} + v_{c3} + v_{c4} = 1 \\ v_{d1} + v_{d2} + v_{d3} + v_{d4} = 1 \\ v_{a1} + v_{b1} + v_{c1} \leq 1 \\ v_{a2} + v_{b2} + v_{c2} \leq 1 \\ v_{a3} + v_{b3} + v_{c3} \leq 1 \\ v_{a1} + v_{d1} \leq 1 \\ v_{a2} + v_{d2} \leq 1 \\ v_{a3} + v_{d3} \leq 1 \\ v_{b1} + v_{d1} \leq 1 \\ v_{b2} + v_{d2} \leq 1 \\ v_{b3} + v_{d3} \leq 1 \\ v_{ij} \in \{0, 1\}, \forall i \in V \text{ et } \forall j \in \{1, \dots, 3\} \end{array} \right.$$

## 2.2.2 Programmation par contraintes

### COP

La programmation par contraintes est un formalisme apparu dans les années 1970. Il est constitué d'un ensemble de variables et d'un ensemble de contraintes formant un réseau de contraintes. Formellement, un réseau de contraintes est défini comme suit dans [Montanari \(1974\)](#).

#### Définition 86 Réseau de contraintes

Un réseau de contraintes est un triplet  $\langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$  où :

- $\mathcal{X}$  est un ensemble fini de variables  $\{x_1, \dots, x_n\}$  ;
- $\mathcal{D}$  est un ensemble de domaines  $\{dom(x_1), \dots, dom(x_n)\}$  de sorte que  $\forall x \in \mathcal{X}, \exists ! dom(x) \in \mathcal{D}$  ;
- $\mathcal{C}$  est un ensemble de contraintes  $\{c_1, \dots, c_n\}$  où toute contrainte  $c_i \in \mathcal{C}$  est une relation portant sur un ensemble de variables appartenant à  $\mathcal{X}$ .

Nous avons proposé dans [Lecat et al. \(2015a,b\)](#) une modélisation de MSCP vers un problème d'optimisation de contraintes (COP). Un COP est un réseau de contraintes augmenté d'une fonction objectif. Formellement, nous le définissons de la façon suivante :

**Définition 87 COP**

Un problème d'optimisation de contraintes (COP) est un problème NP-difficile défini par un quadruplet  $\langle \mathcal{X}, \mathcal{D}, \mathcal{C}, \mathcal{F} \rangle$  où :

- $\mathcal{X}$  est un ensemble fini de variables  $\{x_1, \dots, x_n\}$  ;
- $\mathcal{D}$  est un ensemble de domaines  $\{dom(x_1), \dots, dom(x_n)\}$  de sorte que  $\forall x \in \mathcal{X}, \exists ! dom(x) \in \mathcal{D}$  ;
- $\mathcal{C}$  est un ensemble de contraintes  $\{c_1, \dots, c_n\}$  où toute contrainte  $c_i \in \mathcal{C}$  est une relation portant sur un sous-ensemble de variables appartenant à  $\mathcal{X}$  ;
- $\mathcal{F}$  est une fonction objectif  $\mathcal{F} : \mathcal{D}_{x_1} \times \dots \times \mathcal{D}_{x_n} \rightarrow \mathbb{N}$ .

**Définition 88 Modélisation de MSCP en COP**

Soit  $G = (V, E)$  un graphe et  $k$  un nombre fixé de couleurs. Le modèle COP pour la résolution de MSCP est le suivant :

- à chaque sommet  $i$  est associé une variable  $x_i : \mathcal{X} = \{x_1, \dots, x_{|V|}\}$  ;
- à chaque sommet  $i$  peut être assigné une couleur comprise entre 1 et  $k : dom(x_1) = \dots = dom(x_{|V|}) = \{1, \dots, k\}$  ;
- deux sommets adjacents  $i$  et  $u$  ne peuvent être colorés avec une même couleur :  $\forall (i, u) \in E, \exists c_{iu} \in \mathcal{C}$  tel que  $c_{iu} : x_i \neq x_u$  ;
- minimiser  $\mathcal{F} : \sum_{i=1}^{|V|} x_i$ .

**Exemple 22** Soit le graphe  $G$  de la Figure 2.2 et 3 le nombre de couleurs considéré. Le modèle COP pour la résolution de MSCP est le suivant :

- $\mathcal{X} = \{x_a, x_b, x_c, x_d\}$  ;
- $dom(x_a) = dom(x_b) = dom(x_c) = dom(x_d) = \{1, 2, 3\}$  ;
- $\mathcal{C} = \{x_a \neq x_b, x_a \neq x_c, x_a \neq x_d, x_b \neq x_c, x_b \neq x_d\}$  ;
- minimiser  $\mathcal{F} = x_a + x_b + x_c + x_d$ .

La Figure 2.3 représente le réseau de contraintes  $\langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$ .

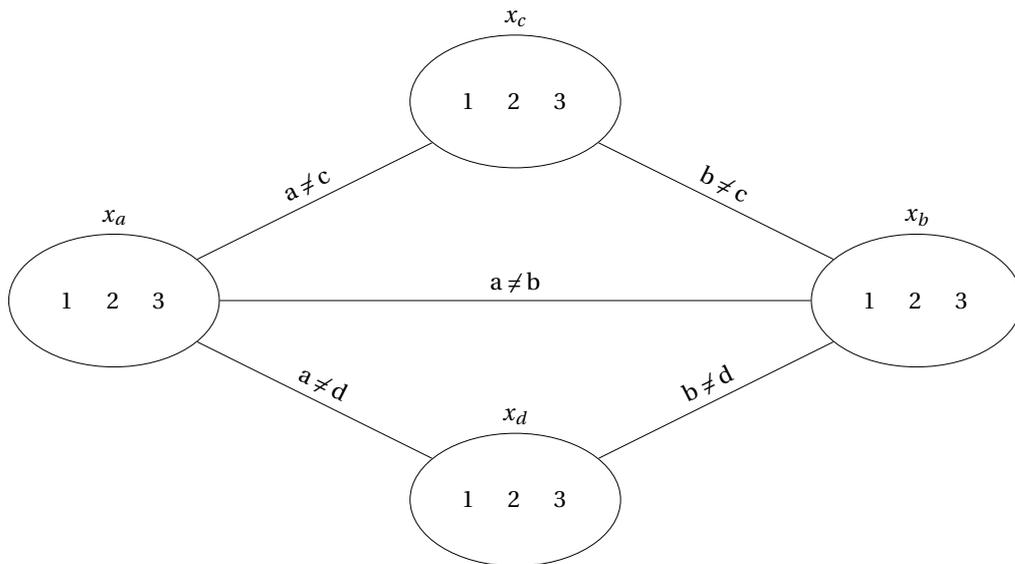


FIGURE 2.3 – Un réseau de contraintes pour le problème MSCP.

Nous remarquons qu'une contrainte peut impliquer plusieurs variables. Ainsi, le nombre de variables sur lequel porte la contrainte est appelé *arité* de la contrainte.

**Définition 89 Arité**

Soit un réseau de contraintes  $\langle X, D, C \rangle$ . L'arité d'une contrainte  $c$  de  $C$  est la cardinalité de l'ensemble des variables  $X_c \subseteq X$  constituant  $c$ . Une contrainte est dite unaire si  $|X_c| = 1$ , binaire si  $|X_c| = 2$  et  $|X_c|$ -aire si  $|X_c| \geq 3$ .

**Remarque 8** Une contrainte portant sur un sous-ensemble de variables dont la cardinalité n'est pas bornée est appelée contrainte globale.

Ainsi, si nous étudions la modélisation proposée dans [Lecat et al. \(2015a,b\)](#) (Définition 88), nous remarquons que les contraintes considérées sont binaires. De ce fait, comme dans la modélisation précédente, il est proposé dans [Minot et al. \(2016a,b, 2017\)](#) de considérer une décomposition du graphe en un ensemble de cliques afin d'incorporer à la modélisation une contrainte globale appelée *AllDifferent* qui associe une contrainte de différence à chaque variable constituant la contrainte *AllDifferent*.

**Définition 90 Modélisation améliorée de MSCP en COP**

Soit  $G = (V, E)$  un graphe,  $C_{clique}$  une décomposition de  $G$  en cliques et  $k$  un nombre de couleurs fixé. Le modèle COP pour la résolution de MSCP est le suivant :

- à chaque sommet  $i$  est associé une variable  $x_i$  :  $\mathcal{X} = \{x_1, \dots, x_{|V|}\}$  ;
- à chaque sommet  $i$  peut être assigné une couleur comprise entre 1 et  $k$  :  $dom(x_1) = \dots = dom(x_{|V|}) = \{1, \dots, k\}$  ;
- tous les sommets appartenant à une même clique  $C$  de  $C_{clique}$  ne peuvent être colorés avec une même couleur :  $\forall C \in C_{clique}, \exists c \in \mathcal{C}$  tel que  $c : AllDifferent(C)$  ;
- deux sommets adjacents  $i$  et  $u$  n'appartenant pas à une même clique de  $C_{clique}$  ne peuvent être colorés avec une même couleur  $j$  :  $\forall (i, u) \in E$  tel que  $\nexists C \in C_{clique}$  où  $\{i, u\} \subseteq C$ ,  $\exists c_{iu} \in \mathcal{C}$  tel que  $c_{iu} : x_i \neq x_u$  ;
- minimiser  $\mathcal{F} : \sum_{i=1}^{|V|} x_i$ .

**Exemple 23** Soit le graphe  $G$  de la Figure 2.2,  $C_{clique} = \{a, b, c\}$  une décomposition de  $G$  en cliques et 3 le nombre de couleur considéré. Le modèle COP pour la résolution de MSCP est le suivant :

- $\mathcal{X} = \{x_a, x_b, x_c, x_d\}$  ;
- $dom(x_a) = dom(x_b) = dom(x_c) = dom(x_d) = \{1, 2, 3\}$  ;
- $\mathcal{C} = \{AllDifferent(\{x_a, x_b, x_c\}), x_a \neq x_d, x_b \neq x_d\}$  ;
- minimiser  $\mathcal{F} = x_a + x_b + x_c + x_d$ .

**WCSP**

Dans [Minot et al. \(2016a\)](#) une modélisation de MSCP en un problème de satisfaction de contraintes pondérées (WCSP) est proposé. Contrairement à un réseau de contraintes classiques, l'ensemble de contraintes est représenté par une fonction de pondération qui associe un poids à un sous-ensemble de variables. Formellement, nous définissons un réseau de contraintes pondérées de la façon suivante :

**Définition 91 Réseau de contraintes pondérées**

Un réseau de contraintes pondérées est un triplet  $\langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$  où :

- $\mathcal{X}$  est un ensemble fini de variables  $\{x_1, \dots, x_{|V|}\}$  ;
- $\mathcal{D}$  est un ensemble de domaines  $\{dom(x_1), \dots, dom(x_{|V|})\}$  de sorte que  $\forall x \in \mathcal{X}, \exists ! dom(x) \in \mathcal{D}$  ;
- $\mathcal{C}$  est un ensemble de fonctions de coût  $f : 2^X \rightarrow \mathbb{N} \cup \{+\infty\}$ , qui associe à un sous-ensemble de variables de  $X$  un poids. Un coût  $+\infty$  représente un coût interdit.

Une approche de type *WCSP* ne nécessite pas l'intégration d'une fonction objectif. L'objectif d'un *WCSP* étant de minimiser la somme des coûts, une fonction de coût unaire correspondant à la couleur attribuée est donc associée à chaque variable. Nous remarquons également que la contrainte de différenciation peut être modélisée par un coût  $+\infty$ . Par conséquent, le problème *MSCP* se modélise en *WCSP* de la façon suivante :

**Définition 92 Modélisation de *MSCP* en *WCSP***

Soit  $G = (V, E)$  un graphe,  $C_{clique}$  une décomposition de  $G$  en cliques et  $k$  un nombre de couleurs fixé. Le modèle *WCSP* pour la résolution de *MSCP* est le suivant :

- à chaque sommet  $i$  est associé une variable  $x_i$  :  $\mathcal{X} = \{x_1, \dots, x_{|V|}\}$  ;
- à chaque sommet  $i$  peut être assigné une couleur comprise entre 1 et  $k$  :  $dom(x_1) = \dots = dom(x_{|V|}) = \{1, \dots, k\}$  ;
- à chaque couleur  $j$  assignée à un sommet  $i$  est associé un poids :  $\forall x_i \in \mathcal{X}, \exists f(x_i) \in \mathcal{C}$  tel que  $f(x_i) = j$  ;
- tous les sommets appartenant à une même clique de  $C_{clique}$  ne peuvent être colorés avec une même couleur :  $\forall C \in C_{clique}, \exists f(C) \in \mathcal{C}$  tel que  $f(C) = +\infty$  si  $\exists i, u \in C$  tel que  $x_i = x_u$ , 0 sinon ;
- deux sommets adjacents  $i$  et  $u$  n'appartenant pas à une même clique de  $C_{clique}$  ne peuvent être colorés avec une même couleur :  $\forall (i, u) \in E$  tel que  $\nexists C \in C_{clique}$  où  $\{i, u\} \subseteq C$ ,  $\exists f(x_i, x_u) \in \mathcal{C}$  tel que  $f(x_i, x_u) = +\infty$  si  $x_i = x_u$ , 0 sinon.

**Exemple 24** Soit le graphe  $G$  de la Figure 2.2,  $C_{clique} = \{\{a, b, c\}\}$  une décomposition de  $G$  en cliques et 3 le nombre de couleurs considéré. Le modèle *WCSP* pour la résolution de *MSCP* est le suivant :

- $\mathcal{X} = \{x_a, x_b, x_c, x_d\}$  ;
- $dom(x_a) = dom(x_b) = dom(x_c) = dom(x_d) = \{1, 2, 3\}$  ;
- $\mathcal{C} = \{f(x_a), f(x_b), f(x_c), f(x_d), f(x_a, x_d), f(x_b, x_d), f(x_a, x_b, x_c)\}$ .

Les coûts associés à chacune des fonctions sont donnés par la Figure 2.4.

|       |          |       |          |       |          |       |          |
|-------|----------|-------|----------|-------|----------|-------|----------|
| $x_a$ | $f(x_a)$ | $x_b$ | $f(x_b)$ | $x_c$ | $f(x_c)$ | $x_d$ | $f(x_d)$ |
| 1     | 1        | 1     | 1        | 1     | 1        | 1     | 1        |
| 2     | 2        | 2     | 2        | 2     | 2        | 2     | 2        |
| 3     | 3        | 3     | 3        | 3     | 3        | 3     | 3        |

|       |       |               |       |       |               |
|-------|-------|---------------|-------|-------|---------------|
| $x_a$ | $x_d$ | $f(x_a, x_d)$ | $x_b$ | $x_d$ | $f(x_b, x_d)$ |
| 1     | 1     | $+\infty$     | 1     | 1     | $+\infty$     |
| 1     | 2     | 0             | 1     | 2     | 0             |
| 1     | 3     | 0             | 1     | 3     | 0             |
| 3     | 3     | $+\infty$     | 3     | 3     | $+\infty$     |

|       |       |       |                    |
|-------|-------|-------|--------------------|
| $x_a$ | $x_b$ | $x_c$ | $f(x_a, x_b, x_c)$ |
| 1     | 1     | 1     | $+\infty$          |
| 1     | 1     | 2     | $+\infty$          |
| 1     | 1     | 3     | $+\infty$          |
| 1     | 2     | 1     | $+\infty$          |
| 1     | 2     | 2     | $+\infty$          |
| 1     | 2     | 3     | 0                  |
| 1     | 3     | 1     | $+\infty$          |
| 1     | 3     | 2     | 0                  |
| 1     | 3     | 3     | $+\infty$          |
| 3     | 3     | 3     | $+\infty$          |

FIGURE 2.4 – Coût associé aux différentes fonctions de coût de l'Exemple 24.

## 2.3 Benchmark

Dans cette section, nous introduisons les différentes familles d'instances sur lesquelles ont été réalisées l'ensemble de nos tests. Ces instances sont issues des benchmarks **COLOR** et **Dimacs**. Pour chaque graphe  $G$  de cette section est représenté dans les différentes colonnes, le nombre de sommets  $|V|$ , la densité du graphe  $d$ , le degré du graphe  $\Delta(G)$ , la meilleure borne connue du nombre chromatique  $k^*$  **Gualandi et Malucelli (2012)**; **Jin et al. (2014)**; **Malaguti et al. (2011)** (entre parenthèses quand  $k^*$  n'est pas optimale), la meilleure borne connue de la somme chromatique  $\Sigma^*$  **Jin et al. (2014)**; **Jin et Hao (2016)**.

### 2.3.1 Graphe de Mycielski

Les graphes de Mycielski sont des graphes sans triangle. En d'autres termes, il n'existe pas de clique de taille supérieure à 2. Ils ont été introduits dans **Mycielski (1955)**. Le nombre chromatique de ces graphes est corrélé avec le nombre de sommets. Si ce dernier croît alors le nombre chromatique croît également. Le tableau 2.1 synthétise un ensemble de résultats connus pour les graphes de Mycielski.

| Graphes | $ V $ | $d$   | $\Delta(G)$ | $k^*$ | $\Sigma^*$ |
|---------|-------|-------|-------------|-------|------------|
| myciel3 | 11    | 0.363 | 5           | 4     | 21         |
| myciel4 | 23    | 0.280 | 11          | 5     | 45         |
| myciel5 | 47    | 0.218 | 23          | 6     | 93         |
| myciel6 | 95    | 0.169 | 47          | 7     | 189        |
| myciel7 | 191   | 0.130 | 95          | 8     | 381        |

TABLEAU 2.1 – Graphes de Mycielski

### 2.3.2 Graphe SGB

Les graphes SGB ont été introduits dans **Knuth (1993)**. Ils se divisent en quatre catégories : *bookgraphs*, *games*, *miles* et *queens*. Un *bookgraph* est un graphe créé à partir d'une oeuvre littéraire. Un sommet du graphe représente un personnage et une arête est créée entre deux personnages si ces derniers interagissent. Les ouvrages considérés sont Anna Karenina (anna) **Tolstoy (1877)**, les Misérables (jean) **Hugo (1862)**, David Copperfield (david) **Dickens (1850)**, l'Iliade (homer) **Homère** et les aventures de Huckleberry Finn (huck) **Twain (1884)**. Le graphe *games* a été construit suivant la saison universitaire de football américain de 1990. Un sommet du graphe représente une équipe et une arête est modélisée entre deux sommets si ces deux équipes se sont affrontées lors de cette saison. Un graphe *miles* est une "cartographie" entre 128 villes nord-américaines. Une ville est donc modélisée par un sommet et une arête est générée entre deux sommets si une route relie les deux villes. Un graphe *queen* modélise un problème de  $n$ -reines. Un graphe représente un échiquier  $n \times m$ , un sommet modélise une case et une arête connecte deux sommets si ces deux cases sont sur une même ligne, même colonne ou une même diagonale. Une coloration valide de  $k$  couleurs signifie qu'il est possible de placer  $k$  ensembles de reines sur le plateau de sorte que les reines appartenant à un même ensemble ne peuvent se prendre entre elles. Le tableau 2.2 synthétise un ensemble de résultats connus pour les graphes SGB.

| Graphes  | $ V $ | $d$   | $\Delta(G)$ | $k^*$ | $\Sigma^*$ |
|----------|-------|-------|-------------|-------|------------|
| anna     | 138   | 0.052 | 71          | 11    | 276        |
| david    | 87    | 0.108 | 82          | 11    | 237        |
| huck     | 74    | 0.111 | 53          | 11    | 243        |
| jean     | 80    | 0.080 | 36          | 10    | 217        |
| homer    | 561   | 0.010 | 99          | 13    | 1150       |
| games120 | 120   | 0.089 | 13          | 9     | 443        |

Suite page suivante...

| Graphes    | $ V $ | $d$   | $\Delta(G)$ | $k^*$ | $\Sigma^*$ |
|------------|-------|-------|-------------|-------|------------|
| miles250   | 128   | 0.047 | 16          | 8     | 325        |
| miles500   | 128   | 0.143 | 38          | 20    | 705        |
| miles750   | 128   | 0.259 | 64          | 31    | 1173       |
| miles1000  | 128   | 0.395 | 86          | 42    | 1666       |
| miles1500  | 128   | 0.639 | 106         | 73    | 3354       |
| queen5_5   | 25    | 0.533 | 16          | 5     | 75         |
| queen6_6   | 36    | 0.460 | 19          | 7     | 138        |
| queen7_7   | 49    | 0.404 | 24          | 7     | 196        |
| queen8_8   | 64    | 0.361 | 27          | 9     | 291        |
| queen8_12  | 96    | 0.300 | 32          | 12    | 624        |
| queen9_9   | 81    | 0.325 | 32          | 10    | 409        |
| queen10_10 | 100   | 0.296 | 35          | 11    | 553        |
| queen11_11 | 121   | 0.272 | 40          | 11    | 733        |
| queen12_12 | 144   | 0.252 | 43          | 12    | 943        |
| queen13_13 | 169   | 0.234 | 48          | 13    | 1191       |
| queen14_14 | 196   | 0.219 | 51          | 14    | 1482       |
| queen15_15 | 225   | 0.205 | 56          | 15    | 1814       |
| queen16_16 | 256   | 0.193 | 59          | 16    | 2193       |

TABLEAU 2.2 – Graphes SGB

### 2.3.3 Graphe DSJC

La famille de graphes  $DSJCN_y$  est un ensemble de graphes aléatoires où  $n$  représente le nombre de sommets du graphe et  $y$  la probabilité avec laquelle une arête est ajoutée à chaque couple de sommets du graphe. Ils ont été introduits dans [Johnson \*et al.\* \(1991\)](#). Le tableau 2.3 synthétise un ensemble de résultats connus pour les graphes DSJC.

| Graphes    | $ V $ | $d$   | $\Delta(G)$ | $k^*$ | $\Sigma^*$ |
|------------|-------|-------|-------------|-------|------------|
| DSJC125.1  | 125   | 0.094 | 23          | 5     | 326        |
| DSJC125.5  | 125   | 0.502 | 75          | 17    | 1012       |
| DSJC125.9  | 125   | 0.898 | 120         | 44    | 2503       |
| DSJC250.1  | 250   | 0.103 | 38          | 8     | 970        |
| DSJC250.5  | 250   | 0.503 | 147         | (28)  | 3210       |
| DSJC250.9  | 250   | 0.896 | 234         | 72    | 8277       |
| DSJC500.1  | 500   | 0.099 | 68          | (12)  | 2836       |
| DSJC500.5  | 500   | 0.501 | 286         | (48)  | 10886      |
| DSJC500.9  | 500   | 0.901 | 471         | (126) | 29862      |
| DSJC1000.1 | 1000  | 0.099 | 127         | (20)  | 8991       |
| DSJC1000.5 | 1000  | 0.500 | 551         | (83)  | 37575      |
| DSJC1000.9 | 1000  | 0.899 | 924         | (222) | 103445     |

TABLEAU 2.3 – Graphes DSJC

### 2.3.4 Graphe de Leighton

Les graphes de Leighton ont été construits par Morgenstern en suivant la procédure de Leighton [Leighton \(1979\)](#). Ces graphes sont aléatoires. Cependant, pour chacun de ces graphes le nombre de sommets est fixe, 450 sommets. Le nombre chromatique est également prédéterminé. Le tableau 2.4 synthétise un ensemble de résultats connus pour les graphes de Leighton.

| Graphes   | $ V $ | $d$   | $\Delta(G)$ | $k^*$ | $\Sigma^*$ |
|-----------|-------|-------|-------------|-------|------------|
| le450_5a  | 450   | 0.056 | 42          | 5     | 1350       |
| le450_5b  | 450   | 0.056 | 42          | 5     | 1350       |
| le450_5c  | 450   | 0.097 | 66          | 5     | 1350       |
| le450_5d  | 450   | 0.096 | 68          | 5     | 1350       |
| le450_15a | 450   | 0.080 | 99          | 15    | 2632       |
| le450_15b | 450   | 0.080 | 94          | 15    | 2632       |
| le450_15c | 450   | 0.165 | 139         | 15    | 3487       |
| le450_15d | 450   | 0.165 | 138         | 15    | 3505       |
| le450_25a | 450   | 0.081 | 128         | 25    | 3153       |
| le450_25b | 450   | 0.081 | 111         | 25    | 3365       |
| le450_25c | 450   | 0.171 | 179         | 25    | 4515       |
| le450_25d | 450   | 0.172 | 157         | 25    | 4544       |

TABLEAU 2.4 – Graphes de Leighton

### 2.3.5 Graphe d'allocation

La famille de graphes d'allocation de registres a été générée dans [Lewandowski et Condon \(1994\)](#). Elle regroupe les graphes *fpsol2*, *inithx*, *mulsol* et *zeroin*. Ces graphes sont la modélisation de problèmes réels d'allocation de registres. L'objectif est d'affecter à chacune des opérations un registre tel que deux opérations nécessitant une même ressource ne puissent être affectées simultanément.

Les graphes *school* ont également été introduits dans [Lewandowski et Condon \(1994\)](#). Ils correspondent à un problème d'emplois du temps dans le milieu scolaire. Il est à noter que l'ensemble des cours souhaités est sélectionné par l'étudiant. L'objectif est d'affecter les étudiants à une section afin qu'ils puissent suivre les cours sans risque de cours simultanés. Un cours est par conséquent modélisé par un sommet, et une arête entre deux sommets est créée si un étudiant a sélectionné ces deux cours ou si les deux enseignements sont donnés par un même enseignant. Le tableau 2.5 synthétise un ensemble de résultats connus pour les graphes d'allocation.

| Graphes     | $ V $ | $d$   | $\Delta(G)$ | $k^*$ | $\Sigma^*$ |
|-------------|-------|-------|-------------|-------|------------|
| fpsol2_i_1  | 496   | 0.094 | 252         | 65    | 3403       |
| fpsol2_i_2  | 451   | 0.085 | 346         | 30    | 1668       |
| fpsol2_i_3  | 425   | 0.096 | 346         | 30    | 1636       |
| inithx.i.1  | 864   | 0.050 | 502         | 54    | 3676       |
| inithx.i.2  | 645   | 0.067 | 541         | 31    | 2050       |
| inithx.i.3  | 621   | 0.072 | 542         | 31    | 1986       |
| mulsol.i.1  | 197   | 0.203 | 121         | 49    | 1957       |
| mulsol.i.2  | 188   | 0.221 | 156         | 31    | 1191       |
| mulsol.i.3  | 184   | 0.232 | 157         | 31    | 1187       |
| mulsol.i.4  | 185   | 0.231 | 158         | 31    | 1189       |
| mulsol.i.5  | 186   | 0.230 | 159         | 31    | 1160       |
| zeroin.i.1  | 211   | 0.185 | 111         | 49    | 1822       |
| zeroin.i.2  | 211   | 0.159 | 140         | 30    | 1004       |
| zeroin.i.3  | 206   | 0.167 | 140         | 30    | 998        |
| school1     | 385   | 0.258 | 282         | 14    | 2674       |
| school1-nsh | 352   | 0.236 | 232         | 14    | 2392       |

TABLEAU 2.5 – Graphes d'allocation

### 2.3.6 Graphe flat

Les graphes flats sont des graphes introduits dans Culberson (1992). La génération de ces graphes se fait par le partitionnement des sommets en  $k$  ensembles de sommets de cardinalité égale. La construction des arêtes prend en considération que chaque ensemble de sommets préalablement défini forme un stable. Dans la plupart des problèmes de colorations, la résolution de ces graphes revient à restaurer le partitionnement initial. Le tableau 2.6 synthétise un ensemble de résultats connus pour les graphes flats.

| Graphes       | $ V $ | $d$   | $\Delta(G)$ | $k^*$ | $\Sigma^*$ |
|---------------|-------|-------|-------------|-------|------------|
| flat300-20-0  | 300   | 0.476 | 160         | 20    | 3150       |
| flat300-26-0  | 300   | 0.482 | 158         | 26    | 3966       |
| flat300-28-0  | 300   | 0.483 | 162         | 28    | 4238       |
| flat1000-50-0 | 1000  | 0.490 | 520         | 50    | 25500      |
| flat1000-60-0 | 1000  | 0.492 | 524         | 60    | 30100      |
| flat1000-76-0 | 1000  | 0.493 | 532         | 76    | 37164      |

TABLEAU 2.6 – Graphes flats

## 2.4 Conclusions

À travers ce chapitre, nous avons présenté différentes méthodes de résolution pour le problème *MSCP*. Pour cela, nous avons présenté différentes approches incomplètes (recherche locale, algorithme glouton, ...). Puis, nous nous sommes intéressé à la modélisation de *MSCP* en différents paradigmes (*SAT*, *CSP*, *PNL*, ...) afin de pouvoir appliquer des méthodes de résolutions complètes. Pour terminer, nous avons présenté l'ensemble des instances sur lesquelles portent les différentes expérimentations de ce manuscrit.

## Chapitre 3

# Nouvelles Bornes pour le Problème de la Somme Coloration Minimum d'un Graphe

### Sommaire

---

|  |           |
|--|-----------|
| <b>3.1 Coloration majeure et motif</b> . . . . .   | <b>42</b> |
| 3.1.1 Relation de dominance entre les motifs . . . . .                                   | 44        |
| 3.1.2 Ordre total sur $\phi(n)$ . . . . .  | 45        |
| <b>3.2 Nouvelles bornes pour la force d'un graphe</b> . . . . .                          | <b>47</b> |
| 3.2.1 Bornes supérieures de la littérature pour la force d'un graphe . . . . .           | 47        |
| 3.2.2 Principe général . . . . .   | 48        |
| 3.2.3 $UB_A$ une borne algébrique . . . . .  | 48        |
| 3.2.4 $UB_S$ une borne algorithmique . . . . .   | 50        |
| <b>3.3 Expérimentations de <math>UB_A</math> et de <math>UB_S</math></b> . . . . .       | <b>54</b> |
| <b>3.4 <math>LBM_\Sigma</math> une nouvelle borne pour la somme coloration</b> . . . . . | <b>58</b> |
| 3.4.1 Présentation de $LBM_\Sigma$ . . . . .   | 58        |
| 3.4.2 Résultats expérimentaux de $LBM_\Sigma$ . . . . .                                  | 59        |
| <b>3.5 Conclusions</b> . . . . .   | <b>62</b> |

---

La résolution de *MSCP* est un problème complexe. À ce jour, seuls les graphes de moins de 5000 sommets peuvent être résolus de manière exacte. L'une des difficultés majeures de *MSCP* est la complexité de son espace de recherche. Il est donc nécessaire d'effectuer un travail sur ce dernier en amont de la recherche afin de pouvoir réduire sa taille et extraire un ensemble d'informations pouvant aider les différentes méthodes de résolution existantes.

L'un des facteurs influant sur la taille de l'espace de recherche est le nombre de couleurs  $k$  initialement considéré. En effet, pour les différentes méthodes de résolution présentées dans le chapitre précédent, la taille de l'espace de recherche est équivalent à  $k^{|V|}$ . Une borne supérieure de la force d'un graphe de bonne qualité est donc primordiale pour optimiser la recherche afin de réduire au maximum  $k$ . La force d'un graphe peut être supérieure à celle du nombre chromatique. En effet, comme l'illustre la Figure 3.1, nous remarquons que bien que 2 couleurs suffisent à l'obtention d'une coloration valide, il est nécessaire de considérer une couleur supplémentaire afin d'obtenir la somme chromatique. De manière générale, la force d'un graphe peut être très éloignée du nombre chromatique. Par conséquent, l'espace de recherche de *MSCP* est plus compliqué à explorer que celui de *GCP*.

Dans ce chapitre, nous proposons deux nouvelles approches basées sur la notion de *motif* afin de borner la force d'un graphe arbitraire. Nous nous appuyons également sur ces motifs pour définir une borne inférieure de la somme chromatique. Une borne inférieure de la somme chromatique permet dans certains cas de prouver l'optimalité d'une solution d'une heuristique ou

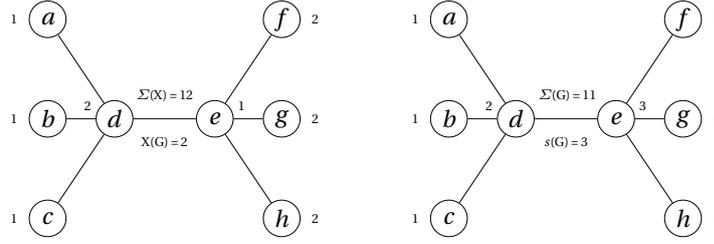


FIGURE 3.1 – Une solution optimale de *GCP*,  $X = \{\{a, b, c, e\}_1, \{d, f, g, h\}_2\}$  où  $\Sigma(X) = 12$ , et une solution optimale de *MSCP*,  $X' = \{\{a, b, c, f, g, h\}_1, \{d\}_2, \{e\}_3\}$  où  $\Sigma(X') = 11$ .

métho-heuristique pour *MSCP*. En effet, une borne inférieure de *MSCP* donne la garantie qu'il n'existe pas de solution valide telle que la somme coloration associée soit inférieure à celle-ci. Par conséquent, une coloration ayant pour somme chromatique une borne inférieure est une solution optimale de *MSCP*.

Ce chapitre est organisé de la manière suivante. Dans la section 1, nous introduisons les notions de coloration majeure et de motif. Dans la section 2, nous présentons deux nouvelles bornes supérieures de la force d'un graphe basées sur les motifs. Dans la section 3, nous présentons les bornes existantes de la littérature et étudions expérimentalement nos bornes supérieures. Dans la section 4, nous présentons et étudions expérimentalement notre borne inférieure de la somme chromatique. Pour terminer, la section 5 conclut ce chapitre.

Ces travaux ont donné lieu à plusieurs publications [Lecat et al. \(2017a,b,c,d,e\)](#).

### 3.1 Coloration majeure et motif

Comme nous l'avons remarqué, une coloration valide d'un graphe est un ensemble de classes couleurs tel que chaque classe couleur forme un stable. La permutation de deux classes couleurs d'une coloration n'impacte donc pas la validité de cette dernière. Par conséquent, il existe un ensemble de colorations symétriques  $\Psi(X)$  dans  $\Omega$  que nous définissons comme suit :

#### Définition 93 $\Psi(X)$

Soit  $G = (V, E)$  un graphe et  $X$  une coloration valide de  $G$ . L'ensemble  $\Psi(X) \subseteq \Omega$  est l'ensemble des permutations des classes couleurs de  $X$ .

**Exemple 25** Soit le graphe  $G$  de la Figure 1.4 et la coloration valide  $X = \{\{a, e\}_1, \{c, d, f\}_2, \{b\}_3\}$ . L'ensemble des colorations symétriques de  $X$  est :

$$\begin{aligned} \Psi(X) = & \{\{\{a, e\}_1, \{c, d, f\}_2, \{b\}_3\}, \\ & \{\{a, e\}_1, \{b\}_2, \{c, d, f\}_3\}, \\ & \{\{c, d, f\}_1, \{a, e\}_2, \{b\}_3\}, \\ & \{\{b\}_1, \{a, e\}_2, \{c, d, f\}_3\}, \\ & \{\{c, d, f\}_1, \{b\}_2, \{a, e\}_3\}, \\ & \{\{b\}_1, \{c, d, f\}_2, \{a, e\}_3\} \end{aligned}$$

Si la permutation d'un ensemble de classes couleurs d'une coloration valide n'impacte pas la validité de la solution ainsi obtenue, il n'en est pas de même pour la somme coloration associée à cette coloration de  $\Psi(X)$ . L'objectif de *MSCP* étant de déterminer la somme coloration minimum d'un graphe, nous extrayons de  $\Psi(X)$  la coloration ayant la plus petite somme coloration. Une telle coloration est dite majeure et se définit formellement comme suit :

#### Définition 94 Coloration majeure

Soit  $\Psi(X)$  un ensemble de colorations symétriques. Une coloration  $X$  de  $\Psi(X)$  est dite majeure, notée  $X^m$ , si  $\forall i \in [1, |X^m| - 1], |X_i^m| \geq |X_{i+1}^m|$ .

**Exemple 26** Soit  $\Psi(X)$  l'ensemble des colorations symétriques de l'Exemple 25. La coloration  $\{\{c, d, f\}_1, \{a, e\}_2, \{b\}_3\}$  est une coloration majeure de  $\Psi(X)$ .

De part la construction d'une coloration majeure, la propriété suivante est une conséquence directe de la Définition 94.

**Propriété 1** Soit  $\Psi(X)$  un ensemble de colorations symétriques. Si  $X^m$  est une coloration majeure de  $\Psi(X)$  alors  $\forall X' \in \Psi(X), \Sigma(X^m) \leq \Sigma(X')$ .

L'une des conséquences de la Propriété 1 est que pour l'obtention d'une solution pour *MSCP*, seule la considération des colorations majeures est nécessaire. Dans la suite de ce chapitre, nous considérons que toute coloration  $X$  est majeure.

Une coloration majeure étant une coloration dont les classes couleurs sont triées par ordre décroissant de leur cardinalité, il est donc possible d'associer à chacune de ces colorations une séquence d'entiers. Une telle représentation est appelée motif et se définit formellement comme suit :

**Définition 95 Motif**

Soit  $X$  une coloration majeure telle que  $X = \{X_1, X_2, \dots, X_k\}$ . Un motif est une séquence décroissante d'entiers  $p$  où chacun de ces entiers est égal à la cardinalité d'une classe couleur. La cardinalité d'un motif  $p$  est le nombre de couleurs utilisées par la coloration  $X$ , notée  $|p|$  ( $|p| = k$ ), et le  $i^{\text{ème}}$  entier de  $p$  est notée  $p[i]$  tel que  $p[i] = |X_i|$

La somme coloration associée à un motif  $p$  est définie par l'Équation 3.1

$$\Sigma(X) = \Sigma(p) = 1 \times p[1] + 2 \times p[2] + \dots + k \times p[k] \quad (3.1)$$

**Exemple 27** Soit la coloration majeure  $X = \{\{c, d, f\}_1, \{a, e\}_2, \{b\}_3\}$ . Le motif associé est  $p = (3, 2, 1)$ , avec  $p[1] = 3, p[2] = 2, p[3] = 1, |p| = 3$  et  $\Sigma(X) = \Sigma(p) = 10$ .

Puisqu'un motif est une séquence d'entiers, un même motif peut par conséquent représenter plusieurs colorations. De ce fait, considérer uniquement l'ensemble des motifs permet une abstraction non négligeable de l'espace de recherche de *MSCP*.

**Exemple 28** Soit le graphe  $G$  de la Figure 1.4 et deux colorations majeures  $X = \{\{c, d, f\}_1, \{b, e\}_2, \{a\}_3\}$  et  $X' = \{\{a, d, f\}_1, \{b, c\}_2, \{e\}_3\}$ . Le motif  $p$  associé aux colorations  $X$  et  $X'$  est  $p = (3, 2, 1)$

Pour un graphe et un nombre de couleurs  $k$  donné, il est possible de modéliser l'ensemble des motifs de cardinalité  $k$ . Un tel ensemble est noté  $\phi(n, k)$ .

**Définition 96  $\phi(n, k)$**

Soit  $G = (V, E)$  un graphe et  $k$  un nombre de couleurs. L'ensemble des motifs associés à  $G$  de cardinalité  $k$ , noté  $\phi(n, k)$  tel que  $n = |V|$ , est défini par :

$$\phi(n, k) = \{p \mid p \text{ est un motif, } |p| = k \text{ et } \sum_{i=1}^k p[i] = |V|\}$$

Ainsi, à l'aide de la Définition 96, il nous est possible de définir l'ensemble de tous les motifs associés à un graphe.

**Définition 97  $\phi(n)$**

Soit  $G = (V, E)$  un graphe et  $k$  un nombre de couleurs. L'ensemble des motifs associés à  $G$ , noté  $\phi(n)$  tel que  $n = |V|$ , est défini par :

$$\phi(n) = \bigcup_{k=1}^{|V|} \{\phi(n, k)\}$$

Nous remarquons que la construction de l'ensemble  $\phi(n)$  est similaire à la construction de l'ensemble des partitions d'un entier  $n$ . La cardinalité de cet ensemble croît exponentiellement suivant  $n$ . Dans [Hardy et Ramanujan \(1918\)](#), une approximation de la cardinalité de l'ensemble des partitions d'un entier est donnée. De ce fait, la cardinalité de  $\phi(n)$  est donnée par l'Équation 3.2.

$$|\phi(n)| \sim \frac{1}{4n\sqrt{3}} e^{\pi\sqrt{2n/3}} \quad (3.2)$$

Le tableau 3.1 donne une approximation de la cardinalité de  $\phi(n)$  selon une variation de  $n$ .

|             |   |   |   |   |   |    |    |    |    |    |     |         |             |                |
|-------------|---|---|---|---|---|----|----|----|----|----|-----|---------|-------------|----------------|
| n           | 1 | 2 | 3 | 4 | 5 | 6  | 7  | 8  | 9  | 10 | 20  | 50      | 100         | 150            |
| $ \phi(n) $ | 1 | 2 | 3 | 5 | 7 | 11 | 15 | 22 | 30 | 42 | 697 | 204 226 | 190 569 292 | 40 853 235 313 |

TABLEAU 3.1 – Cardinalité de  $\phi(n)$ .

Nous remarquons que tous les motifs de  $\phi(n)$  ne sont pas associés à des colorations valides du graphe.

**Exemple 29** Soit le graphe  $G$  de la Figure 1.4. Les motifs  $(3, 3)$  et  $(4, 1, 1)$  ne correspondent à aucune coloration valide.

Afin de ne pas considérer l'ensemble des motifs de  $\phi(n)$ , nos approches se basent sur la relation de dominance existante entre les motifs de  $\phi(n)$ . De ce fait, il n'est donc pas nécessaire d'énumérer  $\phi(n)$  mais uniquement un très petit sous-ensemble de motifs de  $\phi(n)$ . Nous allons à présent définir la relation de dominance entre les motifs.

### 3.1.1 Relation de dominance entre les motifs

La relation de dominance, notée  $\succeq$ , sur l'ensemble des motifs de  $\phi(n)$  a été introduite dans [Bonomo et Valencia-Pabon \(2009, 2014\)](#) afin de calculer la somme chromatique d'un sous-ensemble de graphes de la famille  $P_4$ -sparses. De plus, la relation de dominance introduite par Bonomo a également permis d'extraire une borne supérieure de la somme chromatique pour l'ensemble des graphes  $P_4$ -sparses. Cependant, dans l'approche considérée par Bonomo, un motif représentait la modélisation d'une coloration majeure partielle. Nous adaptons cette relation de dominance à notre approche.

#### Définition 98 Relation de dominance

Soit  $p$  et  $q$  deux motifs de  $\phi(n)$ .  $p$  domine  $q$ , noté  $p \succeq q$ , si et seulement si  $\forall t$  tel que

$$1 \leq t \leq \min\{|p|, |q|\}, \sum_{x=1}^t p[x] \geq \sum_{x=1}^t q[x].$$

**Exemple 30** Soit  $p$  et  $q$  deux motifs, tels que  $p = (5, 2, 1)$  et  $q = (4, 3, 1)$ .

Pour  $t = 1$  :  $5 > 4$  ;

Pour  $t = 2$  :  $5 + 2 \geq 4 + 3$  ;

Pour  $t = 3$  :  $5 + 2 + 1 \geq 4 + 3 + 1$ .

Donc  $p \succeq q$ .

Comme nous le remarquons dans la Définition 98, la principale différence est que notre approche considère uniquement l'ensemble des motifs correspondant à une coloration majeure complète. C'est-à-dire,  $\sum_{x=1}^{|p|} p[x] = \sum_{x=1}^{|q|} q[x] = |V|$ .

Cependant, la relation d'ordre existante entre deux motifs est une relation d'ordre partiel. En effet, deux motifs ne sont pas nécessairement comparables, comme montrés dans l'Exemple 31.

**Exemple 31** Soit  $G = (V, E)$  un graphe tel que  $|V| = 15$ , et  $p$  et  $q$  deux motifs de  $\phi(15)$  tels que  $p = (9, 3, 3)$  et  $q = (8, 6, 1)$ .

Pour  $t = 1 : 9 > 8$ ;

Pour  $t = 2 : 9+3 < 8+6$ .

Par conséquent,  $p \not\geq q$  et  $q \not\geq p$ . Donc,  $p$  et  $q$  ne sont pas comparables.

L'intérêt de la relation de dominance est l'impact qu'elle apporte sur la somme coloration associée aux motifs. Ainsi, la propriété suivante permet de comparer les sommes colorations associées à deux motifs, tels que l'un des motifs domine le second.

**Propriété 2** Soit  $G$  un graphe,  $c_1, c_2, \dots, c_k$  un ensemble de  $k$  couleurs et  $w_i$  le poids associé à chaque couleur  $c_i$  ( $1 \leq i \leq k$ ) tel que  $w_1 \leq w_2 \leq \dots \leq w_k$ . Soit  $p$  et  $q$  deux motifs correspondant respectivement aux colorations  $X$  et  $X'$  de  $G$ . Si  $p \geq q$ , alors  $\sum_{i=1}^{|p|} w_i |X_i| \leq \sum_{i=1}^{|q|} w_i |X'_i|$ .

**Preuve 1** Nous remarquons que si  $p \geq q$  alors  $|p| \leq |q|$ . En effet, si nous supposons que  $|p| > |q|$ , alors  $n = \sum_{i=1}^{|p|} p[i] > \sum_{i=1}^{|q|} p[i] \geq \sum_{i=1}^{|q|} q[i] = n$ . Or d'après la définition même de la dominance (Définition 98), cela est impossible. Donc  $|p| \leq |q|$  si  $p \geq q$ .

Par conséquent, nous pouvons réécrire  $\sum_{i=1}^{|p|} w_i |X_i| - \sum_{i=1}^{|q|} w_i |X'_i|$  comme  $\sum_{i=1}^{|p|} w_i (p[i] - q[i]) - \sum_{i=|p|+1}^{|q|} w_i q[i]$ .

Puisque  $\forall t$  tel que  $1 \leq t \leq |p|$ ,  $\sum_{i=1}^t p[i] \geq \sum_{i=1}^t q[i]$ , nous avons  $\sum_{i=1}^{|p|} w_i (p[i] - q[i]) \leq w_2 (p[1] - q[1]) + \sum_{i=2}^{|p|} w_i (p[i] - q[i]) = w_2 (p[1] + p[2] - q[1] - q[2]) + \sum_{i=3}^{|p|} w_i (p[i] - q[i]) \leq w_3 (p[1] + p[2] - q[1] - q[2]) + \sum_{i=3}^{|p|} w_i (p[i] - q[i]) = w_3 (p[1] + p[2] + p[3] - q[1] - q[2] - q[3]) + \sum_{i=4}^{|p|} w_i (p[i] - q[i]) \leq \dots \leq w_{|p|} (\sum_{i=1}^{|p|} p[i] - \sum_{i=1}^{|p|} q[i])$ .

Dans chacune des inégalités précédentes, nous pouvons remplacer  $w_t$  dans  $w_t (\sum_{i=1}^t p[i] - \sum_{i=1}^t q[i])$  avec  $w_{t+1}$  ( $w_{t+1} \geq w_t$ ), afin d'obtenir  $w_{t+1} (\sum_{i=1}^t p[i] - \sum_{i=1}^t q[i])$ . En effet,  $\sum_{i=1}^t p[i] - \sum_{i=1}^t q[i] \geq 0$ .

Comme  $\sum_{i=1}^{|p|} p[i] - \sum_{i=1}^{|p|} q[i] = n - \sum_{i=1}^{|p|} q[i] = \sum_{i=|p|+1}^{|q|} q[i]$ , nous avons  $\sum_{i=1}^{|p|} w_i |X_i| - \sum_{i=1}^{|q|} w_i |X'_i| \leq w_{|p|} \sum_{i=|p|+1}^{|q|} q[i] - \sum_{i=|p|+1}^{|q|} w_i q[i] \leq w_{|p|} \sum_{i=|p|+1}^{|q|} q[i] - \sum_{i=|p|+1}^{|q|} w_{|p|} q[i] = 0$  parce que  $w_i \geq w_{|p|}$  si  $i > |p|$ .

Par conséquent,  $\sum_{i=1}^{|p|} w_i |X_i| = \sum_{i=1}^{|p|} w_i p[i] \leq \sum_{i=1}^{|q|} w_i q[i] = \sum_{i=1}^{|q|} w_i |X'_i|$ .  $\square$

La Propriété 2 est une généralisation de la propriété pour *MSCP* proposée dans [Bonomo et Valencia-Pabon \(2009, 2014\)](#). L'intérêt de cette généralisation est de permettre d'étendre les propriétés présentées dans la suite de ce manuscrit au problème *OCCP*.

### 3.1.2 Ordre total sur $\phi(n)$

Comme nous l'avons vu précédemment, la relation de dominance introduite dans [Bonomo et Valencia-Pabon \(2009, 2014\)](#) est une relation d'ordre partiel permettant la comparaison de la somme coloration de deux motifs. Nous proposons dans [Lecat et al. \(2017c,d\)](#) une nouvelle relation d'ordre total afin d'ordonner l'ensemble  $\phi(n)$ . Pour cela, nous considérons que les ensembles  $\phi(n, k)$  de  $\phi(n)$  sont triés par ordre croissant de  $k$  et que les motifs d'un sous-ensemble  $\phi(n, k)$

sont triés par ordre lexicographique décroissant. Ainsi, l'ordre lexicographique décroissant permet d'assigner un index  $i$  à chaque motif de  $\phi(n, k) : p_k^i$  est le  $i^{\text{ème}}$  motif de  $\phi(n, k)$  dans l'ordre lexicographique décroissant.

**Définition 99 Ordre total**

Nous notons  $p_k^i$  le  $i^{\text{ème}}$  motif dans l'ordre lexicographique décroissant de  $\phi(n, k)$ . Soit  $p_k^i$  et  $p_{k'}^j$  deux motifs de  $\phi(n)$ . Le motif  $p_k^i$  précède (est plus grand que)  $p_{k'}^j$  dans  $\phi(n)$  si  $k < k'$  ou si  $k = k'$  et  $i < j$ .

La Table 3.2 présente l'ensemble des motifs de  $\phi(8)$  trié suivant l'ordre défini dans la Définition 99 ainsi que la somme coloration associée à chaque motif.

| $\phi(n,k)$ | $p \in \phi(n,k)$ | $\Sigma(p)$ |
|-------------|-------------------|-------------|
| $\phi(8,1)$ | (8)               | 8           |
| $\phi(8,2)$ | (7,1)             | 9           |
|             | (6,2)             | 10          |
|             | (5,3)             | 11          |
|             | (4,4)             | 12          |
| $\phi(8,3)$ | (6,1,1)           | 11          |
|             | (5,2,1)           | 12          |
|             | (4,3,1)           | 13          |
|             | (4,2,2)           | 14          |
|             | (3,3,2)           | 15          |
| $\phi(8,4)$ | (5,1,1,1)         | 14          |
|             | (4,2,1,1)         | 15          |
|             | (3,3,1,1)         | 16          |
|             | (3,2,2,1)         | 17          |
|             | (2,2,2,2)         | 20          |
| $\phi(8,5)$ | (4,1,1,1,1)       | 18          |
|             | (3,2,1,1,1)       | 19          |
|             | (2,2,2,1,1)       | 21          |
| $\phi(8,6)$ | (3,1,1,1,1,1)     | 23          |
|             | (2,2,1,1,1,1)     | 24          |
| $\phi(8,7)$ | (2,1,1,1,1,1,1)   | 29          |
| $\phi(8,8)$ | (1,1,1,1,1,1,1,1) | 36          |

TABLEAU 3.2 –  $\phi(8)$

Les Propriétés suivants s'intéressent à la construction du premier motif et du dernier motif de  $\phi(n, k)$ .

**Propriété 3** Le premier motif d'un ensemble  $\phi(n, k)$  est  $(n - k + 1, \overbrace{1, 1, \dots, 1}^{k-1 \text{ fois}})$ .

**Propriété 4** Le dernier motif d'un ensemble  $\phi(n, k)$  est  $(\overbrace{\left\lfloor \frac{n}{k} \right\rfloor, \dots, \left\lfloor \frac{n}{k} \right\rfloor}^{n \text{ mod } k \text{ fois}}, \overbrace{\left\lfloor \frac{n}{k} \right\rfloor, \dots, \left\lfloor \frac{n}{k} \right\rfloor}^{k - n \text{ mod } k \text{ fois}})$ .

L'Algorithme 9 énumère l'ensemble des motifs de  $\phi(n, k)$  suivant l'ordre lexicographique décroissant. Il prend en paramètre deux entiers positifs  $n$  et  $k$  tels que  $n \geq k$ , un motif  $p$  en cours de construction (initialement vide) et l'ensemble résultant  $\phi$  de motifs. À chaque appel récursif, l'algorithme commence par générer le premier entier devant être compris entre  $\left\lfloor \frac{n}{k} \right\rfloor$  et  $sup$ , où  $sup$  est égal à  $\min(n - k + 1, \min(p))$ , de sorte que l'ordre non croissant est respecté. À la fin de l'algorithme le paramètre initial  $\phi$  est l'ensemble de motifs  $\phi(n, k)$ .

**Algorithme 9 :** MOTIF( $n, k, p, \phi$ )

---

**Entrées :** Deux entiers positifs  $n$  et  $k$  tel que  $n \geq k$ ,  $p$  un motif en cours de construction, l'ensemble des motifs  $\phi$  de  $\phi(n, k)$  en cours de construction

**Sortie :**  $\phi$  l'ensemble des motifs de  $\phi(n, k)$  triés dans l'ordre lexicographique décroissant

```

1 début
2   si  $k = 1$  alors
3      $\phi \leftarrow \phi \cup \{p \cup \{n\}\}$ ;
4   sinon
5     si  $p = \emptyset$  alors
6        $sup \leftarrow n - k + 1$ ;
7     sinon
8        $sup \leftarrow \min(n - k + 1, \min(p))$ ;
9        $inf \leftarrow \lceil \frac{n}{k} \rceil$ ;
10    pour chaque  $x$  allant de  $sup$  à  $inf$  faire
11      Motif( $n - x, k - 1, p \cup \{x\}, \phi$ );

```

---

**Exemple 32** Pour générer l'ensemble des motifs de  $\phi(8, 3)$ , nous appelons la fonction MOTIF(8, 3,  $\emptyset, \emptyset$ ) ( $p = \phi = \emptyset$ ). L'algorithme génère chaque entier compris entre 6 et 3. Pour l'entier 6, l'algorithme appelle la fonction MOTIF(2, 2, {6},  $\emptyset$ ), qui génère l'unique partition de l'entier 2 de taille 2 (i.e., (1, 1)). Cela forme le motif complet (6, 1, 1). Pour l'entier 4, l'algorithme appelle la fonction MOTIF(4, 2, {4}, {(6, 1, 1), (5, 2, 1)}), qui génère les deux partitions de l'entier 4 de taille 2, (3, 1) et (2, 2) formant les motifs complets (4, 3, 1) et (4, 2, 2).

## 3.2 Nouvelles bornes pour la force d'un graphe

Dans cette section, après un rappel des bornes pour la force d'un graphe de la littérature, nous présentons deux nouvelles bornes. La première, appelée  $UB_A$ , est une borne algébrique, tandis que la seconde, appelée  $UB_S$ , est une borne algorithmique.

### 3.2.1 Bornes supérieures de la littérature pour la force d'un graphe

Dans la littérature, différentes relations ont été établies entre les propriétés structurelles d'un graphe  $G$  (degré, nombre chromatique) et sa somme chromatique [Bar-Noy et Kortsarz \(1998\)](#); [Jansen \(2000\)](#); [Salavatipour \(2003\)](#); [Bonomo et Valencia-Pabon \(2014\)](#), mais peu de relations ont été établies entre les propriétés structurelles d'un graphe arbitraire  $G$  et sa force.

Trivialement, [Mitchem et Morriss \(1997\)](#) proposèrent de borner la force d'un graphe  $G$  par le degré de ce dernier (Propriété 5).

**Propriété 5** Soit un graphe  $G$  et  $\Delta(G)$  son degré.

$$s(G) \leq \Delta(G) + 1$$

Une borne de meilleure qualité fut proposée dans [Hajiabolhassan et al. \(2000\)](#) (Propriété 6). Il propose de considérer la cardinalité d'une coloration valide afin de borner la force.

**Propriété 6** Soit  $G$  un graphe, et  $X$  une coloration avec  $k$  couleurs.

$$s(G) \leq \left\lceil \frac{\Delta(G) + \chi(G)}{2} \right\rceil \leq \left\lceil \frac{\Delta(G) + k}{2} \right\rceil$$

D'autres travaux ont permis de borner la force pour certaines familles de graphes. Les Propriétés 7 et 8 respectivement proposées dans Kubicka et Schwenk (1989) et dans Mitchem et Morris (1997) introduisent une borne de la force d'un graphe pour la famille des arbres.

**Propriété 7** Soit  $T$  un arbre de  $n$  sommets.  $s(T)$  est borné par  $\Omega(\log(n))$ .

**Propriété 8** Soit  $T$  un arbre et  $\rho$  la cardinalité de l'ensemble des sommets du chemin maximum de  $T$ .

$$s(T) \leq 1 + \left\lceil \frac{\rho}{2} \right\rceil$$

Les Propriétés 9 et 10 respectivement proposées dans Jansen (1997) et dans Nicoloso (2004) introduisent une borne de la force d'un graphe pour la famille des  $k$ -arbres partiels.

**Propriété 9** Soit un  $k$ -arbre partiel  $G$  de  $n$  sommets.  $s(G)$  est borné par  $\Omega(k \times \log(n))$ .

**Propriété 10** Soit un graphe d'intervalles  $G$ .

$$s(G) \leq 2 \times \chi(G) - 1$$

La Propriété 11 proposée dans Halldórsson *et al.* (2001); Halldórsson et Kortsarz (2004) introduit une borne supérieure du nombre de couleurs d'une solution optimale pour le problème de la somme multi-coloration de la famille des graphes planaires.

**Propriété 11** Soit  $G = (V, E)$  un graphe planaire,  $F(v)$  le nombre de couleurs nécessaire au sommet  $v$  et  $F_{\max} = \max_{v \in V} |F(v)|$ . Le nombre maximal de couleurs nécessaire à une multi-coloration de  $G$  minimisant la somme des poids des couleurs utilisées est borné par  $O(F_{\max} \times \chi(G) \times \log(n))$ .

De ce fait, *MSCP* étant un cas particulier de multi-coloration tel que  $F_{\max} = 1$ , la Propriété 11 borne la force d'un graphe planaire en  $O(\chi(G) \times \log(n))$ .

### 3.2.2 Principe général

Le principe de construction des bornes  $UB_A$  et de  $UB_S$  est le suivant : étant donné un motif  $p$  associé à une coloration valide  $X$  d'un graphe  $G = (V, E)$ , déterminer le nombre de couleurs  $k_{\max}$ , tel que  $k_{\max} \in \{1, \dots, |V|\}$ , pour lequel tout motif  $q$  de cardinalité supérieure à  $k_{\max}$  est dominé par  $p$ , comme spécifié dans l'Équation 3.3.

$$\forall q \in \mathcal{U}_{k_{\max}} = \bigcup_{x=k_t}^n \phi(n, x), p \succeq q \quad (3.3)$$

Nous remarquons que l'ensemble  $\mathcal{U}_{k_{\max}}$  n'est jamais vide. Dans le pire des cas, si la cardinalité de la coloration initiale est égale à  $|V|$  alors  $\mathcal{U}_{k_{\max}}$  est constitué de l'unique motif  $(1, 1, \dots, 1)$  de  $\phi(n, n)$ . En effet, le motif se domine lui-même. Par conséquent, du fait que toute coloration de  $\mathcal{U}_{k_{\max}}$  susceptible d'être valide soit dominée par  $p$ , d'après la Propriété 2, aucune de ces colorations ne peut donc être solution de *MSCP*. Donc,  $k_{\max}$  est une borne supérieure de la force du graphe.

### 3.2.3 $UB_A$ une borne algébrique

$UB_A$  est une borne algébrique reposant sur un ensemble de propriétés. En effet, comme expliqué précédemment, il est nécessaire de garantir la dominance d'un motif  $p$  sur l'ensemble des motifs qui le succède dans  $\phi(n)$  afin d'établir  $UB_A$ .

**Propriété 12** Soit un ensemble  $\phi(n, k) \in \phi(n)$ . Tout motif  $q$  de  $\phi(n, k)$  est dominé par le motif  $p_k^1$ .

**Preuve 2** Soit  $q$  un motif de  $\phi(n, k)$ . D'après la Propriété 3,  $p_k^1 = (n - k + 1, \overbrace{1, 1, \dots, 1}^{k-1 \text{ fois}})$ . Ainsi,  $\forall t$  tel que  $1 \leq t \leq k$ ,  $\sum_{x=t+1}^k p_k^1[x] \leq \sum_{x=t+1}^k q[x]$  car  $p_k^1[x] = 1 \leq q[x]$  quand  $x > 1$ . Par conséquent,  $\sum_{x=1}^t p_k^1[x] = n - \sum_{x=t+1}^k p_k^1[x] \geq n - \sum_{x=t+1}^k q[x] = \sum_{x=1}^t q[x]$ .  $\square$

La Propriété 12 établit que le motif  $p_k^1$  domine l'ensemble des motifs qui le succède dans  $\phi(n, k)$  comme montré dans l'Exemple 33.

**Exemple 33** Soit l'ensemble de motifs  $\phi(8)$  de la Table 3.2. Considérons l'ensemble  $\phi(8, 5)$ . D'après la Propriété 12, le motif  $(4, 1, 1, 1, 1)$  domine les motifs  $(3, 2, 1, 1, 1)$  et  $(2, 2, 2, 1, 1)$ .

**Propriété 13** Soit un ensemble  $\phi(n, k) \in \phi(n)$ . Tout motif  $p_{k'}^1$  de  $\bigcup_{x=k}^n \phi(n, x)$  est dominé par le motif  $p_k^1$ .

**Preuve 3** Soit  $p_k^1$  un motif de  $\phi(n)$ .  $\forall p_{k'}^1 \in \phi(n)$  tel que  $k < k'$ , d'après la Propriété 3, nous avons  $p_k^1[1] = n - k + 1$  et  $p_{k'}^1[1] = n - k' + 1$ . Puisque  $k < k'$ ,  $n - k + 1 > n - k' + 1$ . De plus,  $\forall x \in \{2, \dots, k\}$ ,  $p_k^1[x] = p_{k'}^1[x] = 1$ . Donc  $\forall t$  tel que  $1 \leq t \leq k$ ,  $\sum_{x=1}^t p_k^1[x] \geq \sum_{x=1}^t p_{k'}^1[x]$ .  $\square$

La Propriété 13 établit que le motif  $p_k^1$  domine l'ensemble des motifs  $p_{k'}^1$ , tel que  $k' > k$ , qui le succède dans  $\phi(n)$  comme montré dans l'Exemple 34.

**Exemple 34** Soit l'ensemble de motifs  $\phi(8)$  de la Table 3.2. Considérons l'ensemble  $\phi(8, 5)$ . D'après la Propriété 13, le motif  $(4, 1, 1, 1, 1)$  domine le motif  $(3, 1, 1, 1, 1)$ , le motif  $(2, 1, 1, 1, 1)$  et le motif  $(1, 1, 1, 1, 1)$ .

Ainsi, la propriété suivante est une conséquence directe des Propriétés 12 et 13

**Propriété 14** Soit un ensemble  $\phi(n)$ . Tout motif  $q$  de  $\bigcup_{x=k}^n \phi(n, x)$  est dominé par le motif  $p_k^1$ .

Par conséquent le motif  $p_k^1$  domine l'ensemble des motifs  $p_{k'}^i$ , qui le succède dans  $\phi(n)$ .

**Exemple 35** Soit l'ensemble de motifs  $\phi(8)$  de la Table 3.2. Considérons l'ensemble  $\phi(8, 5)$ . D'après la Propriété 14, le motif  $(4, 1, 1, 1, 1)$  domine les motifs  $(3, 2, 1, 1, 1)$ ,  $(2, 2, 2, 1, 1)$ ,  $(3, 1, 1, 1, 1)$ ,  $(2, 2, 1, 1, 1)$ ,  $(2, 1, 1, 1, 1)$  et  $(1, 1, 1, 1, 1)$ .

Ainsi, notre borne algébrique  $UB_A$  est définie par le théorème suivant.

**Théorème 3** Soit  $X$  une coloration valide. Pour tout graphe  $G$ .

$$s(G) \leq \max\left(\left\lceil \frac{1 + \sqrt{1 + 8 \times (\Sigma(X) - n)}}{2} \right\rceil - 1, |X|\right)$$

**Preuve 4** Soit  $X$  une coloration valide de  $G = (V, E)$  et  $p_k^1$  le motif de  $\phi(|V|)$  tel que  $k$  est le plus petit nombre de couleurs pour lequel :

$$\Sigma(p_k^1) \geq \Sigma(X) \quad (3.4)$$

En effet, d'après la Propriété 14, le motif  $p_k^1$  dominant l'ensemble des motifs qui le succède dans  $\phi(n)$ , il ne peut donc y avoir de coloration  $X'$  solution pour MSCP tel que  $|X'| \geq k$ .

D'après la Propriété 3, nous en déduisons que :

$$\Sigma(p_k^1) = (n - k + 1) + \sum_{x=2}^k x$$

Soit encore :

$$\Sigma(p_k^1) = \frac{1}{2}k^2 - \frac{1}{2}k + n \quad (3.5)$$

Ainsi, en considérant l'Équation 3.5, l'Équation 3.4 devient :

$$\frac{1}{2}k^2 - \frac{1}{2}k + n - \Sigma(X) \geq 0 \quad (3.6)$$

L'Équation 3.6 est valide si :

$$k \geq \frac{1 + \sqrt{1 + 8 \times (\Sigma(X) - n)}}{2} \quad (3.7)$$

Par conséquent :

$$s(G) \leq \max\left(\left\lceil \frac{1 + \sqrt{1 + 8 \times (\Sigma(X) - n)}}{2} \right\rceil - 1, |X|\right) \quad (3.8)$$

□

La principale faiblesse de notre borne  $UB_A$  est qu'elle ne considère aucune information structurale du graphe  $G$  autre que la coloration  $X$ . Afin d'affiner le résultat de cette borne, nous proposons une approche algorithmique qui considère, en plus d'une coloration valide  $X$  du graphe, la cardinalité de son stable maximum.

### 3.2.4 $UB_S$ une borne algorithmique

Pour renforcer l'efficacité de notre approche, nous proposons de considérer la cardinalité du stable maximum d'un graphe. Comme mentionné dans la Section 1.3.1, toute coloration valide d'un graphe ne peut avoir une classe couleur dont la cardinalité est supérieure à celle du stable maximum. Ainsi, notre borne algorithmique  $UB_S$  se base sur la propriété suivante.

**Propriété 15** Soit  $\alpha(G)$  la cardinalité du stable maximum de  $G$ . Tout motif  $p$ , tel que  $p[1] > \alpha(G)$ , ne peut correspondre à une coloration valide de  $G$ .

La Propriété 15 induit le fait que tout motif  $p$  de  $\phi(n)$ , dont le premier entier est supérieur à la cardinalité du stable maximum du graphe, est à exclusion de l'espace de recherche. En effet, il ne peut être associé à une coloration valide du graphe. De plus, un autre ensemble de motifs peut être exclu de  $\phi(n)$  en se basant sur l'observation suivante. Étant données deux classes couleurs  $X_i$  et  $X_j$  d'une coloration  $X$ , tel que  $i < j$ , il est parfois possible de transférer un sommet  $x$  de  $X_j$  à  $X_i$  afin d'obtenir une nouvelle coloration  $X'$ . L'intérêt de cette transformation que nous appelons *left-shifting* est que la somme coloration associée à  $X'$  est clairement inférieure à celle de  $X$  ( $\Sigma(X') < \Sigma(X)$ ). Cependant, une opération de left-shifting n'est pas toujours réalisable. En effet, il est nécessaire que la résultante d'une telle opération reste un motif, c'est-à-dire, une suite d'entiers décroissante, telle que  $\forall x < |p|, p[x] > 0$ . Ainsi, un motif de  $\phi(n, k)$  qui ne peut être transformé en un autre motif de  $\phi(n, k)$  par une opération de left-shifting sans incrémenter le premier entier est appelé *motif majeur*.

Intuitivement, un motif majeur de  $\phi(n, k)$  est reconnaissable à sa structuration. En effet, les  $\beta$  premiers entiers d'un motif majeur sont égaux à  $\lambda$ , tel que  $\lceil \frac{n}{k} \rceil \leq \lambda \leq n - k + 1$ . Puis, les  $n - \beta \times \lambda$  sommets restant sont partitionnés en  $k - \beta$  entiers positifs de sorte que le  $\beta + 1$  ème entier est égal à  $\mu$ , avec  $\mu = n - \beta \times \lambda - (k - \beta - 1)$ , et  $\forall i > 1$ , les  $\beta + i$  ème entiers suivant sont égaux à 1. En d'autres termes, un motif majeur est de la forme :  $(\underbrace{\lambda, \dots, \lambda}_{\beta}, \underbrace{\mu, 1, \dots, 1}_{k-\beta})$ . Après exclusion du motif tri-

vial  $(1, 1, \dots, 1)$ , nous remarquons que  $\beta = \lfloor \frac{n-k}{\lambda-1} \rfloor$ . Formellement, nous définissons un motif majeur comme suit :

**Définition 100 Motif majeur**

Soit  $\lambda$  et  $\beta$  deux entiers tels que  $\lceil \frac{n}{k} \rceil \leq \lambda \leq n - k + 1$  et  $\beta = \lfloor \frac{n-k}{\lambda-1} \rfloor$ . Un motif majeur de  $\phi(n, k)$  est un motif  $p_k^i$  dont les propriétés structurelles sont telles que :

- $p_k^i[x] = \lambda$ , si  $1 \leq x \leq \beta$ ;
- $p_k^i[x] = n - \beta \times \lambda - (k - \beta - 1)$ , si  $x = \beta + 1$ ;
- $p_k^i[x] = 1$ , si  $\beta + 1 < x \leq k$ .

**Exemple 36** Si nous nous rapportons à la Table 3.2, les motifs majeurs de  $\phi(8, 4)$  sont  $(5, 1, 1, 1)$ ,  $(4, 2, 1, 1)$ ,  $(3, 3, 1, 1)$ ,  $(2, 2, 2, 2)$ . Pour le motif  $p = (3, 3, 1, 1)$  :

$$\begin{array}{cccc} & \lambda=3 & & \\ \underbrace{\boxed{3} \ \boxed{3}}_{\beta=2} & \boxed{1} & \boxed{1} & \\ & & & \end{array}$$

L'importance de la notion de motif majeur pour MSCP est due aux conséquences qu'impliquent leur structure sur l'ensemble des motifs de  $\phi(n)$ . Ainsi, les Propriétés 16 et 17 introduisent la notion de dominance entre un motif majeur et l'ensemble des motifs qui le succède dans  $\phi(n, k)$ .

**Propriété 16** Si  $p$  un motif majeur de  $\phi(n, k)$  alors  $p$  domine tout motif  $q \in \phi(n, k)$  tel que  $p[1] = q[1]$ .

**Preuve 5** Soit  $\beta = \lfloor \frac{n-k}{p[1]-1} \rfloor$ . Puisque  $p$  est majeur, nous avons du fait de la Définition 100  $p[1] = p[2] = \dots = p[\beta] = q[1] \geq q[2] \geq \dots \geq q[\beta]$ . Ainsi,  $\forall t$  tel que  $1 \leq t \leq \beta$ ,  $\sum_{x=1}^t p[x] \geq \sum_{x=1}^t q[x]$ .

De plus,  $\forall t$  tel que  $\beta + 1 \leq t < k$ , nous avons  $p[t+1] = p[t+2] = \dots = p[k] = 1 \leq q[k] \leq q[k-1] \leq \dots \leq q[t+1]$ . Donc,  $\sum_{x=1}^t p[x] = n - \sum_{x=t+1}^k p[x] \geq n - \sum_{x=t+1}^k q[x] = \sum_{x=1}^t q[x]$ .

Donc,  $p \geq q$ .

La Propriété 16 induit le fait qu'un motif majeur  $p$  de  $\phi(n, k)$  domine le sous-ensemble de motifs  $q$  de  $\phi(n, k)$  tel que  $p[1] = q[1]$ .

**Exemple 37** Soit l'ensemble de motifs  $\phi(8)$  de la Table 3.2. Considérons l'ensemble  $\phi(8, 4)$ . D'après la Propriété 16, le motif  $(3, 3, 1, 1)$  domine le motif  $(3, 2, 2, 1)$ .

**Propriété 17** Si  $p$  un motif majeur de  $\phi(n, k)$  alors  $p$  domine tout motif majeur  $q \in \phi(n, k)$  tel que  $p[1] > q[1]$ .

**Preuve 6** Soit  $\beta_p = \lfloor \frac{n-k}{p[1]-1} \rfloor$  et  $\beta_q = \lfloor \frac{n-k}{q[1]-1} \rfloor$ . Il est clair que  $\beta_p < \beta_q$ . Nous avons  $p[1] = p[2] = \dots = p[\beta_p] > q[1] = q[2] = \dots = q[\beta_p]$ . Par conséquent,  $\forall t$  tel que  $1 \leq t \leq \beta_p$ ,  $\sum_{x=1}^t p[x] > \sum_{x=1}^t q[x]$ .

De plus,  $\forall t$  tel que  $\beta_p + 1 \leq t \leq k$ , nous avons  $p[t+1] = p[t+2] = \dots = p[k] = 1 \leq q[k] \leq q[k-1] \leq \dots \leq q[t+1]$ , implique que  $\sum_{x=1}^t p[x] = n - \sum_{x=t+1}^k p[x] \geq n - \sum_{x=t+1}^k q[x] = \sum_{x=1}^t q[x]$ . Donc,  $p \geq q$ .  $\square$

La Propriété 17 induit le fait qu'un motif majeur de  $\phi(n, k)$  domine le sous-ensemble de motifs majeurs de  $\phi(n, k)$  dont le premier entier est inférieur à celui du motif majeur comme montré dans l'Exemple 38.

**Exemple 38** Soit l'ensemble de motifs  $\phi(8)$  de la Table 3.2. Considérons l'ensemble  $\phi(8, 4)$ . D'après la Propriété 17, le motif  $(4, 2, 1, 1)$  domine les motifs  $(3, 3, 1, 1)$  et  $(2, 2, 2, 2)$ .

Trivialement, les Propriétés 16 et 17 impliquent qu'un motif majeur de  $\phi(n, k)$  domine l'ensemble des motifs qui le succède dans  $\phi(n, k)$  (Propriété 18).

**Propriété 18** Soit  $p_k^i \in \phi(n, k)$  un motif majeur. Si  $p_k^j$  est un motif de  $\phi(n, k)$  tel que  $j > i$  alors  $p_k^i \succeq p_k^j$ .

**Exemple 39** Soit l'ensemble de motifs  $\phi(8)$  de la Table 3.2. Considérons l'ensemble  $\phi(8, 4)$ . D'après la Propriété 18, le motif  $(4, 2, 1, 1)$  domine les motifs  $(3, 3, 1, 1)$ ,  $(3, 2, 2, 1)$  et  $(2, 2, 2, 2)$ .

La propriété suivante établit la relation de dominance entre un motif majeur de  $\phi(n)$  et l'ensemble des motifs qui le succède.

**Propriété 19** Soient  $k$  et  $k'$  deux entiers tels que  $k < k'$  et  $\mathcal{U}_{k'} = \bigcup_{y=k'}^n \{\phi(n, y)\}$ . Si  $p_k^i$  est un motif majeur alors  $\forall q \in \mathcal{U}_{k'}$  tel que  $q[1] \leq p_k^i[1]$ ,  $p_k^i \succeq q$ .

**Preuve 7** Soit  $\beta = \left\lfloor \frac{n-k}{p_k^i[1]-1} \right\rfloor$ . Puisque  $p_k^i$  est majeur, nous avons  $p_k^i[1] = p_k^i[2] = \dots = p_k^i[\beta] \geq q[1] \geq q[2] \geq \dots \geq q[\beta]$ . Par conséquent,  $\forall t$  tel que  $1 \leq t \leq \beta$ ,  $\sum_{x=1}^t p_k^i[x] \geq \sum_{x=1}^t q[x]$ .

De plus,  $\forall t$  tel que  $\beta+1 \leq t \leq k$ , nous avons  $p_k^i[t+1] = p_k^i[t+2] = \dots = p_k^i[k] = 1 \leq q[k] \leq q[k-1] \leq \dots \leq q[t+1]$  ceci implique  $\sum_{x=1}^t p_k^i[x] = n - \sum_{x=t+1}^k p_k^i[x] \geq n - \sum_{x=t+1}^k q[x] = \sum_{x=1}^t q[x]$ . Donc,  $p_k^i \succeq q$ .  $\square$

**Exemple 40** Soit l'ensemble de motifs  $\phi(8)$  de la Table 3.2. Considérons l'ensemble  $\phi(8, 4)$ . D'après la Propriété 19, le motif  $(2, 2, 2, 2)$  domine les motifs  $(2, 2, 2, 1, 1)$ ,  $(2, 2, 1, 1, 1, 1)$ ,  $(2, 1, 1, 1, 1, 1, 1)$  et  $(1, 1, 1, 1, 1, 1, 1, 1)$ .

La Propriété 19 est fondamentale pour la construction de  $UB_S$ . En effet, étant donnée une coloration  $X$  d'un graphe  $G$ , si nous déterminons le plus petit nombre de couleurs pour lequel il existe un motif majeur  $p_k^i$ , tel que son premier entier est égal à la cardinalité du stable maximum du graphe ( $p_k^i[1] = \alpha(G)$ ) et dont la somme coloration est supérieure à celle de la coloration  $X$  ( $\Sigma(X) < \Sigma(p_k^i)$ ), alors la Propriété 19 garantit le fait qu'il n'existe pas de meilleur coloration utilisant plus de  $k$  couleurs. Par conséquence, la force du graphe  $G$  est donc bornée supérieurement par  $k-1$  couleurs.

L'Algorithme 10,  $UB_S(n, \alpha, X)$ , calcule suivant ce principe une borne supérieure de la force d'un graphe de  $n$  sommets, dont le stable maximum (ou une borne supérieure) est  $\alpha^*$  et pour lequel nous disposons d'une coloration valide  $X$ . La fonction  $constructMajorMotif(\lambda, k)$  construit un motif majeur  $p \in \phi(n, k)$ , tel que  $p[1] = \lambda$ , comme précisé dans la Définition 100. La fonction  $computeSumColoring(p)$  calcule la somme coloration associée au motif  $p$  (Équation 3.1). La complexité de ces deux fonctions est en  $O(k)$ . Puisque  $k \leq n$ , la complexité de l'Algorithme 10 est donc en  $O(n^2)$ .

**Exemple 41** Soit le graphe *queen6\_6* où  $n = 36$ ,  $\alpha = 6$  et la coloration  $X$  tel que  $\Sigma(X) = 138$  et  $|X| = 7$ . Par conséquent, nous avons  $UB_S(36, 6, X)$ .

- pour  $k = 7$  : nous avons  $\beta = \left\lfloor \frac{36-7}{6-1} \right\rfloor = 5$ . Donc  $p = (6, 6, 6, 6, 6, 5, 1)$  et  $\Sigma(p) = 127$  ;
- pour  $k = 8$  : nous avons  $\beta = \left\lfloor \frac{36-8}{6-1} \right\rfloor = 5$ . Donc  $p = (6, 6, 6, 6, 6, 4, 1, 1)$  et  $\Sigma(p) = 129$  ;
- pour  $k = 9$  : nous avons  $\beta = \left\lfloor \frac{36-9}{6-1} \right\rfloor = 5$ . Donc  $p = (6, 6, 6, 6, 6, 3, 1, 1, 1)$  et  $\Sigma(p) = 132$  ;
- pour  $k = 10$  : nous avons  $\beta = \left\lfloor \frac{36-10}{6-1} \right\rfloor = 5$ . Donc  $p = (6, 6, 6, 6, 6, 2, 1, 1, 1, 1)$  et  $\Sigma(p) = 136$  ;
- pour  $k = 11$  : nous avons  $\beta = \left\lfloor \frac{36-11}{6-1} \right\rfloor = 5$ . Donc  $p = (6, 6, 6, 6, 6, 1, 1, 1, 1, 1, 1)$  et  $\Sigma(p) = 141$ .

Ainsi, il n'est pas nécessaire de considérer plus de 10 couleurs.

---

**Algorithme 10 :**  $UB_S(n, \alpha^*, X)$

---

**Entrées :**  $n$  le nombre de sommets d'un graphe  $G$ ,  $\alpha^*$  la cardinalité (ou une borne supérieure) du stable maximum de  $G$  et  $X$  une coloration valide de  $G$

**Sortie :** Une borne supérieure de  $s(G)$

```
1 début
2    $SUM \leftarrow 0$ ;
3    $k \leftarrow |X|$ ; /*initialise  $k^*$ */
4    $\lambda \leftarrow \alpha^*$ ;
5   tant que  $SUM \leq \Sigma(X)$  faire
6      $k \leftarrow k + 1$ ;
7     si  $\lambda > n - k + 1$  alors
8        $\lambda \leftarrow n - k + 1$ ;
9      $p \leftarrow \text{constructMajorMotif}(\lambda, k)$ ;
10     $SUM \leftarrow \text{computeSumColoring}(p)$ ;
11  retourner  $k - 1$ ;
```

---

Extraire un stable maximum d'un graphe  $G$  revient à déterminer une clique maximum du graphe complémentaire  $\overline{G}$  de  $G$ . Cependant, nous remarquons que si l'extraction d'une clique maximum d'un graphe est *NP*-difficile, il est plus facile de résoudre ce problème que de déterminer la somme chromatique d'un graphe  $G$  en pratique. En effet, l'extraction d'une clique maximum d'un graphe aléatoire de densité 0,5 tel que  $|V| = 200$  ne prend que quelques secondes (Li *et al.* (2013)). Alors qu'aucun algorithme exact n'est capable de trouver la somme chromatique pour un tel graphe en un temps raisonnable.

### 3.3 Expérimentations de $UB_A$ et de $UB_S$

Afin d'évaluer notre borne supérieure algébrique  $UB_A$  et notre borne algorithmique  $UB_S$ , nous les comparons à celle proposée dans [Hajiabolhassan et al. \(2000\)](#) (Propriété 6). Pour déterminer la cardinalité d'un stable maximum de  $G$ , nous extrayons la clique maximum du graphe complémentaire  $\bar{G}$  à l'aide du solveur IncMaxCLQ proposé dans [Li et al. \(2013\)](#). Bien que le problème de stable maximum soit  $NP$ -difficile, nous remarquons que pour un grand nombre de graphes, la solution optimale est obtenue instantanément. Néanmoins, quand il n'est pas possible de le déterminer en un temps raisonnable, une borne supérieure de  $\alpha(G)$  est utilisée. Nous avons mené nos expérimentations sur un processeur Intel Westmere Xeon E7-8837 de 2.66GHz sur les célèbres benchmarks *COLOR* ([COLOR](#)) et *DIMACS* ([Dimacs](#)).

Le tableau 3.3 montre les résultats expérimentaux pour  $UB_A$  et  $UB_S$ . Pour chaque graphe  $G$ , sont représentés dans les différentes colonnes, le nombre de sommets  $|V|$ , la densité du graphe  $d$ ,  $k^*$  représente le nombre chromatique de  $G$  si celui-ci est connu ou une borne supérieure entre parenthèses sinon [Malaguti et al. \(2011\)](#); [Gualandi et Malucelli \(2012\)](#); [Jin et al. \(2014\)](#), la meilleure borne connue de la somme chromatique  $\Sigma^*$  [Jin et al. \(2014\)](#); [Jin et Hao \(2016\)](#), le degré du graphe  $\Delta(G)$ ,  $\alpha^*$  représente la cardinalité du stable maximum de  $G$  si celle-ci est connue ou une borne supérieure entre parenthèses sinon,  $Time_s$  est le temps en secondes pour déterminer  $\alpha^*$ ,  $UB_{hmt}$  le résultat de la borne de Hajiabolhassan (Propriété 6),  $UB_A$  le résultat de notre borne algébrique (Théorème 3) et  $UB_S$  (Algorithme 10) le résultat de notre borne algorithmique. Nous remarquons que le temps nécessaire à l'élaboration de la borne  $UB_S$  est négligeable. En effet, dans l'ensemble des instances testées,  $UB_S$  est obtenue immédiatement une fois  $\alpha^*$  connue.

| Graphes    | $ V $ | $d$   | $k^*$ | $\Sigma^*$ | $\Delta(G)$ | $\alpha^*$ | $Time_s$ | $UB_{hmt}$ | $UB_A$ | $UB_S$     |
|------------|-------|-------|-------|------------|-------------|------------|----------|------------|--------|------------|
| DSJC125.1  | 125   | 0.094 | 5     | 326        | 23          | 34         | 0        | 14         | 21     | <b>11</b>  |
| DSJC125.5  | 125   | 0.502 | 17    | 1012       | 75          | 10         | 0        | 46         | 43     | <b>29</b>  |
| DSJC125.9  | 125   | 0.898 | 44    | 2503       | 120         | 4          | 0        | 82         | 69     | <b>58</b>  |
| DSJC250.1  | 250   | 0.103 | 8     | 970        | 38          | 44         | 117      | 23         | 38     | <b>21</b>  |
| DSJC250.5  | 250   | 0.503 | (28)  | 3210       | 147         | 12         | 0        | 88         | 77     | <b>50</b>  |
| DSJC250.9  | 250   | 0.896 | 72    | 8277       | 234         | 5          | 24       | 153        | 127    | <b>105</b> |
| DSJC500.1  | 500   | 0.099 | (12)  | 2836       | 68          | (85)       | 1039     | <b>40</b>  | 69     | 52         |
| DSJC500.5  | 500   | 0.501 | (48)  | 10886      | 286         | 13         | 16       | 167        | 145    | <b>81</b>  |
| DSJC500.9  | 500   | 0.901 | (126) | 29862      | 471         | 5          | 84       | 299        | 243    | <b>185</b> |
| DSJC1000.1 | 1000  | 0.099 | (20)  | 8991       | 127         | (175)      | 1540     | <b>74</b>  | 127    | 111        |
| DSJC1000.5 | 1000  | 0.500 | (83)  | 37575      | 551         | 15         | 408      | 317        | 271    | <b>150</b> |
| DSJC1000.9 | 1000  | 0.899 | (222) | 103445     | 924         | 6          | 223      | 573        | 453    | <b>347</b> |
| le450_5a   | 450   | 0.056 | 5     | 1350       | 42          | (94)       | 2430     | 24         | 43     | <b>13</b>  |
| le450_5b   | 450   | 0.056 | 5     | 1350       | 42          | (96)       | 1083     | 24         | 43     | <b>15</b>  |
| le450_5c   | 450   | 0.097 | 5     | 1350       | 66          | 90         | 2        | 36         | 43     | <b>5</b>   |
| le450_5d   | 450   | 0.096 | 5     | 1350       | 68          | 90         | 2        | 37         | 43     | <b>5</b>   |
| le450_15a  | 450   | 0.080 | 15    | 2632       | 99          | 75         | 45       | 57         | 67     | <b>51</b>  |
| le450_15b  | 450   | 0.080 | 15    | 2632       | 94          | 78         | 7        | 55         | 67     | <b>52</b>  |
| le450_15c  | 450   | 0.165 | 15    | 3487       | 139         | 41         | 1602     | 77         | 78     | <b>50</b>  |
| le450_15d  | 450   | 0.165 | 15    | 3505       | 138         | 41         | 2266     | 77         | 79     | <b>50</b>  |
| le450_25a  | 450   | 0.081 | 25    | 3153       | 128         | 91         | 0        | 77         | 74     | <b>64</b>  |
| le450_25b  | 450   | 0.081 | 25    | 3365       | 111         | 78         | 0        | 68         | 77     | <b>66</b>  |
| le450_25c  | 450   | 0.171 | 25    | 4515       | 179         | 47         | 24       | 102        | 91     | <b>74</b>  |
| le450_25d  | 450   | 0.172 | 25    | 4544       | 157         | 43         | 82       | 91         | 91     | <b>72</b>  |
| queen5_5   | 25    | 0.533 | 5     | 75         | 16          | 5          | 0        | 11         | 11     | <b>5</b>   |
| queen6_6   | 36    | 0.460 | 7     | 138        | 19          | 6          | 0        | 13         | 15     | <b>10</b>  |
| queen7_7   | 49    | 0.404 | 7     | 196        | 24          | 7          | 0        | 16         | 18     | <b>7</b>   |

Continue sur la page suivante...

| Graphes       | $ V $ | $d$   | $k^*$ | $\Sigma^*$ | $\Delta(G)$ | $\alpha^*$ | Time <sub>s</sub> | $UB_{hmt}$ | $UB_A$ | $UB_S$     |
|---------------|-------|-------|-------|------------|-------------|------------|-------------------|------------|--------|------------|
| queen8_8      | 64    | 0.361 | 9     | 291        | 27          | 8          | 0                 | 18         | 22     | <b>10</b>  |
| queen8_12     | 96    | 0.300 | 12    | 624        | 32          | 8          | 0                 | 22         | 33     | <b>12</b>  |
| queen9_9      | 81    | 0.325 | 10    | 409        | 32          | 9          | 0                 | 21         | 26     | <b>11</b>  |
| queen10_10    | 100   | 0.296 | 11    | 553        | 35          | 10         | 0                 | 23         | 31     | <b>12</b>  |
| queen11_11    | 121   | 0.272 | 11    | 733        | 40          | 11         | 0                 | 26         | 35     | <b>14</b>  |
| queen12_12    | 144   | 0.252 | 12    | 943        | 43          | 12         | 0                 | 28         | 40     | <b>15</b>  |
| queen13_13    | 169   | 0.234 | 13    | 1191       | 48          | 13         | 0                 | 31         | 46     | <b>16</b>  |
| queen14_14    | 196   | 0.219 | 14    | 1482       | 51          | 14         | 0                 | 33         | 51     | <b>18</b>  |
| queen15_15    | 225   | 0.205 | 15    | 1814       | 56          | 15         | 0                 | 36         | 57     | <b>19</b>  |
| queen16_16    | 256   | 0.193 | 16    | 2193       | 59          | 16         | 0                 | 38         | 63     | <b>21</b>  |
| myciel3       | 11    | 0.363 | 4     | 21         | 5           | 5          | 0                 | 5          | 5      | <b>4</b>   |
| myciel4       | 23    | 0.280 | 5     | 45         | 11          | 11         | 0                 | 8          | 7      | <b>6</b>   |
| myciel5       | 47    | 0.218 | 6     | 93         | 23          | 23         | 0                 | 15         | 10     | <b>8</b>   |
| myciel6       | 95    | 0.169 | 7     | 189        | 47          | 47         | 0                 | 27         | 14     | <b>11</b>  |
| myciel7       | 191   | 0.130 | 8     | 381        | 95          | 95         | 0                 | 52         | 20     | <b>15</b>  |
| fpsol2_i_1    | 496   | 0.094 | 65    | 3403       | 252         | 307        | 0                 | 159        | 77     | <b>75</b>  |
| fpsol2_i_2    | 451   | 0.085 | 30    | 1668       | 346         | 261        | 0                 | 188        | 50     | <b>46</b>  |
| fpsol2_i_3    | 425   | 0.096 | 30    | 1636       | 346         | 238        | 0                 | 188        | 50     | <b>46</b>  |
| inithx.i.1    | 864   | 0.050 | 54    | 3676       | 502         | 566        | 0                 | 278        | 75     | <b>72</b>  |
| inithx.i.2    | 645   | 0.067 | 31    | 2050       | 541         | 365        | 0                 | 286        | 54     | <b>48</b>  |
| inithx.i.3    | 621   | 0.072 | 31    | 1986       | 542         | 360        | 0                 | 287        | 53     | <b>48</b>  |
| mulsol.i.1    | 197   | 0.203 | 49    | 1957       | 121         | 100        | 0                 | 85         | 60     | <b>59</b>  |
| mulsol.i.2    | 188   | 0.221 | 31    | 1191       | 156         | 90         | 0                 | 94         | 45     | <b>44</b>  |
| mulsol.i.3    | 184   | 0.232 | 31    | 1187       | 157         | 86         | 0                 | 94         | 45     | <b>44</b>  |
| mulsol.i.4    | 185   | 0.231 | 31    | 1189       | 158         | 86         | 0                 | 95         | 45     | <b>44</b>  |
| mulsol.i.5    | 186   | 0.230 | 31    | 1160       | 159         | 88         | 0                 | 95         | 45     | <b>43</b>  |
| school1       | 385   | 0.258 | 14    | 2674       | 282         | 41         | 2                 | 148        | 68     | <b>45</b>  |
| school1-nsh   | 352   | 0.236 | 14    | 2392       | 232         | 39         | 0                 | 123        | 64     | <b>43</b>  |
| zeroin.i.1    | 211   | 0.185 | 49    | 1822       | 111         | 120        | 0                 | 80         | 57     | <b>56</b>  |
| zeroin.i.2    | 211   | 0.159 | 30    | 1004       | 140         | 127        | 0                 | 85         | 40     | <b>39</b>  |
| zeroin.i.3    | 206   | 0.167 | 30    | 998        | 140         | 123        | 0                 | 85         | 40     | <b>39</b>  |
| anna          | 138   | 0.052 | 11    | 276        | 71          | 80         | 0                 | 41         | 17     | <b>14</b>  |
| david         | 87    | 0.108 | 11    | 237        | 82          | 36         | 0                 | 47         | 18     | <b>15</b>  |
| huck          | 74    | 0.111 | 11    | 243        | 53          | 27         | 0                 | 32         | 19     | <b>16</b>  |
| jean          | 80    | 0.080 | 10    | 217        | 36          | 38         | 0                 | 23         | 17     | <b>15</b>  |
| games120      | 120   | 0.089 | 9     | 443        | 13          | 22         | 0                 | <b>11</b>  | 26     | 15         |
| flat300-20-0  | 300   | 0.476 | 20    | 3150       | 160         | 15         | 0                 | 90         | 76     | <b>20</b>  |
| flat300-26-0  | 300   | 0.482 | 26    | 3966       | 158         | 12         | 0                 | 92         | 86     | <b>36</b>  |
| flat300-28-0  | 300   | 0.483 | 28    | 4238       | 162         | 12         | 0                 | 95         | 89     | <b>49</b>  |
| flat1000-50-0 | 1000  | 0.490 | 50    | 25500      | 520         | 20         | 263               | 285        | 222    | <b>50</b>  |
| flat1000-60-0 | 1000  | 0.492 | 60    | 30100      | 524         | 17         | 296               | 292        | 242    | <b>77</b>  |
| flat1000-76-0 | 1000  | 0.493 | 76    | 37164      | 532         | 15         | 466               | 304        | 269    | <b>145</b> |
| miles250      | 128   | 0.047 | 8     | 325        | 16          | 44         | 0                 | <b>12</b>  | 20     | 14         |
| miles500      | 128   | 0.143 | 20    | 705        | 38          | 18         | 0                 | 29         | 34     | <b>25</b>  |
| miles750      | 128   | 0.259 | 31    | 1173       | 64          | 12         | 0                 | 48         | 46     | <b>38</b>  |
| miles1000     | 128   | 0.395 | 42    | 1666       | 86          | 8          | 0                 | 64         | 56     | <b>47</b>  |
| miles1500     | 128   | 0.639 | 73    | 3354       | 106         | 5          | 0                 | 90         | 81     | <b>77</b>  |

TABLEAU 3.3 – Comparaison de notre borne algébrique ( $UB_A$ ) et notre borne algorithmique ( $UB_S$ ) face à la borne proposée dans [Hajjabolhassan et al. \(2000\)](#) ( $UB_{hmt}$ )

Les Figures 3.2 et 3.3 représentent les nuages de points relatifs aux comparaisons des scores effectuées entre les bornes  $UB_{hmt}$  et  $UB_A$  d'une part, et les bornes  $UB_{hmt}$  et  $UB_S$  d'autre part.

Sur ces figures, un point représente la borne supérieure de la force obtenue par chacune des deux méthodes pour une instance. L'axe des abscisses est associé au nombre de couleurs (échelle logarithmique) défini par  $UB_{hmt}$  et l'axe des ordonnées est associé au nombre de couleurs (échelle logarithmique) défini par l'une de nos méthodes. Chaque point en dessous de la diagonale signifie que notre approche est meilleure.

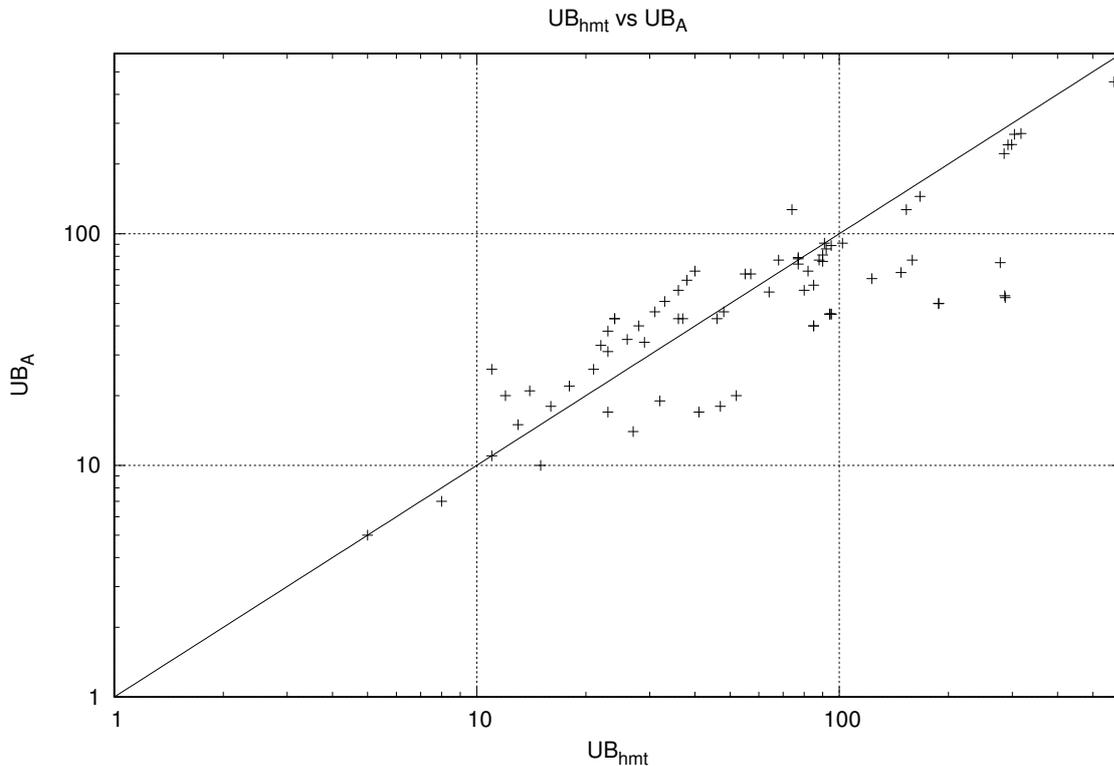


FIGURE 3.2 – Nuage de points comparant les bornes  $UB_{hmt}$  et  $UB_A$ .

Avant de commencer l'analyse de l'impact de nos nouvelles bornes  $UB_A$  et  $UB_S$ , nous remarquons dans le Tableau 3.3 que pour 60 instances sur 74, l'extraction du stable maximum est obtenue en moins d'une minute. Moins de dix minutes suffisent pour obtenir le stable maximum pour 91% des instances.

En ce qui concerne les résultats de  $UB_A$  et  $UB_S$ , le Tableau 3.3 ainsi que les Figures 3.2 et 3.3 montrent que notre approche algébrique  $UB_A$  permet d'améliorer la borne supérieure de la force pour 43 instances sur 74. Malgré le fait que  $UB_A$  ne considère aucune information structurelle du graphe hormis une borne supérieure de la somme chromatique, cette borne algébrique a permis d'améliorer pour 58% des instances la borne supérieure de la force. Pour notre borne  $UB_S$ , nous améliorons considérablement la borne supérieure de la force pour 70 instances sur 74, soit 94% des instances testées. Si nous considérons les graphes de la famille *inithx*, les résultats obtenus par  $UB_{hmt}$  sont respectivement 278, 286 et 287, en ce qui concerne  $UB_S$ , ces derniers sont 72, 48 et 48 ( $UB_A$  75, 54 et 53). Ainsi, nous pouvons donc considérer 200 couleurs de moins pour la résolution de *MSCP* pour ces graphes. De plus, pour 8 instances (le450\_5c, le450\_5d, myciel3, queen5\_5, queen7\_7, queen8\_12, flat300-20-0 et flat1000-50-0)  $UB_S = k^* = \chi(G)$ . De ce fait,  $UB_S$  a permis de prouver l'optimalité de la force pour ces instances. Cependant,  $UB_{hmt}$  donne de meilleurs résultats que  $UB_S$  pour les graphes *DSJC500.1*, *DSJC1000.1*, *games120*, *miles250*. Une des raisons est que la qualité de notre borne supérieure de  $\alpha(G)$  est trop faible sur certaines instances. Néanmoins, au vu des résultats obtenus par notre approche, nous pouvons affirmer l'efficacité de notre

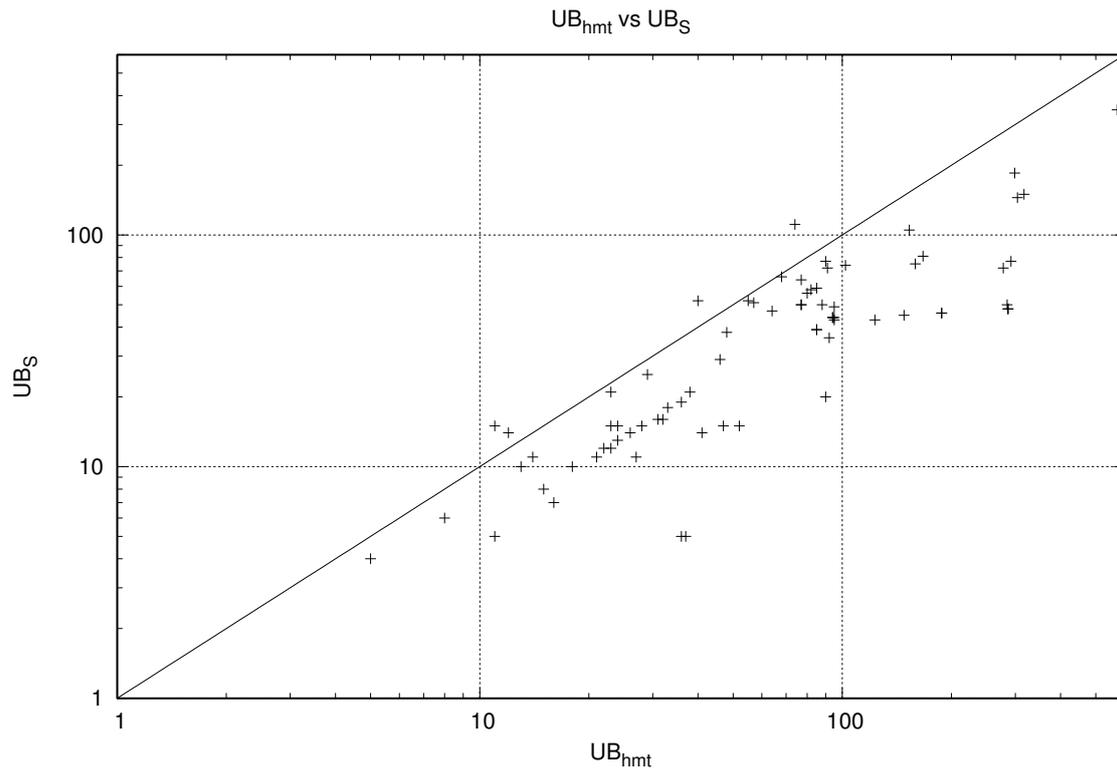


FIGURE 3.3 – Nuage de points comparant les bornes  $UB_{hmt}$  et  $UB_S$ .

borne  $UB_S$  dans la réduction du nombre de couleurs à considérer lors de la résolution de *MSCP*.

### 3.4 LBM $\Sigma$ une nouvelle borne pour la somme coloration

#### 3.4.1 Présentation de LBM $\Sigma$

Les différentes propriétés élaborées autour des motifs ont également amené à l'élaboration d'une borne inférieure algébrique, appelée LBM $\Sigma$ , pour la somme coloration. Avant de présenter LBM $\Sigma$ , il est intéressant de présenter la borne inférieure algébrique proposée dans [Kokosiński et Kwarciány \(2007\)](#). La somme chromatique est bornée comme suit :

**Propriété 20** Soit  $G$  un graphe.  $\Sigma(G) \geq (n - \chi(G)) + \frac{\chi(G) \times (\chi(G) + 1)}{2}$

Les différentes propriétés que nous avons établies sur les motifs nous permettent d'analyser la portée de cette borne (Propriété 20) sur la structure même du graphe. En effet, si nous considérons la Propriété 14, nous savons que le motif  $p_k^1$  domine l'ensemble des motifs qui le succède dans  $\phi(n)$ . Par conséquent, la somme coloration associée à tout motif  $p_k^1$  est une borne inférieure. Si le nombre chromatique ou une borne inférieure du nombre chromatique est connue alors la somme coloration associée au motif  $p_{\chi(G)}^1$  est une borne inférieure de la somme chromatique. La somme coloration du motif  $p_{\chi(G)}^1$  est donnée par l'Équation 3.9.

$$\Sigma(p_{\chi(G)}^1) = (n - \chi(G) + 1) + \sum_{x=2}^{\chi(G)} x \quad (3.9)$$

Soit encore :

$$\Sigma(p_{\chi(G)}^1) = (n - \chi(G)) + \frac{\chi(G) \times (\chi(G) + 1)}{2} \quad (3.10)$$

Ainsi, nous retrouvons la borne proposée par Kokosiński. Cependant, tout comme nous avons pu le constater pour  $UB_A$ , le principal défaut pour cette approche est qu'elle ne considère aucune propriété structurelle du graphe. Nous proposons donc de renforcer cette borne en considérant la cardinalité du stable maximum. En considérant la Propriété 19, un motif majeur  $p$  domine l'ensemble des motifs  $q$  qui le succède dans  $\phi(n)$ , tel que  $p[1] \geq q[1]$ . Par conséquent, la considération du premier motif majeur de cardinalité équivalente au nombre chromatique et dont le premier entier le constituant est équivalent à la valeur de la cardinalité du stable maximum est une borne inférieure de la somme chromatique, que nous appelons LBM $\Sigma$ .

**Exemple 42** Soit  $G = (V, E)$  un graphe tel que  $|V| = 8$  et  $\alpha = 2$ . En considérant l'ensemble de motifs  $\phi(8)$  (Tableau 3.2), nous remarquons que le premier motif pouvant être associé à une coloration valide de  $G$  est le motif majeur  $p = (2, 2, 2, 2)$ . D'après la Propriété 19, le motif  $p$  domine l'ensemble des motifs de  $\phi(8)$  pouvant être associé à une coloration valide du graphe. Par conséquent,  $\Sigma(p) \leq \Sigma(G)$ .

En considérant la structuration d'un motif majeur (Définition 100) et la Propriété 19, notre borne inférieure algébrique LBM $\Sigma$  est définie par le théorème suivant :

**Théorème 4** Soit  $G = (V, E)$  un graphe et  $\beta = \left\lfloor \frac{|V| - \chi(G)}{\alpha(G) - 1} \right\rfloor$ .

$$\begin{aligned} \text{LBM}\Sigma = & \alpha(G) \times \frac{\beta \times (\beta + 1)}{2} + \\ & (|V| - \beta \times \alpha(G) - (\chi(G) - \beta - 1)) \times (\beta + 1) + \\ & \sum_{j=\beta+2}^{\chi(G)} j \end{aligned}$$

La détermination du nombre chromatique et de la cardinalité du stable maximum sont des problèmes NP-difficiles, il n'est pas toujours possible de les déterminer. Dans ce cas, une borne inférieure du nombre chromatique ( $\chi(G)$ ) ou une borne supérieure de la cardinalité du stable maximum ( $\alpha(G)$ ) est utilisée. De telles bornes peuvent être obtenues à l'aide d'algorithmes gloutons.

### 3.4.2 Résultats expérimentaux de $LBM_{\Sigma}$

Afin d'étudier notre borne algébrique  $LBM_{\Sigma}$ , nous comparons les résultats obtenus avec les meilleurs résultats connus pour la borne inférieure de la somme coloration de la littérature. Ces bornes sont basées sur une heuristique décomposant le graphe en une partition de cliques [Wu et Hao \(2013b\)](#); [Moukrim et al. \(2014\)](#), comme illustrée dans la Figure 3.4. La somme chromatique est bornée comme suit :

**Propriété 21** Soit  $G$  un graphe et  $C_1, C_2, \dots, C_k$ , un partitionnement du graphe en  $k$  cliques.

$$\Sigma(G) \geq \sum_{i=1}^k \frac{|C_i| \times (|C_i| + 1)}{2}$$



FIGURE 3.4 – Partitionnement d'un graphe  $G = (V, E)$  en cliques avec  $\Sigma(G) \geq 10$ .

Nous avons mené nos expérimentations sur un processeur Intel Westmere Xeon E7-8837 de 2.66GHz sur les benchmarks *COLOR* ([COLOR](#)) et *DIMACS* ([Dimacs](#)).

Le tableau 3.4 reporte les résultats expérimentaux de  $LBM_{\Sigma}$ . Pour chaque graphe  $G$ , sont représentés dans les différentes colonnes, le nombre de sommets  $|V|$ , la densité du graphe  $d$ , le degré du graphe  $\Delta(G)$ ,  $[k^*]$  le nombre chromatique ou la meilleure borne inférieure connue du nombre chromatique (entre parenthèses) [Gvozdrenović et Laurent \(2008\)](#); [Gualandi et Malucelli \(2012\)](#),  $\alpha^*$  la cardinalité du stable de  $G$  (entre parenthèses quand  $\alpha^*$  n'est pas optimale),  $Time_s$  est le temps en secondes pour déterminer  $\alpha^*$ ,  $\Sigma^*$  la meilleure borne connue de la somme chromatique [Jin et al. \(2014\)](#); [Jin et Hao \(2016\)](#),  $LB_{\Sigma}$  la meilleure borne inférieure connue de la somme chromatique,  $LB_{kok}$  le résultat de la borne de la Propriété 20 ([Kokosiński et Kwarciány \(2007\)](#)),  $LBM_{\Sigma}$  le résultat de notre borne algébrique (Théorème 4).

| Graphes    | $ V $ | $d$   | $\Delta(G)$ | $[k^*]$ | $\alpha^*$ | $Time_s$ | $\Sigma^*$ | $LB_{\Sigma}$ | $LB_{kok}$ | $LBM_{\Sigma}$ |
|------------|-------|-------|-------------|---------|------------|----------|------------|---------------|------------|----------------|
| DSJC125.1  | 125   | 0.094 | 23          | 5       | 34         | 0        | 326        | 247           | 135        | <b>297</b>     |
| DSJC125.5  | 125   | 0.502 | 75          | 17      | 10         | 0        | 1012       | 549           | 261        | <b>855</b>     |
| DSJC125.9  | 125   | 0.898 | 120         | 44      | 4          | 0        | 2503       | 1691          | 1071       | <b>2124</b>    |
| DSJC250.1  | 250   | 0.103 | 38          | 8       | 44         | 117      | 970        | 570           | 278        | <b>843</b>     |
| DSJC250.5  | 250   | 0.503 | 147         | (26)    | 12         | 0        | 3210       | 1287          | 575        | <b>2745</b>    |
| DSJC250.9  | 250   | 0.896 | 234         | 72      | 5          | 24       | 8277       | 4311          | 2806       | <b>6678</b>    |
| DSJC500.1  | 500   | 0.099 | 68          | (7)     | (85)       | 1039     | 2836       | 1250          | 521        | <b>1726</b>    |
| DSJC500.5  | 500   | 0.501 | 286         | (28)    | 13         | 16       | 10886      | 2923          | 878        | <b>9867</b>    |
| DSJC500.9  | 500   | 0.901 | 471         | (123)   | 5          | 84       | 29862      | 11053         | 8003       | <b>25581</b>   |
| DSJC1000.1 | 1000  | 0.099 | 127         | (9)     | (175)      | 1540     | 8991       | 2762          | 1036       | <b>3381</b>    |
| DSJC1000.5 | 1000  | 0.500 | 551         | (32)    | 15         | 408      | 37575      | 6708          | 1496       | <b>33835</b>   |
| DSJC1000.9 | 1000  | 0.899 | 924         | (215)   | 6          | 223      | 103445     | 26557         | 24005      | <b>85235</b>   |
| le450_5a   | 450   | 0.056 | 42          | 5       | (94)       | 2430     | 1350       | 1193          | 460        | <b>1310</b>    |
| le450_5b   | 450   | 0.056 | 42          | 5       | (96)       | 1083     | 1350       | 1189          | 460        | <b>1290</b>    |
| le450_5c   | 450   | 0.097 | 66          | 5       | 90         | 2        | 1350       | 1278          | 460        | <b>1350</b>    |

Continue sur la page suivante...

CHAPITRE 3. NOUVELLES BORNES POUR LE PROBLÈME DE LA SOMME  
COLORATION MINIMUM D'UN GRAPHE

| Graphes      | $ V $ | $d$   | $\Delta(G)$ | $[k^*]$ | $\alpha^*$ | Time <sub>s</sub> | $\Sigma^*$ | $LB_{\Sigma}$ | $LB_{kok}$ | $LBM_{\Sigma}$ |
|--------------|-------|-------|-------------|---------|------------|-------------------|------------|---------------|------------|----------------|
| le450_5d     | 450   | 0.096 | 68          | 5       | 90         | 2                 | 1350       | 1282          | 460        | <b>1350</b>    |
| le450_15a    | 450   | 0.080 | 99          | 15      | 75         | 45                | 2632       | <b>2331</b>   | 555        | 1620           |
| le450_15b    | 450   | 0.080 | 94          | 15      | 78         | 7                 | 2632       | <b>2348</b>   | 555        | 1575           |
| le450_15c    | 450   | 0.165 | 139         | 15      | 41         | 1602              | 3487       | 2610          | 555        | <b>2705</b>    |
| le450_15d    | 450   | 0.165 | 138         | 15      | 41         | 2266              | 3505       | 2628          | 555        | <b>2705</b>    |
| le450_25a    | 450   | 0.081 | 128         | 25      | 91         | 0                 | 3153       | <b>3003</b>   | 750        | 1550           |
| le450_25b    | 450   | 0.081 | 111         | 25      | 78         | 0                 | 3365       | <b>3305</b>   | 750        | 1720           |
| le450_25c    | 450   | 0.171 | 179         | 25      | 47         | 24                | 4515       | <b>3657</b>   | 750        | 2505           |
| le450_25d    | 450   | 0.172 | 157         | 25      | 43         | 82                | 4544       | <b>3698</b>   | 750        | 2690           |
| queen5_5     | 25    | 0.533 | 16          | 5       | 5          | 0                 | 75         | <b>75</b>     | 35         | <b>75</b>      |
| queen6_6     | 36    | 0.460 | 19          | 7       | 6          | 0                 | 138        | 126           | 57         | <b>127</b>     |
| queen7_7     | 49    | 0.404 | 24          | 7       | 7          | 0                 | 196        | <b>196</b>    | 70         | <b>196</b>     |
| queen8_8     | 64    | 0.361 | 27          | 9       | 8          | 0                 | 291        | 288           | 100        | <b>289</b>     |
| queen8_12    | 96    | 0.300 | 32          | 12      | 8          | 0                 | 624        | <b>624</b>    | 162        | <b>624</b>     |
| queen9_9     | 81    | 0.325 | 32          | 10      | 9          | 0                 | 409        | 405           | 126        | <b>406</b>     |
| queen10_10   | 100   | 0.296 | 35          | 11      | 10         | 0                 | 553        | 550           | 155        | <b>551</b>     |
| queen11_11   | 121   | 0.272 | 40          | 11      | 11         | 0                 | 733        | <b>726</b>    | 176        | <b>726</b>     |
| queen12_12   | 144   | 0.252 | 43          | 12      | 12         | 0                 | 943        | <b>936</b>    | 210        | <b>936</b>     |
| queen13_13   | 169   | 0.234 | 48          | 13      | 13         | 0                 | 1191       | <b>1183</b>   | 247        | <b>1183</b>    |
| queen14_14   | 196   | 0.219 | 51          | 14      | 14         | 0                 | 1482       | <b>1470</b>   | 287        | <b>1470</b>    |
| queen15_15   | 225   | 0.205 | 56          | 15      | 15         | 0                 | 1814       | <b>1800</b>   | 330        | <b>1800</b>    |
| queen16_16   | 256   | 0.193 | 59          | 16      | 16         | 0                 | 2193       | <b>2176</b>   | 376        | <b>2176</b>    |
| myciel3      | 11    | 0.363 | 5           | 4       | 5          | 0                 | 21         | 16            | 17         | <b>20</b>      |
| myciel4      | 23    | 0.280 | 11          | 5       | 11         | 0                 | 45         | 34            | 33         | <b>41</b>      |
| myciel5      | 47    | 0.218 | 23          | 6       | 23         | 0                 | 93         | 70            | 62         | <b>81</b>      |
| myciel6      | 95    | 0.169 | 47          | 7       | 47         | 0                 | 189        | 142           | 116        | <b>158</b>     |
| myciel7      | 191   | 0.130 | 95          | 8       | 95         | 0                 | 381        | 286           | 219        | <b>308</b>     |
| fpsol2_i_1   | 496   | 0.094 | 252         | 65      | 307        | 0                 | 3403       | <b>3403</b>   | 2576       | 2701           |
| fpsol2_i_2   | 451   | 0.085 | 346         | 30      | 261        | 0                 | 1668       | <b>1668</b>   | 886        | 1047           |
| fpsol2_i_3   | 425   | 0.096 | 346         | 30      | 238        | 0                 | 1636       | <b>1636</b>   | 860        | 1018           |
| inithx.i.1   | 864   | 0.050 | 502         | 54      | 566        | 0                 | 3676       | <b>3676</b>   | 2295       | 2540           |
| inithx.i.2   | 645   | 0.067 | 541         | 31      | 365        | 0                 | 2050       | <b>2050</b>   | 1110       | 1360           |
| inithx.i.3   | 621   | 0.072 | 542         | 31      | 360        | 0                 | 1986       | <b>1986</b>   | 1086       | 1317           |
| mulsol.i.1   | 197   | 0.203 | 121         | 49      | 100        | 0                 | 1957       | <b>1957</b>   | 1373       | 1422           |
| mulsol.i.2   | 188   | 0.221 | 156         | 31      | 90         | 0                 | 1191       | <b>1191</b>   | 653        | 721            |
| mulsol.i.3   | 184   | 0.232 | 157         | 31      | 86         | 0                 | 1187       | <b>1187</b>   | 649        | 717            |
| mulsol.i.4   | 185   | 0.231 | 158         | 31      | 86         | 0                 | 1189       | <b>1189</b>   | 650        | 719            |
| mulsol.i.5   | 186   | 0.230 | 159         | 31      | 88         | 0                 | 1160       | <b>1160</b>   | 651        | 719            |
| school1      | 385   | 0.258 | 282         | 14      | 41         | 2                 | 2674       | <b>2439</b>   | 476        | 2015           |
| school1-nsh  | 352   | 0.236 | 232         | 14      | 39         | 0                 | 2392       | <b>2176</b>   | 443        | 1779           |
| zeroin.i.1   | 211   | 0.185 | 111         | 49      | 120        | 0                 | 1822       | <b>1822</b>   | 1387       | 1430           |
| zeroin.i.2   | 211   | 0.159 | 140         | 30      | 127        | 0                 | 1004       | <b>1004</b>   | 646        | 701            |
| zeroin.i.3   | 206   | 0.167 | 140         | 30      | 123        | 0                 | 998        | <b>998</b>    | 641        | 695            |
| anna         | 138   | 0.052 | 71          | 11      | 80         | 0                 | 276        | <b>273</b>    | 193        | 241            |
| david        | 87    | 0.108 | 82          | 11      | 36         | 0                 | 237        | <b>234</b>    | 142        | 189            |
| huck         | 74    | 0.111 | 53          | 11      | 27         | 0                 | 243        | <b>243</b>    | 129        | 177            |
| jean         | 80    | 0.080 | 36          | 10      | 38         | 0                 | 217        | <b>216</b>    | 125        | 158            |
| games120     | 120   | 0.089 | 13          | 9       | 22         | 0                 | 443        | <b>442</b>    | 156        | 396            |
| flat300-20-0 | 300   | 0.476 | 160         | 20      | 15         | 0                 | 3150       | 1531          | 490        | <b>3150</b>    |

Continue sur la page suivante...

| Graphes       | $ V $ | $d$   | $\Delta(G)$ | $[k^*]$ | $\alpha^*$ | Time <sub>s</sub> | $\Sigma^*$ | $LB_{\Sigma}$ | $LB_{kok}$ | $LBM_{\Sigma}$ |
|---------------|-------|-------|-------------|---------|------------|-------------------|------------|---------------|------------|----------------|
| flat300-26-0  | 300   | 0.482 | 158         | 26      | 12         | 0                 | 3966       | 1548          | 625        | <b>3901</b>    |
| flat300-28-0  | 300   | 0.483 | 162         | 28      | 12         | 0                 | 4238       | 1547          | 678        | <b>3906</b>    |
| flat1000-50-0 | 1000  | 0.490 | 520         | 50      | 20         | 263               | 25500      | 6601          | 2225       | <b>25500</b>   |
| flat1000-60-0 | 1000  | 0.492 | 524         | 60      | 17         | 296               | 30100      | 6640          | 2770       | <b>29914</b>   |
| flat1000-76-0 | 1000  | 0.493 | 532         | 76      | 15         | 466               | 37164      | 6632          | 3850       | <b>33880</b>   |
| miles250      | 128   | 0.047 | 16          | 8       | 44         | 0                 | 325        | <b>318</b>    | 156        | 267            |
| miles500      | 128   | 0.143 | 38          | 20      | 18         | 0                 | 705        | <b>686</b>    | 318        | 609            |
| miles750      | 128   | 0.259 | 64          | 31      | 12         | 0                 | 1173       | <b>1145</b>   | 593        | 973            |
| miles1000     | 128   | 0.395 | 86          | 42      | 8          | 0                 | 1666       | <b>1623</b>   | 989        | 1475           |
| miles1500     | 128   | 0.639 | 106         | 73      | 5          | 0                 | 3354       | <b>3239</b>   | 2756       | 3107           |

TABLEAU 3.4 – Comparaison de notre borne algébrique ( $LBM_{\Sigma}$ ) avec les meilleures bornes inférieures connues de la littérature pour la somme chromatique.

La Figure 3.5 représente les nuages de points relatifs aux comparaisons de la bornes inférieure de  $MSCP$  entre  $LB_{\Sigma}$  et  $LBM_{\Sigma}$ .

Sur ces figures, un point représente une borne inférieure de la somme chromatique obtenue par  $LB_{\Sigma}$  et  $LBM_{\Sigma}$  pour une instance. L'axe des abscisses est associé à la somme coloration (échelle logarithmique) de  $LB_{\Sigma}$  et l'axes des ordonnées est associé à la somme coloration (échelle logarithmique) obtenue par notre méthode  $LBM_{\Sigma}$ . Chaque point au dessus de la diagonale signifie que notre approche est meilleure.

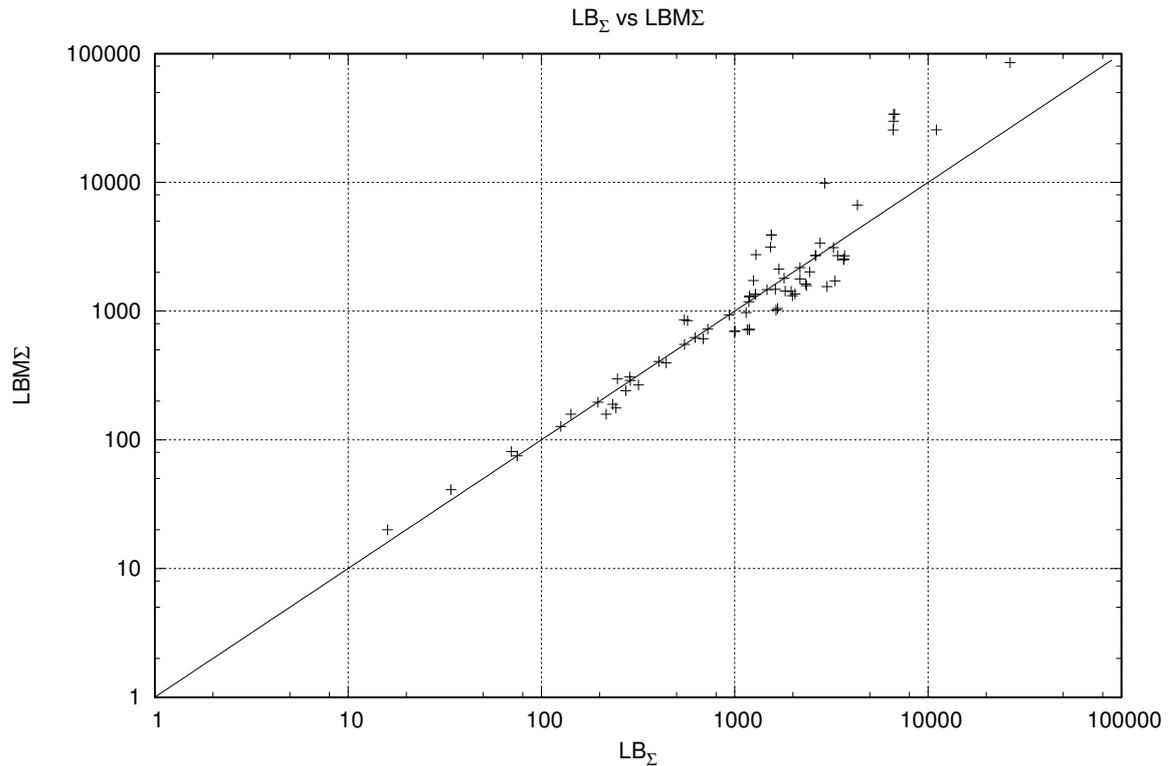


FIGURE 3.5 – Nuage de points comparant les bornes  $LB_{\Sigma}$  et  $LBM_{\Sigma}$ .

Les résultats reportés dans le Tableau 3.4 et synthétisés dans la Figure 3.5 montrent que  $LBM_{\Sigma}$  permet d'améliorer strictement les résultats de la littérature pour 33 instances sur 74. Pour 10 autres instances,  $LBM_{\Sigma}$  atteint la meilleure borne connue. De plus, nous remarquons que  $LBM_{\Sigma}$  permet de prouver l'optimalité de la somme chromatique pour les instances *flat1000-50-0*, *flat300-20-0*, *le450-5c* et *le450-5d*. En effet,  $LBM_{\Sigma}$  est équivalent à la meilleure somme chromatique de

la littérature pour ces graphes. Cependant, pour 31 instances notre approche est moins efficace. L'une des raisons est que dans certains cas, une méthode de partitionnement en cliques peut donner une borne inférieure extrêmement proche de l'optimum. Il est donc très difficile d'améliorer cette dernière. De plus, dans le cas des graphes faiblement denses, une telle approche permet une meilleure considération des propriétés structurelles du graphe.

### 3.5 Conclusions

Dans ce chapitre, après avoir adapté la notion de motif présentée dans [Bonomo et Valencia-Pabon \(2009, 2014\)](#), nous avons introduit deux nouvelles bornes supérieures  $UB_A$  et  $UB_S$  pour la force d'un graphe arbitraire. Ces bornes se basent sur un ensemble de propriétés que nous avons introduit et prouvé dans ce chapitre. L'originalité de cette approche repose sur le fait que  $UB_A$  et  $UB_S$  s'intéressent, à travers la notion de motif, à la structure d'une coloration pouvant être solution pour *MSCP* en faisant abstraction de la validité de cette dernière. En effet, un motif étant une séquence décroissante d'entiers, correspondant aux cardinalités des classes couleurs, il peut donc représenter un ensemble de colorations symétriques. Cependant,  $UB_A$  et  $UB_S$  sont deux méthodes différentes.  $UB_A$  est une borne algébrique exploitant une coloration valide du graphe. La principale faiblesse de cette approche est qu'hormis une coloration valide du graphe,  $UB_A$  n'exploite aucune propriété structurelle de celui-ci.  $UB_S$  est une approche algorithmique en  $O(n^2)$  dérivée de  $UB_A$ . Nous proposons, en plus d'une coloration valide du graphe, de considérer la cardinalité du stable maximum (ou une borne supérieure) du graphe. Cela nous conduit à introduire la notion de motif majeur.

$UB_A$  et  $UB_S$  ont été comparées à la meilleure borne connue pour borner la force d'un graphe, proposée dans [Hajiabolhassan et al. \(2000\)](#), sur les benchmarks *COLOR (COLOR)* et *DIMACS (Dimacs)*. Les résultats expérimentaux ont montré que  $UB_A$  améliore la borne inférieure de la force à l'exception des graphes peu denses. Les résultats sont considérablement améliorés par notre approche.  $UB_S$  permet une réduction substantielle de l'espace de recherche lors de la résolution du *MSCP*.

Notons que la borne  $UB_S$  nécessite de calculer le stable maximum de  $G$  qui est un problème *NP*-difficile comme *MSCP*. Cependant, en pratique déterminer la cardinalité du stable maximum est plus facile à calculer que la force d'un graphe. Pour la majorité des graphes du benchmarks, l'algorithme exact *IncMaxCLQ* ([Li et al. \(2013\)](#)) a pu déterminer le stable maximum en quelques secondes. En effet, une borne supérieure du stable a été utilisée pour seulement 4 graphes sur 74. De plus, pour la première fois dans la littérature, nous avons pu déterminer la valeur exacte de la force pour 8 graphes.

Les motifs ont également permis d'introduire une nouvelle borne inférieure algébrique  $LBM\Sigma$  pour la somme chromatique. Contrairement à l'approche proposée dans [Kokosiński et Kwarciány \(2007\)](#) qui ne considère qu'une coloration valide du graphe,  $LBM\Sigma$  considère également la cardinalité du stable maximum (ou une borne supérieure) du graphe.

$LBM\Sigma$  a été comparée aux meilleures bornes connues de la littérature sur les benchmarks *COLOR (COLOR)* et *DIMACS (Dimacs)*. Ces bornes sont basées sur une heuristique décomposant le graphe en une partition en cliques ([Wu et Hao \(2013b\)](#); [Moukrim et al. \(2014\)](#)). Les résultats ont montré que notre approche améliore significativement la borne inférieure de la somme chromatique pour quelques familles de graphes. De plus,  $LBM\Sigma$  a permis de démontrer pour la première fois dans la littérature l'optimalité de la meilleure borne supérieure connue pour 4 instances.

## Chapitre 4

# Un Nouveau Paradigme de Résolution du Problème de la Somme Coloration Minimum d'un Graphe

### Sommaire

---

|  |           |
|--|-----------|
| <b>4.1 Encodage vers MaxSAT</b> . . . . .                                  | <b>63</b> |
| 4.1.1 Weighted partial MaxSAT . . . . .                                    | 63        |
| 4.1.2 Encodage de MSCP en un problème weighted partial MaxSAT . . . . .    | 65        |
| <b>4.2 Encodage vers MinSAT</b> . . . . .                                  | <b>67</b> |
| 4.2.1 weighted partial MinSAT . . . . .                                    | 67        |
| 4.2.2 Encodage de MSCP en un problème weighted partial MinSAT . . . . .    | 68        |
| <b>4.3 Diminution du nombre de variables</b> . . . . .                     | <b>70</b> |
| 4.3.1 Réduction des domaines . . . . .                                     | 70        |
| 4.3.2 Étude expérimentale sur la réduction des domaines . . . . .          | 71        |
| <b>4.4 Étude des performances des encodages MaxSAT et MinSAT</b> . . . . . | <b>81</b> |
| <b>4.5 Conclusions</b> . . . . .   | <b>85</b> |

---

Dans ce chapitre, nous étudions un nouveau paradigme de recherche pour la résolution de *MSCP* basé sur le formalisme de la logique propositionnelle à travers le problème *weighted partial MaxSAT* et le problème *weighted partial MinSAT*. Pour cela, nous proposons deux codages pour transformer *MSCP* en une instance de type *MaxSAT* et *MinSAT*. Afin de réduire significativement les instances générées par ces approches, nous analysons et renforçons la propriété présentée dans [Minot et al. \(2017\)](#) afin d'exploiter en plus du degré d'un sommet, une borne supérieure de la force. Ensuite, nous comparons les temps de résolution de nos approches.

Ce chapitre est organisé de la façon suivante. Après avoir présenté le problème *weighted partial MaxSAT* et le problème *weighted partial MinSAT* ainsi que les différents encodages de *MSCP* associés, nous étudions différentes approches pour diminuer la taille des instances. Avant de conclure ce chapitre, nous comparons expérimentalement nos approches.

Ces travaux ont donné lieu à plusieurs publications [Lecat et al. \(2015a,b\)](#).

## 4.1 Encodage vers MaxSAT

### 4.1.1 Weighted partial MaxSAT

L'objectif du problème *SAT* est de déterminer une interprétation satisfaisant l'ensemble des clauses d'une formule *CNF*. Cependant, dans certains cas, une formule n'admet pas de solution. De ce fait, il devient intéressant de rechercher une interprétation maximisant le nombre de clauses

satisfaites afin de répondre au mieux à la formule soumise. Ce problème, appelé *problème MaxSAT*, est un problème d'optimisation NP-difficile Papadimitriou (2003) et il se définit formellement comme suit :

**Définition 101 Problème MaxSAT**

Soit  $F = \{c_1, c_2, \dots, c_n\}$  une formule CNF. L'objectif du problème MaxSAT est de déterminer une interprétation  $\mathcal{I}$  telle que :

$$\text{MaxSAT}(F) = \max\{\mathcal{I}(c_i) = 1 \mid c_i \in F\}$$

**Exemple 43** Soit  $F = \{x, y, z, \neg x \vee \neg y, \neg x \vee \neg z\}$  une formule CNF. L'interprétation  $\mathcal{I} = \{\neg x, y, z\}$  est une solution optimale du problème MaxSAT avec  $\text{MaxSAT}(F) = 4$ .

Le problème *partial MaxSAT* est une extension du problème MaxSAT. Un problème *partial MaxSAT* considère une CNF dont les clauses sont réparties en deux sous-ensembles : les clauses *dures* (*hard*s) et les clauses *souples* (*soft*s). L'objectif du problème *partial MaxSAT* est de déterminer une interprétation satisfaisant l'ensemble des clauses dures et maximisant le nombre de clauses souples valides. Formellement, il se définit comme suit :

**Définition 102 Problème partial MaxSAT**

Soit  $F = F_d \cup F_s$  une formule CNF, telle que  $F_d$  est l'ensemble des clauses dures et  $F_s$  l'ensemble des clauses souples. L'objectif du problème *partial MaxSAT* est de déterminer une interprétation  $\mathcal{I}$  telle que :

$$\text{partial-MaxSAT}(F) = \max\{\mathcal{I}(c_i) = 1 \mid c_i \in F_s \text{ et } \forall c_j \in F_d, \mathcal{I}(c_j) = 1\}$$

**Exemple 44** Soit  $F = F_d \cup F_s$  une formule CNF, telle que  $F_d = \{\neg x \vee \neg y, \neg x \vee \neg z\}$  et  $F_s = \{x, y, z\}$ . L'interprétation  $\mathcal{I} = \{\neg x, y, z\}$  est une solution optimale du problème *partial MaxSAT*.

La modélisation du problème MSCP vers un formalisme SAT, nécessite de prendre en considération le poids associé à chaque couleur. Pour cela, un poids est associé à chaque clause souple du problème *partial MaxSAT* afin de leur exprimer une préférence. Le problème visant à déterminer une affectation satisfaisant l'ensemble des clauses dures et maximisant la somme des poids des clauses souples satisfaites est appelé *weighted partial MaxSAT* et se définit formellement comme suit :

**Définition 103 Problème weighted partial MaxSAT**

Soit  $F = F_d \cup F_s$  une formule CNF, telle que  $F_d$  est l'ensemble des clauses dures et  $F_s$  l'ensemble des clauses souples, et une fonction de coût  $w : c \rightarrow \mathbb{N}$  qui associe un poids à chacune des clauses  $c$  de  $F$ . L'objectif du problème *weighted partial MaxSAT* est de déterminer une interprétation  $\mathcal{I}$  telle que :

$$\text{weighted-partial-MaxSAT}(F) = \max\{\sum w(c_i) \mid c_i \in F_s, \mathcal{I}(c_i) = 1 \text{ et } \forall c_j \in F_d, \mathcal{I}(c_j) = 1\}$$

**Remarque 9** Dans ce manuscrit, nous associons à une clause dure  $c$  un poids infini :  $w(c) = \infty$ .

**Exemple 45** Soit  $F = F_d \cup F_s$  une formule CNF, telle que  $F_d = \{(\neg x \vee \neg y, \infty), (\neg x \vee \neg z, \infty), (\neg y \vee \neg z, \infty)\}$  et  $F_s = \{(x, 10), (y, 1), (z, 2)\}$ . L'interprétation  $\mathcal{I} = \{x, \neg y, \neg z\}$  est une solution optimale du problème *weighted partial MaxSAT* avec  $\text{weighted-partial-MaxSAT}(F) = 10$ .

### 4.1.2 Encodage de MSCP en un problème weighted partial MaxSAT

La transposition de *MSCP* en un problème *weighted partial MaxSAT* requiert la définition des variables booléennes constituant la *CNF*. Ainsi, une variable booléennes  $x_{vj}$  est assignée à chaque sommet  $v$  du graphe et à chaque couleur  $j$ . La variable  $x_{vj}$  a pour valeur de vérité 1 quand le sommet  $v$  est coloré avec la couleur  $j$ .

Nous rappelons que pour un sommet  $v$  d'un graphe  $G$ , la fonction  $c(v)$  associe à  $v$  une couleur (Définition 53).

#### Définition 104 $x_{vj}$

Soit  $G$  un graphe et  $v$  un sommet de  $G$ . La variable propositionnelle  $x_{vj}$  est telle que :

$$x_{vj} = \begin{cases} 1 & \text{si } c(v) = j \\ 0 & \text{sinon} \end{cases}$$

Nous définissons l'ensemble des clauses dures composant le problème *weighted partial MaxSAT*, noté  $F_{dMSCP}$ , comme étant l'ensemble des contraintes d'un problème de coloration de graphe. Ainsi, les clauses de l'ensemble  $F_{dMSCP}$  modélisent la contrainte d'affectation d'une couleur à un sommet, la contrainte d'unicité de la couleur d'un sommet et la contrainte de voisinage. Nous rappelons que d'après la Propriété 5, il n'est pas nécessaire de considérer plus de  $\Delta(G) + 1$  couleurs.

#### Définition 105 $F_{dMSCP}$

Soit  $G = (V, E)$ . L'ensemble des clauses  $F_{dMSCP}$  est tel que  $\forall u \in V$  et  $\forall (u, v) \in E$  :

$$F_{dMSCP} = \left\{ \bigvee_{i=1}^{\Delta(G)+1} x_{ui} \right\} \cup \left( \bigcup_{i=1}^{\Delta(G)+1} \bigcup_{j=i}^{\Delta(G)+1} \{ \neg x_{ui} \vee \neg x_{uj} \} \right) \cup \left( \bigcup_{i=1}^{\Delta(G)+1} \{ \neg x_{ui} \vee \neg x_{vi} \} \right)$$

**Remarque 10** 1. L'ensemble des clauses  $\left\{ \bigvee_{i=1}^{\Delta(G)+1} x_{ui} \right\}$  représente le fait que tout sommet doit être coloré;

2. L'ensemble des clauses  $\bigcup_{i=1}^{\Delta(G)+1} \bigcup_{j=i}^{\Delta(G)+1} \{ \neg x_{ui} \vee \neg x_{uj} \}$  représente le fait qu'un sommet ne peut être coloré avec plus d'une couleur;

3. L'ensemble des clauses  $\bigcup_{i=1}^{\Delta(G)+1} \{ \neg x_{ui} \vee \neg x_{vi} \}$  représente le fait que deux sommets voisins ne peuvent être colorés avec une même couleur.

D'après la Définition 105, le nombre de clauses générées est donné par le formule de la Propriété 22.

**Propriété 22** Soit  $G = (V, E)$  un graphe. La cardinalité de  $F_{dMSCP}$  est :

$$|F_{dMSCP}| = |V| + (|V| \times \binom{\Delta(G)+1}{2}) + (|E| \times (\Delta(G)+1))$$

L'ensemble des clauses souples composant le problème *weighted partial MaxSAT*, noté  $F_{sMSCP}$ , est un ensemble de clauses unitaires. Afin de modéliser  $F_{sMSCP}$ , nous proposons deux approches. La première approche, notée encodage 1 (Définition 106), définit une clause soft unitaire  $x_{vj}$  pour chaque sommet  $v$  du graphe et pour chaque couleur  $j$ , tel que le poids associé est  $k + 1 - j$  avec  $k = \Delta(G) + 1$ . L'objectif du problème *weighted partial MaxSAT* étant de maximiser la somme des poids des clauses souples satisfaites, les clauses souples associées aux plus petites couleurs sont par conséquent satisfaites en priorité.

**Définition 106 Encodage 1 de  $F_{sMSCP}$  (weighted partial MaxSAT)**

Soit  $G = (V, E)$ . L'ensemble des clauses  $F_{sMSCP}$  est tel que  $\forall u \in V$  :

$$F_{sMSCP} = \bigcup_{i=1}^{\Delta(G)+1} \{x_{ui}\}$$

où le coût associé à chaque clause unitaire  $x_{ui}$  est  $w(x_{ui}) = \Delta(G) + 2 - i$ .

La seconde approche, notée encodage 2 (Définition 107), définit une clause soft unitaire  $\neg x_{vj}$  pour chaque sommet  $v$  du graphe et pour chaque couleur  $j$ , tel que le poids associé est  $j$ . La résolution de cet encodage satisfait les clauses souples associées aux plus grandes couleurs en priorité.

**Définition 107 Encodage 2 de  $F_{sMSCP}$  (weighted partial MaxSAT)**

Soit  $G = (V, E)$ . L'ensemble des clauses  $F_{sMSCP}$  est tel que  $\forall u \in V$  :

$$F_{sMSCP} = \bigcup_{i=1}^{\Delta(G)+1} \{\neg x_{ui}\}$$

où le coût associé à chaque clause unitaire  $\neg x_{ui}$  est  $w(\neg x_{ui}) = i$ .

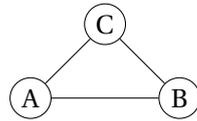


FIGURE 4.1 – Graphe  $G = (V, E)$ .

**Exemple 46** Soit le  $G = (V, E)$  le graphe de la Figure 4.1. L'encodage weighted partial MaxSAT associé est :

$$\begin{array}{l}
 F_{d\text{MSCP}} : \quad \left. \begin{array}{l}
 x_{A1} \vee x_{A2} \vee x_{A3} \\
 x_{B1} \vee x_{B2} \vee x_{B3} \\
 x_{C1} \vee x_{C2} \vee x_{C3}
 \end{array} \right\} \text{ tout sommet doit être coloré} \\
 \\
 \left. \begin{array}{l}
 \neg x_{A1} \vee \neg x_{A2} \\
 \neg x_{A1} \vee \neg x_{A3} \\
 \neg x_{A2} \vee \neg x_{A3} \\
 \neg x_{B1} \vee \neg x_{B2} \\
 \neg x_{B1} \vee \neg x_{B3} \\
 \neg x_{B2} \vee \neg x_{B3} \\
 \neg x_{C1} \vee \neg x_{C2} \\
 \neg x_{C1} \vee \neg x_{C3} \\
 \neg x_{C2} \vee \neg x_{C3}
 \end{array} \right\} \text{ tout sommet ne peut être coloré avec plus d'une couleur} \\
 \\
 \left. \begin{array}{l}
 \neg x_{A1} \vee \neg x_{B1} \\
 \neg x_{A2} \vee \neg x_{B2} \\
 \neg x_{A3} \vee \neg x_{B3} \\
 \neg x_{A1} \vee \neg x_{C1} \\
 \neg x_{A2} \vee \neg x_{C2} \\
 \neg x_{A3} \vee \neg x_{C3} \\
 \neg x_{B1} \vee \neg x_{C1} \\
 \neg x_{B2} \vee \neg x_{C2} \\
 \neg x_{B3} \vee \neg x_{C3}
 \end{array} \right\} \text{ deux sommets voisins ne peuvent pas avoir une même couleur}
 \end{array}$$

Encodage 1 (Définition 106)

$$F_{s\text{MSCP}} : \quad \begin{array}{l}
 x_{A1} \ 3 \\
 x_{A2} \ 2 \\
 x_{A3} \ 1 \\
 x_{B1} \ 3 \\
 x_{B2} \ 2 \\
 x_{B3} \ 1 \\
 x_{C1} \ 3 \\
 x_{C2} \ 2 \\
 x_{C3} \ 1
 \end{array}$$

Encodage 2 (Définition 107)

$$F_{s\text{MSCP}} : \quad \begin{array}{l}
 \neg x_{A1} \ 1 \\
 \neg x_{A2} \ 2 \\
 \neg x_{A3} \ 3 \\
 \neg x_{B1} \ 1 \\
 \neg x_{B2} \ 2 \\
 \neg x_{B3} \ 3 \\
 \neg x_{C1} \ 1 \\
 \neg x_{C2} \ 2 \\
 \neg x_{C3} \ 3
 \end{array}$$

Une solution de l'encodage 1 et 2 est  $\mathcal{I} = \{x_{A1}, \neg x_{A2}, \neg x_{A3}, \neg x_{B1}, x_{B2}, \neg x_{B3}, \neg x_{C1}, \neg x_{C2}, x_{C3}\}$  avec  $\text{weighted-partial-MaxSAT}(F)=6$  pour l'encodage 1 et  $\text{weighted-partial-MaxSAT}(F)=12$  pour l'encodage 2

D'après les Propriétés 106 et 107, les encodages 1 et 2 génèrent le même nombre de clauses.

**Propriété 23** Soit  $G = (V, E)$  un graphe. La cardinalité de  $F_{s\text{MSCP}}$  est :

$$|F_{s\text{MSCP}}| = |V| \times (\Delta(G) + 1)$$

## 4.2 Encodage vers MinSAT

### 4.2.1 weighted partial MinSAT

Le problème *MinSAT* est le problème dual de *MaxSAT*. L'objectif du problème *MinSAT* est de déterminer une interprétation d'une formule *CNF* minimisant le nombre de clauses satisfaites.

En d'autres termes, maximiser le nombre de clauses insatisfaites. L'idée sur laquelle se base le problème *MinSAT* est qu'il est parfois plus facile de déterminer l'insatisfaction d'une instance que la satisfaction de cette dernière. Le problème *MinSAT* a été prouvé *NP*-difficile dans Kohli *et al.* (1994).

**Définition 108 Problème *MinSAT***

Soit  $F = \{c_1, c_2, \dots, c_n\}$  une formule CNF. L'objectif du problème *MinSAT* est de déterminer une interprétation  $\mathcal{I}$  telle que :

$$\text{MinSAT}(F) = \min\{\mathcal{I}(c_i) = 1 \mid c_i \in F\}$$

**Exemple 47** Soit  $F = \{x, y, z, \neg x \vee \neg y, \neg x \vee \neg z\}$  une formule CNF. L'interprétation  $\mathcal{I} = \{\neg x, \neg y, \neg z\}$  est une solution optimale du problème *MinSAT* avec  $\text{MinSAT}(F) = 2$ .

Tout comme le problème *partial MaxSAT*, le problème *partial MinSAT* est une extension du problème *MinSAT* constituée d'un ensemble de clauses dures et de clauses souples. Cependant, à l'instar du problème *partial MaxSAT*, l'objectif du problème *partial MinSAT* est de déterminer une interprétation satisfaisant l'ensemble de clauses dures tout en minimisant (maximisant) l'ensemble de clauses souples satisfaites (insatisfaites).

**Définition 109 Problème *partial MinSAT***

Soit  $F = F_d \cup F_s$  une formule CNF, telle que  $F_d$  est l'ensemble des clauses dures et  $F_s$  l'ensemble des clauses souples. L'objectif du problème *partial MinSAT* est de déterminer une interprétation  $\mathcal{I}$  telle que :

$$\text{partial-MinSAT}(F) = \min\{\mathcal{I}(c_i) = 1 \mid c_i \in F_s \text{ et } \forall c_j \in F_d, \mathcal{I}(c_j) = 1\}$$

**Exemple 48** Soit  $F = F_d \cup F_s$  une formule CNF, telle que  $F_d = \{\neg x \vee \neg y, \neg x \vee \neg z\}$  et  $F_s = \{x, y, z\}$ . L'interprétation  $\mathcal{I} = \{\neg x, \neg y, \neg z\}$  est une solution optimale du problème *partial MinSAT* avec  $\text{partial-MinSAT}(F) = 0$ .

Afin d'encoder le problème *MSCP* vers un formalisme *MinSAT*, il est nécessaire de considérer le problème *weighted partial MinSAT*.

**Définition 110 Problème *weighted partial MinSAT***

Soit  $F = F_d \cup F_s$  une formule CNF, telle que  $F_d$  est l'ensemble des clauses dures et  $F_s$  l'ensemble des clauses souples, et une fonction de coût  $w : c \rightarrow \mathbb{N}$  qui associe un poids à chacune des clauses  $c$  de  $F$ . L'objectif du problème *weighted partial MinSAT* est de déterminer une interprétation  $\mathcal{I}$  telle que :

$$\text{weighted-partial-MinSAT}(F) = \min\{\sum w(c_i) \mid c_i \in F_s, \mathcal{I}(c_i) = 1 \text{ et } \forall c_j \in F_d, \mathcal{I}(c_j) = 1\}$$

**Exemple 49** Soit  $F = F_d \cup F_s$  une formule CNF, telle que  $F_d = \{(\neg x \vee \neg y, \infty), (\neg x \vee \neg z, \infty), (\neg y \vee \neg z, \infty)\}$  et  $F_s = \{(x, 10), (y, 1), (z, 2)\}$ . L'interprétation  $\mathcal{I} = \{\neg x, \neg y, \neg z\}$  est une solution optimale du problème *weighted partial MinSAT* avec  $\text{weighted-partial-MinSAT}(F) = 13$ .

**4.2.2 Encodage de *MSCP* en un problème *weighted partial MinSAT***

Le problème *weighted partial MinSAT* étant le problème dual du problème *weighted partial MaxSAT* problème, l'encodage s'en trouve par conséquent fortement similaire. Ainsi, les variables propositionnelles composant la formule CNF sont identiques à la Définition 104. De plus, le problème *weighted partial MinSAT* doit déterminer une interprétation validant l'ensemble des clauses dures de  $F_{d\text{MSCP}}$ . De ce fait,  $F_{d\text{MSCP}}$  est similaire à l'ensemble proposé dans l'encodage précédent (Définition 105).

L'encodage des clauses souples est dépendant de l'approche selon laquelle nous abordons le problème. Si la priorité est donnée à la plus petite couleur alors un poids plus important lui est attribuée. Cependant, l'objectif du problème *weighted partial MinSAT* étant de maximiser la somme des poids des clauses insatisfaites, chaque clause souple est donc unitaire et constituée de la négation d'une variable propositionnelle.

**Définition 111 Encodage 1 de  $F_{sMSCP}$  (*weighted partial MinSAT*)**

Soit  $G = (V, E)$ . L'ensemble des clauses  $F_{sMSCP}$  est tel que  $\forall u \in V$  :

$$F_{sMSCP} = \bigcup_{i=1}^{\Delta(G)+1} \{\neg x_{ui}\}$$

où le coût associé à chaque clause unitaire  $\neg x_{ui}$  est  $w(\neg x_{ui}) = \Delta(G) + 2 - i$ .

Cependant, tout comme dans la Définition 107, la priorité peut être donnée à une couleur dont nous souhaitons éviter l'affectation. Un faible coût est associé à la clause unitaire correspondant au sommet coloré avec la plus petite couleur. Le coût associé à  $x_{u1}$  est donc plus faible que celui associé à  $x_{u2}$ .

**Définition 112 Encodage 2 de  $F_{sMSCP}$  (*weighted partial MinSAT*)**

Soit  $G = (V, E)$ . L'ensemble des clauses  $F_{sMSCP}$  est tel que  $\forall u \in V$  :

$$F_{sMSCP} = \bigcup_{i=1}^{\Delta(G)+1} \{x_{ui}\}$$

où le coût associé à chaque clause unitaire  $x_{ui}$  est  $w(x_{ui}) = i$ .

**Exemple 50** Soit le  $G = (V, E)$  le graphe de la Figure 4.1. L'encodage *weighted partial MinSAT* associé est :

$$\begin{array}{l}
 F_{d\text{MSCP}} : \quad \left. \begin{array}{l}
 x_{A1} \vee x_{A2} \vee x_{A3} \\
 x_{B1} \vee x_{B2} \vee x_{B3} \\
 x_{C1} \vee x_{C2} \vee x_{C3}
 \end{array} \right\} \text{ tout sommet doit être coloré} \\
 \left. \begin{array}{l}
 \neg x_{A1} \vee \neg x_{A2} \\
 \neg x_{A1} \vee \neg x_{A3} \\
 \neg x_{A2} \vee \neg x_{A3} \\
 \neg x_{B1} \vee \neg x_{B2} \\
 \neg x_{B1} \vee \neg x_{B3} \\
 \neg x_{B2} \vee \neg x_{B3} \\
 \neg x_{C1} \vee \neg x_{C2} \\
 \neg x_{C1} \vee \neg x_{C3} \\
 \neg x_{C2} \vee \neg x_{C3}
 \end{array} \right\} \text{ tout sommet ne peut être coloré avec plus d'une couleur} \\
 \left. \begin{array}{l}
 \neg x_{A1} \vee \neg x_{B1} \\
 \neg x_{A2} \vee \neg x_{B2} \\
 \neg x_{A3} \vee \neg x_{B3} \\
 \neg x_{A1} \vee \neg x_{C1} \\
 \neg x_{A2} \vee \neg x_{C2} \\
 \neg x_{A3} \vee \neg x_{C3} \\
 \neg x_{B1} \vee \neg x_{C1} \\
 \neg x_{B2} \vee \neg x_{C2} \\
 \neg x_{B3} \vee \neg x_{C3}
 \end{array} \right\} \text{ deux sommets voisins ne peuvent pas avoir une même couleur}
 \end{array}$$

Encodage 1 (Définition 111)

$$F_{s\text{MSCP}} : \quad \begin{array}{l}
 \neg x_{A1} \ 3 \\
 \neg x_{A2} \ 2 \\
 \neg x_{A3} \ 1 \\
 \neg x_{B1} \ 3 \\
 \neg x_{B2} \ 2 \\
 \neg x_{B3} \ 1 \\
 \neg x_{C1} \ 3 \\
 \neg x_{C2} \ 2 \\
 \neg x_{C3} \ 1
 \end{array}$$

Encodage 2 (Définition 112)

$$F_{s\text{MSCP}} : \quad \begin{array}{l}
 x_{A1} \ 1 \\
 x_{A2} \ 2 \\
 x_{A3} \ 3 \\
 x_{B1} \ 1 \\
 x_{B2} \ 2 \\
 x_{B3} \ 3 \\
 x_{C1} \ 1 \\
 x_{C2} \ 2 \\
 x_{C3} \ 3
 \end{array}$$

## 4.3 Diminution du nombre de variables

### 4.3.1 Réduction des domaines

Les encodages que nous avons proposés afin de modéliser *MSCP* en problèmes *weighted partial MaxSAT* et *weighted partial MinSAT* ne sont pas optimaux. En effet, si nous nous intéressons à la complexité spatiale de la formule *CNF* générée, nous remarquons que cette dernière est en  $O(n \times k + m \times k + n \times k)$ , où  $n = |V|$ ,  $m = |E|$  et  $k = \Delta(G) + 1$  (Propriétés 22 et 23).

Cependant, comme proposé dans [Minot et al. \(2017\)](#), il est inutile pour un sommet  $v$  du graphe de considérer plus de  $d(v) + 1$  couleurs où  $d(v)$  représente le degré du sommet  $v$ .

**Propriété 24** Soit  $G = (V, E)$  un graphe.  $\forall v \in V, c(v) \leq d(v) + 1$ .

Trivialement, si une solution considère, pour un sommet  $v$ , une couleur supérieure à  $d(v) + 1$  alors il existe une solution pour laquelle  $c(v)$  est inférieure à  $d(v) + 1$ . De ce fait, la somme coloration associée à cette coloration est inférieure à la coloration initiale. Il n'est donc pas nécessaire de considérer plus de  $d(v) + 1$  couleurs pour un sommet  $v$ . En outre, il n'est pas nécessaire de

considérer plus de  $s(G)$  couleurs pour trouver une solution optimale de *MSCP*, où  $s(G)$  est la force chromatique de  $G$ .

Par conséquent, nous proposons de renforcer l'approche de Minot de la façon suivante :

**Propriété 25** Soit  $G = (V, E)$  un graphe.  $\forall v \in V, c(v) \leq \min\{s(G), d(v) + 1\}$ .

Une borne supérieure de  $s(G)$  est utilisée dans la Propriété 25 si  $s(G)$  n'est pas connue.

### 4.3.2 Étude expérimentale sur la réduction des domaines

Pour analyser l'impact des Propriétés 24 et 25 sur la taille de l'instance générée, nous proposons une étude expérimentale sur les benchmarks **COLOR** et **Dimacs**.

Le Tableau 4.1 regroupe l'ensemble des informations obtenues lors de la modélisation des instances *MSCP* en instance *weighted partial MaxSAT*. Pour chaque graphe  $G$ , sont représentés dans les différentes colonnes, le nombre de sommets  $|V|$ , la densité du graphe  $d$ , le degré du graphe  $\Delta(G)$ , une borne supérieure de la force obtenue par l'algorithme  $UB_S$  (Algorithme 10), le nombre de variables propositionnelles ( $\#var$ ) et le nombre de clauses ( $\#clause$ ) associées à un encodage classique considérant  $\Delta(G) + 1$ , un encodage respectant la propriété de Minot (Propriété 24) et un encodage respectant la propriété augmentée de Minot (Propriété 25).

| Graphes    | V    | $d$   | $\Delta(G)$ | $UB_s$ | $\Delta(G) + 1$ |                | $d(v) + 1$  |                | $\min\{k, d(v) + 1\}$ |                |
|------------|------|-------|-------------|--------|-----------------|----------------|-------------|----------------|-----------------------|----------------|
|            |      |       |             |        | <i>#var</i>     | <i>#clause</i> | <i>#var</i> | <i>#clause</i> | <i>#var</i>           | <i>#clause</i> |
| DSJC125.1  | 125  | 0.094 | 23          | 11     | 3000            | 55289          | 1597        | 20454          | 1307                  | 15311          |
| DSJC125.5  | 125  | 0.502 | 75          | 29     | 9500            | 661591         | 7907        | 491926         | 3625                  | 167339         |
| DSJC125.9  | 125  | 0.898 | 120         | 58     | 15125           | 1765031        | 14047       | 1567121        | 7250                  | 617738         |
| DSJC250.1  | 250  | 0.103 | 38          | 21     | 9750            | 320752         | 6686        | 176315         | 5168                  | 122679         |
| DSJC250.5  | 250  | 0.503 | 147         | 50     | 37000           | 5075614        | 31586       | 3936654        | 12500                 | 1102400        |
| DSJC250.9  | 250  | 0.896 | 234         | 105    | 58750           | 13488545       | 56044       | 12496790       | 26250                 | 4320685        |
| DSJC500.1  | 500  | 0.099 | 68          | 52     | 34500           | 2067602        | 25416       | 1266962        | 24335                 | 1197063        |
| DSJC500.5  | 500  | 0.501 | 286         | 81     | 143500          | 38637588       | 125748      | 31295297       | 40500                 | 6733544        |
| DSJC500.9  | 500  | 0.901 | 471         | 185    | 236000          | 108884764      | 225374      | 101204063      | 92500                 | 29403845       |
| DSJC1000.1 | 1000 | 0.099 | 127         | 111    | 128000          | 14609512       | 100258      | 9877170        | 99670                 | 9806559        |
| DSJC1000.5 | 1000 | 0.500 | 551         | 150    | 552000          | 290532952      | 500652      | 248690112      | 150000                | 48799900       |
| DSJC1000.9 | 1000 | 0.899 | 924         | 347    | 925000          | 844016325      | 899898      | 807468555      | 347000                | 216337803      |
| le450_5a   | 450  | 0.056 | 42          | 13     | 19350           | 671852         | 11878       | 309952         | 5850                  | 115682         |
| le450_5b   | 450  | 0.056 | 42          | 15     | 19350           | 672712         | 11918       | 311782         | 6746                  | 140354         |
| le450_5c   | 450  | 0.097 | 66          | 5      | 30150           | 1682351        | 20056       | 877277         | 2250                  | 56215          |
| le450_5d   | 450  | 0.096 | 68          | 5      | 31050           | 1760433        | 19964       | 868539         | 2250                  | 55985          |
| le450_15a  | 450  | 0.080 | 99          | 51     | 45000           | 3089750        | 16786       | 680287         | 15795                 | 612804         |
| le450_15b  | 450  | 0.080 | 94          | 52     | 42750           | 2828505        | 16788       | 675847         | 15944                 | 618415         |
| le450_15c  | 450  | 0.165 | 139         | 50     | 63000           | 6777150        | 33810       | 2546698        | 22063                 | 1374543        |
| le450_15d  | 450  | 0.165 | 138         | 50     | 62550           | 6707200        | 33950       | 2569248        | 22022                 | 1376418        |
| le450_25a  | 450  | 0.081 | 128         | 64     | 58050           | 4839240        | 16970       | 778288         | 16029                 | 698742         |
| le450_25b  | 450  | 0.081 | 111         | 66     | 50400           | 3773506        | 16976       | 749073         | 16355                 | 697335         |
| le450_25c  | 450  | 0.171 | 179         | 74     | 81000           | 10452690       | 35136       | 2862836        | 28844                 | 2117844        |
| le450_25d  | 450  | 0.172 | 157         | 72     | 71100           | 8406050        | 35300       | 2873677        | 28645                 | 2091704        |
| queen5_5   | 25   | 0.533 | 16          | 5      | 425             | 6570           | 345         | 4730           | 125                   | 1200           |
| queen6_6   | 36   | 0.460 | 19          | 10     | 720             | 13396          | 616         | 10452          | 360                   | 4916           |
| queen7_7   | 49   | 0.404 | 24          | 7      | 1225            | 27874          | 1001        | 20258          | 343                   | 4753           |
| queen8_8   | 64   | 0.361 | 27          | 10     | 1792            | 46432          | 1520        | 35744          | 640                   | 10864          |

Continue sur la page suivante...

| Graphes     | V   | $d$   | $\Delta(G)$ | $UB_s$ | $\Delta(G) + 1$ |                | $d(v) + 1$  |                | $\min\{k, d(v) + 1\}$ |                |
|-------------|-----|-------|-------------|--------|-----------------|----------------|-------------|----------------|-----------------------|----------------|
|             |     |       |             |        | <i>#var</i>     | <i>#clause</i> | <i>#var</i> | <i>#clause</i> | <i>#var</i>           | <i>#clause</i> |
| queen8_12   | 96  | 0.300 | 32          | 12     | 3168            | 99096          | 2832        | 82674          | 1152                  | 24000          |
| queen9_9    | 81  | 0.325 | 32          | 11     | 2673            | 80370          | 2193        | 58770          | 891                   | 17043          |
| queen10_10  | 100 | 0.296 | 35          | 12     | 3600            | 119620         | 3040        | 91460          | 1200                  | 25540          |
| queen11_11  | 121 | 0.272 | 40          | 14     | 4961            | 185482         | 4081        | 136202         | 1694                  | 40546          |
| queen12_12  | 144 | 0.252 | 43          | 15     | 6336            | 256928         | 5336        | 195648         | 2160                  | 56364          |
| queen13_13  | 169 | 0.234 | 48          | 16     | 8281            | 370266         | 6825        | 272714         | 2704                  | 76401          |
| queen14_14  | 196 | 0.219 | 51          | 18     | 10192           | 487956         | 8568        | 370580         | 3528                  | 109060         |
| queen15_15  | 225 | 0.205 | 56          | 19     | 12825           | 667410         | 10585       | 492690         | 4275                  | 141395         |
| queen16_16  | 256 | 0.193 | 59          | 21     | 15360           | 847936         | 12896       | 642752         | 5376                  | 192112         |
| myciel3     | 11  | 0.363 | 5           | 4      | 66              | 362            | 51          | 242            | 44                    | 201            |
| myciel4     | 23  | 0.280 | 11          | 6      | 276             | 2669           | 165         | 1182           | 133                   | 882            |
| myciel5     | 47  | 0.218 | 23          | 8      | 1128            | 19811          | 519         | 5817           | 361                   | 3432           |
| myciel6     | 95  | 0.169 | 47          | 11     | 4560            | 148055         | 1605        | 28788          | 993                   | 13783          |
| myciel7     | 191 | 0.130 | 95          | 15     | 18336           | 1116047        | 4911        | 142953         | 2694                  | 54653          |
| fpsol2_i_1  | 496 | 0.094 | 252         | 75     | 125488          | 18885934       | 23804       | 2453789        | 16361                 | 1224832        |
| fpsol2_i_2  | 451 | 0.085 | 346         | 46     | 156497          | 30246706       | 17833       | 1587218        | 11355                 | 456176         |
| fpsol2_i_3  | 425 | 0.096 | 346         | 46     | 147475          | 28675811       | 17801       | 1586717        | 11323                 | 455675         |
| inithx.i.1  | 864 | 0.050 | 502         | 72     | 434592          | 118927669      | 38278       | 4297604        | 24693                 | 1529442        |
| inithx.i.2  | 645 | 0.067 | 541         | 48     | 349590          | 102490948      | 28603       | 2952091        | 17413                 | 720284         |
| inithx.i.3  | 621 | 0.072 | 542         | 48     | 337203          | 99305004       | 28559       | 2957114        | 17343                 | 716051         |
| mulsol.i.1  | 197 | 0.203 | 121         | 59     | 24034           | 1957138        | 8047        | 468357         | 6941                  | 373194         |
| mulsol.i.2  | 188 | 0.221 | 156         | 44     | 29516           | 2941897        | 7958        | 408776         | 5766                  | 228419         |
| mulsol.i.3  | 184 | 0.232 | 157         | 44     | 29072           | 2930136        | 8016        | 413540         | 5795                  | 230057         |
| mulsol.i.4  | 185 | 0.231 | 158         | 44     | 29415           | 2980799        | 8077        | 418184         | 5827                  | 231546         |
| mulsol.i.5  | 186 | 0.230 | 159         | 43     | 29760           | 3031546        | 8132        | 422386         | 5798                  | 229365         |
| school1     | 385 | 0.258 | 282         | 45     | 108955          | 20875880       | 38575       | 4333394        | 15767                 | 1197988        |
| school1-nsh | 352 | 0.236 | 232         | 43     | 82016           | 13000820       | 29576       | 2712478        | 13826                 | 914211         |
| zeroin.i.1  | 211 | 0.185 | 111         | 56     | 23632           | 1794619        | 8411        | 571821         | 6277                  | 358764         |

Continue sur la page suivante...

| Graphes       | V    | $d$   | $\Delta(G)$ | $UB_s$ | $\Delta(G) + 1$ |                | $d(v) + 1$  |                | $\min\{k, d(v) + 1\}$ |                |
|---------------|------|-------|-------------|--------|-----------------|----------------|-------------|----------------|-----------------------|----------------|
|               |      |       |             |        | <i>#var</i>     | <i>#clause</i> | <i>#var</i> | <i>#clause</i> | <i>#var</i>           | <i>#clause</i> |
| zeroin.i.2    | 211  | 0.159 | 140         | 39     | 29751           | 2611813        | 7293        | 401365         | 5026                  | 193830         |
| zeroin.i.3    | 206  | 0.167 | 140         | 39     | 29046           | 2561612        | 7286        | 401207         | 5019                  | 193672         |
| anna          | 138  | 0.052 | 71          | 14     | 9936            | 398298         | 1124        | 17587          | 851                   | 8276           |
| david         | 87   | 0.108 | 82          | 15     | 7221            | 337067         | 899         | 14333          | 731                   | 8416           |
| huck          | 74   | 0.111 | 53          | 16     | 3996            | 126218         | 676         | 8346           | 607                   | 6322           |
| jean          | 80   | 0.080 | 36          | 15     | 2960            | 65718          | 588         | 6252           | 546                   | 5268           |
| games120      | 120  | 0.089 | 13          | 15     | 1680            | 21652          | 1396        | 16206          | 1396                  | 16206          |
| flat300-20-0  | 300  | 0.476 | 160         | 20     | 48300           | 7353975        | 43050       | 6113369        | 6000                  | 490800         |
| flat300-26-0  | 300  | 0.482 | 158         | 36     | 47700           | 7255947        | 43566       | 6264408        | 10800                 | 978888         |
| flat300-28-0  | 300  | 0.483 | 162         | 49     | 48900           | 7546385        | 43690       | 6296586        | 14700                 | 1430855        |
| flat1000-50-0 | 1000 | 0.490 | 520         | 50     | 521000          | 263627000      | 491000      | 239836344      | 50000                 | 13526000       |
| flat1000-60-0 | 1000 | 0.492 | 524         | 77     | 525000          | 267136750      | 492660      | 241264561      | 77000                 | 21932910       |
| flat1000-76-0 | 1000 | 0.493 | 532         | 145    | 533000          | 273807364      | 494416      | 242905731      | 145000                | 46358660       |
| miles250      | 128  | 0.047 | 16          | 14     | 2176            | 26291          | 902         | 7738           | 893                   | 7586           |
| miles500      | 128  | 0.143 | 38          | 25     | 4992            | 145598         | 2468        | 56700          | 2210                  | 46547          |
| miles750      | 128  | 0.259 | 64          | 38     | 8320            | 412033         | 4354        | 166580         | 3756                  | 130884         |
| miles1000     | 128  | 0.395 | 86          | 47     | 11136           | 769904         | 6560        | 362907         | 5149                  | 249153         |
| miles1500     | 128  | 0.639 | 106         | 77     | 13696           | 1295898        | 10524       | 899584         | 8876                  | 700894         |

TABLEAU 4.1 – Comparaison de la taille des instances générées.

Les résultats du Tableau 4.1 montrent que l'approche proposée par Minot permet de réduire de façon significative la taille des instances générées pour certains graphes. Par exemple, nous remarquons que cette approche a permis de diviser par sept le nombre de clauses sur l'instance *mul-sol.i.3*. Cependant, nous remarquons que sur certains graphes, tels que l'instance *DSJC250.9*, la réduction observée est faible. Toutefois, la considération d'une borne supérieure pour la force en plus du degré du sommet (Propriété 25) permet une réduction importante du nombre de clauses pour ces graphes. En effet, pour l'instance *DSJC250.9*, une approche classique considère 13488545 clauses, une approche suivant la propriété de Minot compte 12496790 clauses et une approche augmentée en dénombre 4320685. Cependant, nous remarquons que selon les familles de graphes, l'impact d'une approche augmentée (Propriété 25) sur un encodage respectant la Propriété 24 diffère.

À travers les Figures 4.2, 4.3, 4.4, 4.5, 4.6, 4.7 et 4.8, nous proposons d'étudier l'efficacité de ces approches pour certaines familles de graphes. Chaque figure représente un nuage de points comparant les taux de réduction, en terme de nombre de clauses, des encodages selon les différentes familles de graphes. L'axe des abscisses est associé au taux de réduction entre un encodage classique ( $\Delta(G) + 1$ ) et un encodage respectant la propriété de Minot (Propriété 24). L'axe des ordonnées est associé au taux de réduction entre un encodage classique ( $\Delta(G) + 1$ ) et un encodage respectant la propriété augmentée de Minot (Propriété 25). Un point éloigné de l'axe de la diagonale signifie qu'un encodage respectant la propriété augmentée de Minot a permis de réduire significativement la taille de l'instance générée vis-à-vis d'un encodage considérant la Propriété 24.

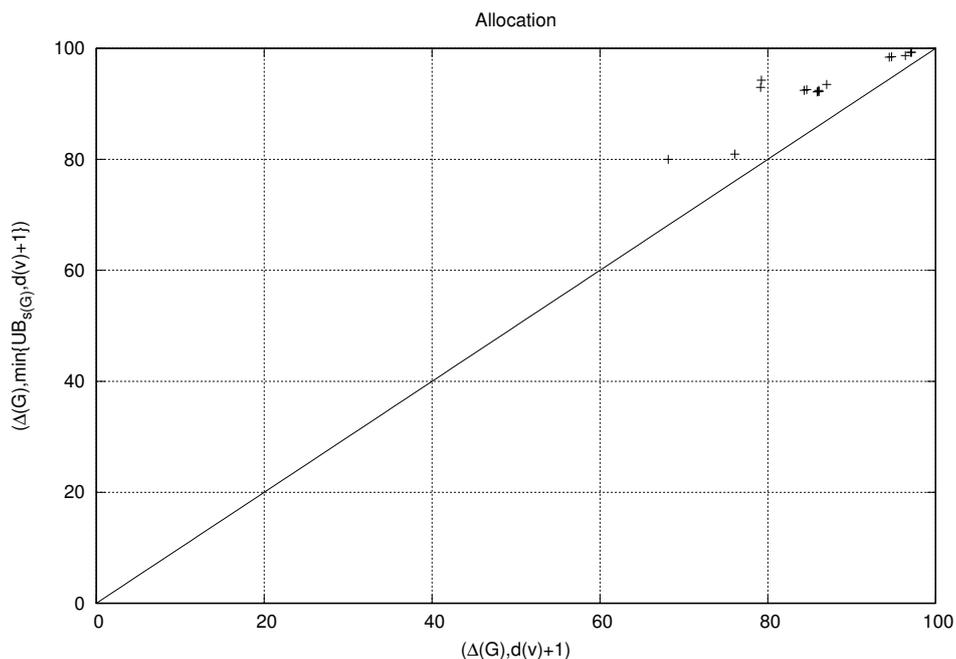


FIGURE 4.2 – Nuage de points comparant le taux de réduction des graphes de type Allocation dont la densité moyenne est égale à 0.149.

L'analyse des différents nuages de points (Figures 4.2, 4.3, 4.4, 4.5, 4.6, 4.7 et 4.8) permet d'observer que pour certaines familles de graphes, telles que les graphes d'allocation, l'instance *MaxSAT/MinSAT* générée par un encodage respectant la Propriété 25 est très similaire à un encodage respectant la Propriété 24. Cela signifie que sur ces graphes, la considération d'une borne supérieure de la force ne permet pas une réduction significative de la taille de l'instance générée. À l'inverse, pour d'autres familles de graphes, telles que les Queens, une approche augmentée a pour conséquence une réduction moyenne de la taille des instances d'environ 70% alors qu'un encodage respectant la Propriété 24 a un taux de réduction d'environ 25%.

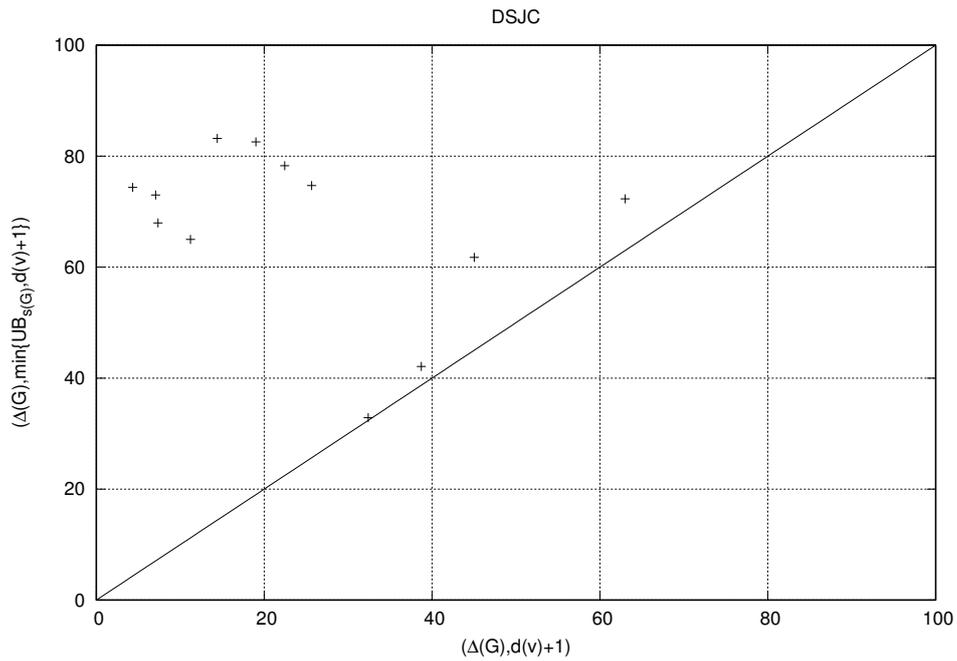


FIGURE 4.3 – Nuage de points comparant le taux de réduction des instances de type DSJC dont la densité moyenne est égale à 0.499.

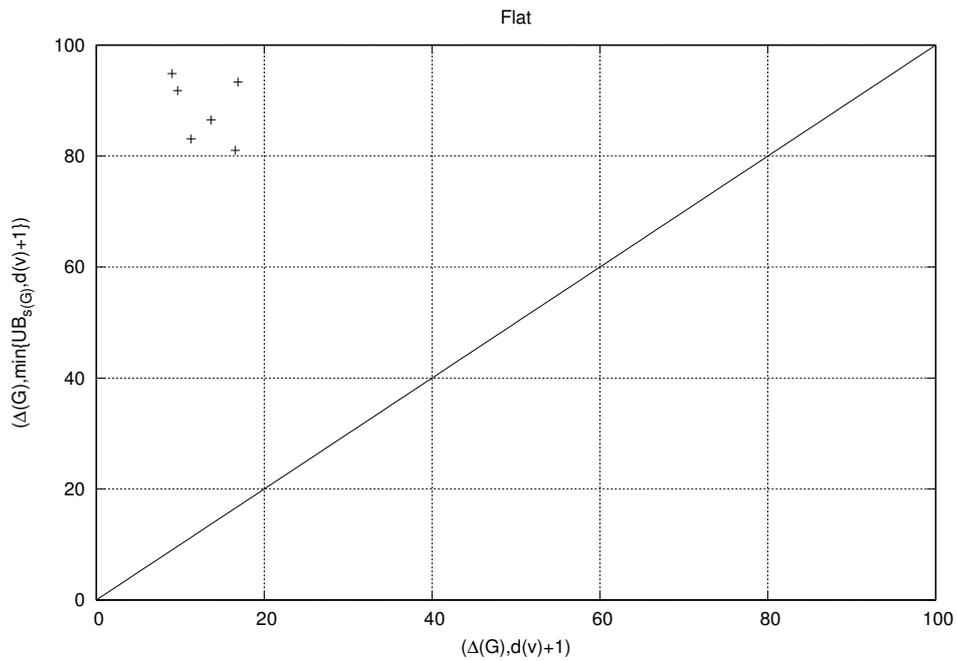


FIGURE 4.4 – Nuage de points comparant le taux de réduction des instances de type Flat dont la densité moyenne est égale à 0.486.

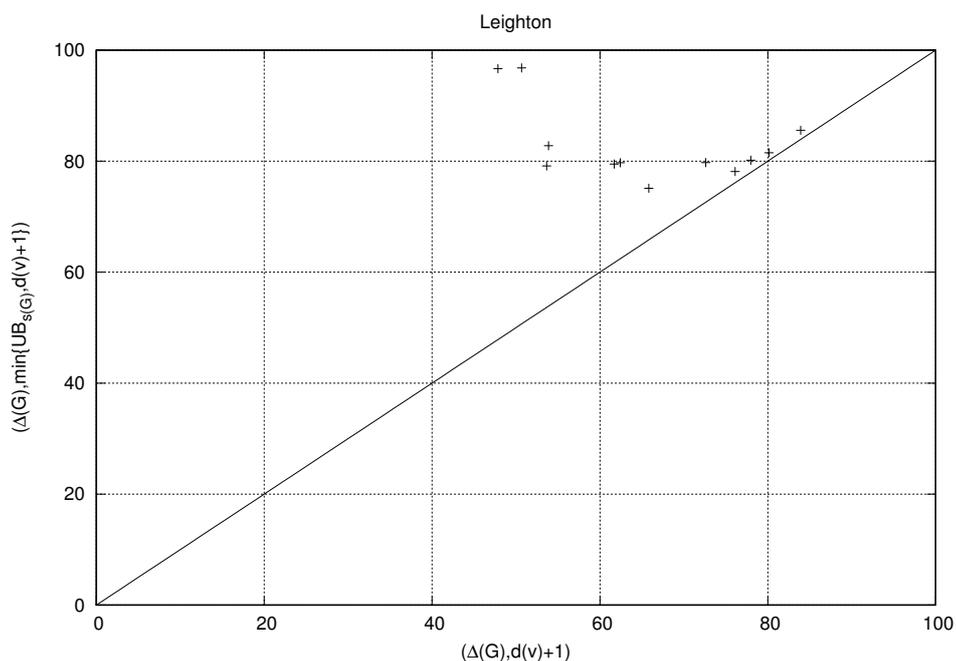


FIGURE 4.5 – Nuage de points comparant le taux de réduction des instances de type Leighton dont la densité moyenne est égale à 0.108.

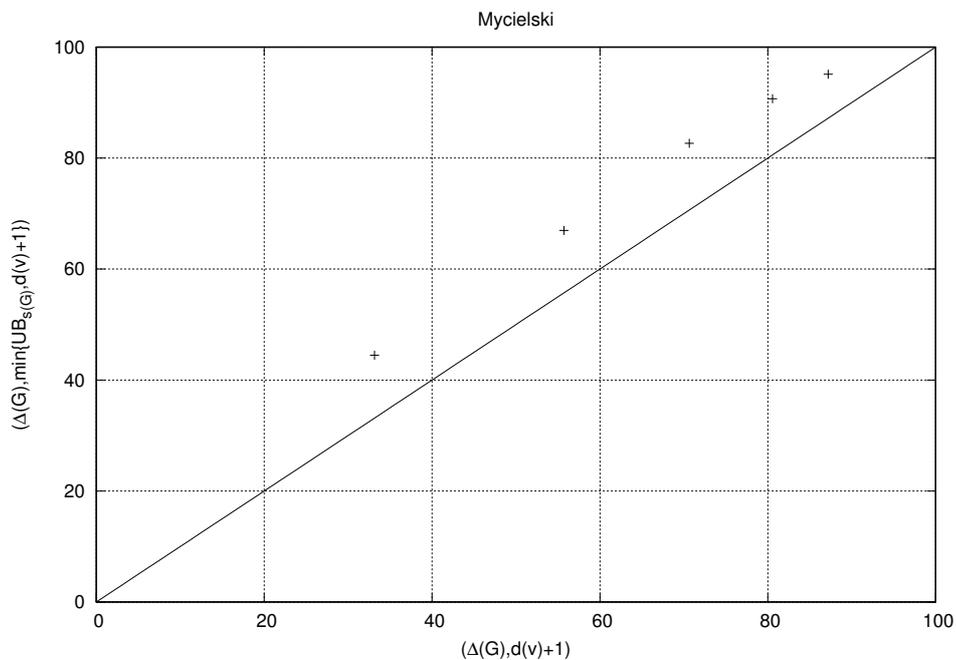


FIGURE 4.6 – Nuage de points comparant le taux de réduction des instances de type Mycielski dont la densité moyenne est égale à 0.232.

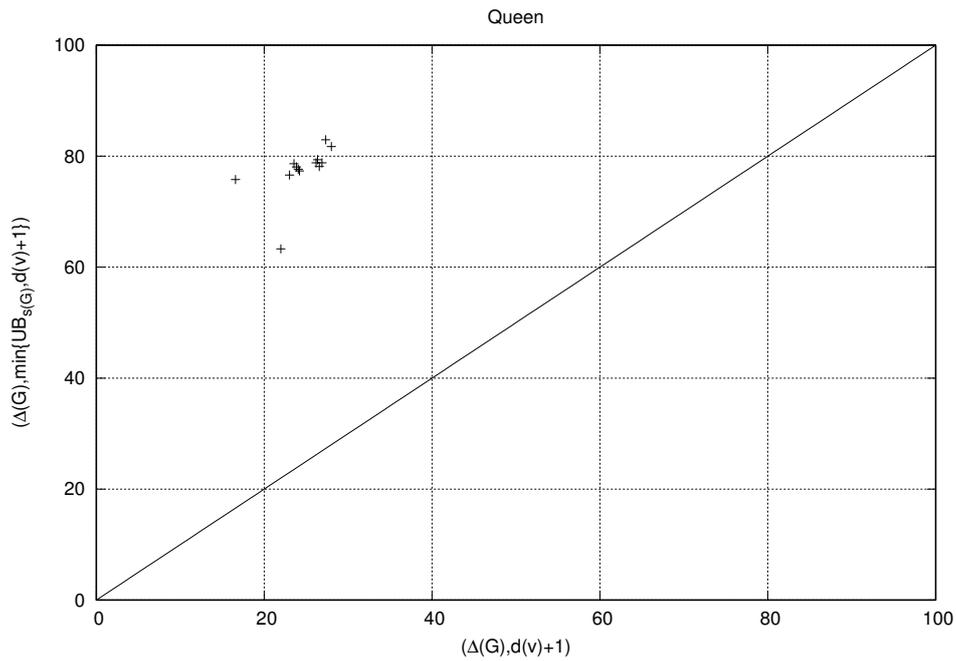


FIGURE 4.7 – Nuage de points comparant le taux de réduction des instances de type Queen dont la densité moyenne est égale à 0.311.

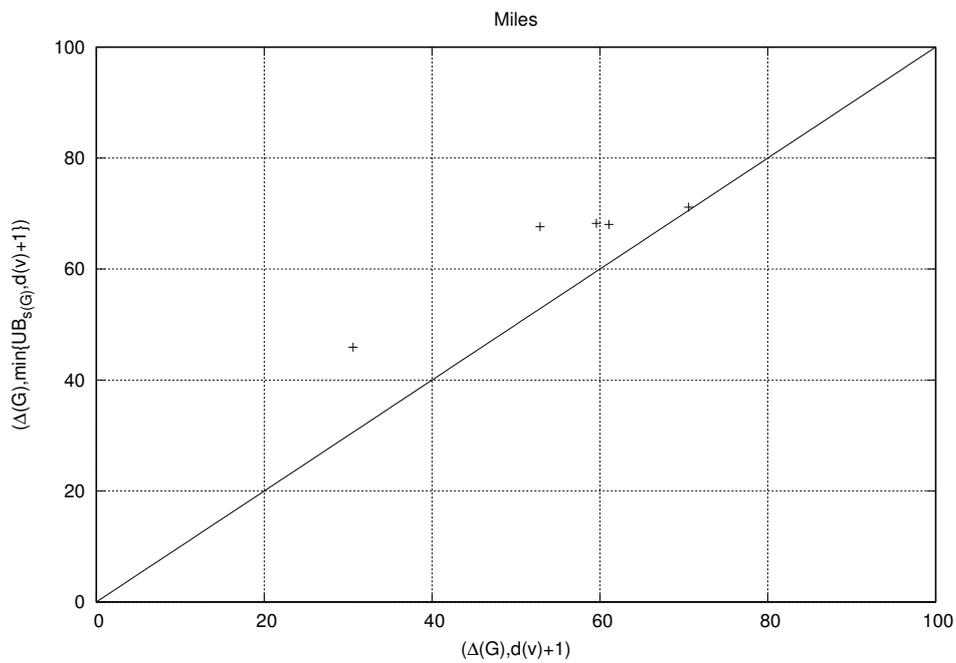


FIGURE 4.8 – Nuage de points comparant le taux de réduction des instances de type Miles dont la densité moyenne est égale à 0.292.

Les Figures 4.9, 4.10 et 4.11 proposent de mettre en relation le taux de réduction du nombre de clauses des encodages et la densité des graphes. L'axe des abscisses est associé à la densité. L'axe des ordonnées est associé au taux de réduction entre :

- un encodage classique ( $\Delta(G) + 1$ ) et un encodage respectant la propriété de Minot (Propriété 24) pour la Figure 4.9;
- un encodage classique ( $\Delta(G) + 1$ ) et un encodage augmenté (Propriété 25) pour la Figure 4.10;
- un encodage respectant la propriété de Minot (Propriété 24) et un encodage augmenté (Propriété 25) pour la Figure 4.11.

Les Figures 4.9 et 4.10 montrent que pour quelques graphes dont la densité est inférieure à 0.1, la prise en considération d'une borne supérieure de la force d'un graphe pour la construction de l'instance *MaxSAT/MinSAT* ne permet qu'une faible réduction de la taille de l'instance en comparaison avec un encodage considérant uniquement le degré d'un sommet. En effet, nous remarquons que les points tracés correspondant à ces graphes sont très similaires. Cependant, pour une densité supérieure à 0.1, un encodage augmenté permet pour la majorité des instances une réduction de 50% du nombre de clauses contrairement à un encodage considérant uniquement le degré d'un sommet. De plus, la Figure 4.11 montre que pour tout graphe dont la densité est supérieure à 0.3, la taille de l'instance générée par un encodage augmenté a un taux de réduction de 50% par rapport à la taille d'une instance encodée en respectant la propriété de Minot (Propriété 24). Une explication de ce phénomène est que pour un graphe de faible densité (c'est à dire, dont la densité est inférieure à 0.1), le degré de la plupart des sommets peut être proche de la force du graphe. Dans ce cas, l'ajout de la force du graphe n'améliore pas significativement l'encodage. Cependant, pour un graphe dense (dont la densité est supérieure à 0.3), le degré de beaucoup de sommets est largement supérieur à la force du graphe. Dans ce cas, l'ajout de la force du graphe réduit significativement la taille de l'instance *MaxSAT/MinSAT* obtenue.

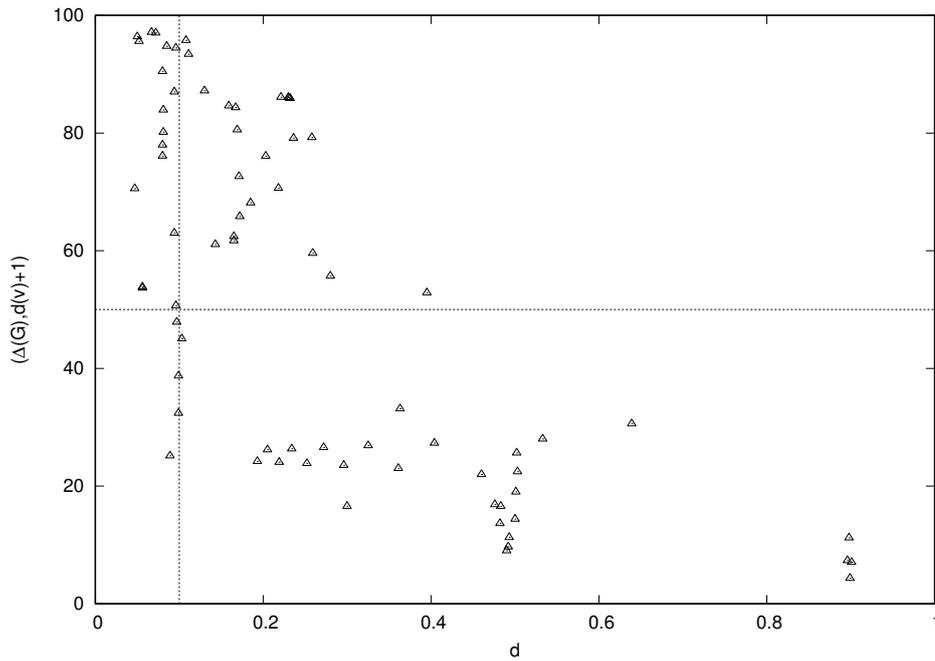


FIGURE 4.9 – Nuage de points comparant le taux de réduction entre un encodage classique  $(\Delta(G) + 1)$  avec un encodage respectant la propriété de Minot (Propriété 24) et la densité d'un graphe.

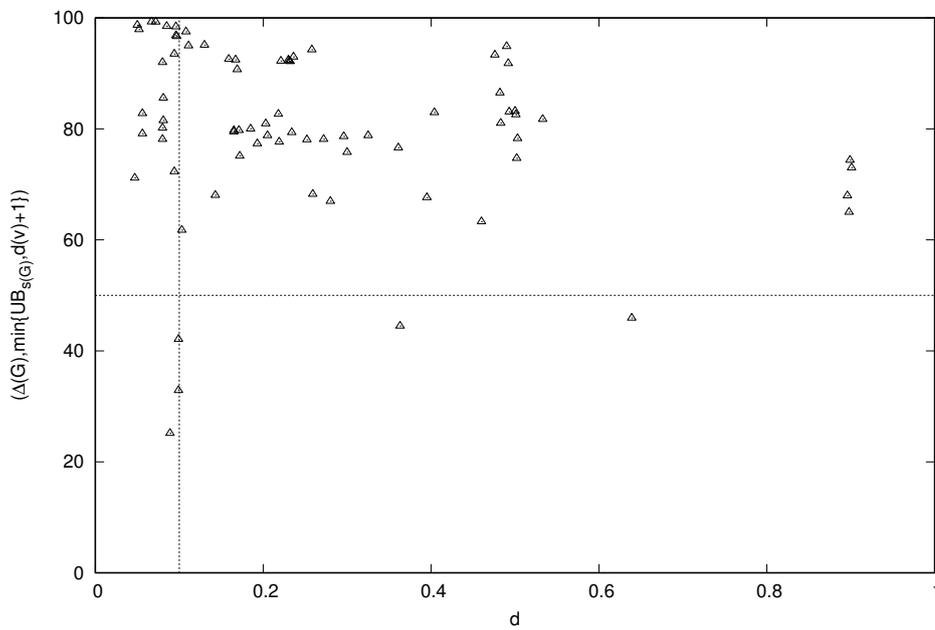


FIGURE 4.10 – Nuage de points comparant le taux de réduction entre un encodage classique  $(\Delta(G) + 1)$  avec un encodage augmenté (Propriété 25) et la densité d'un graphe.

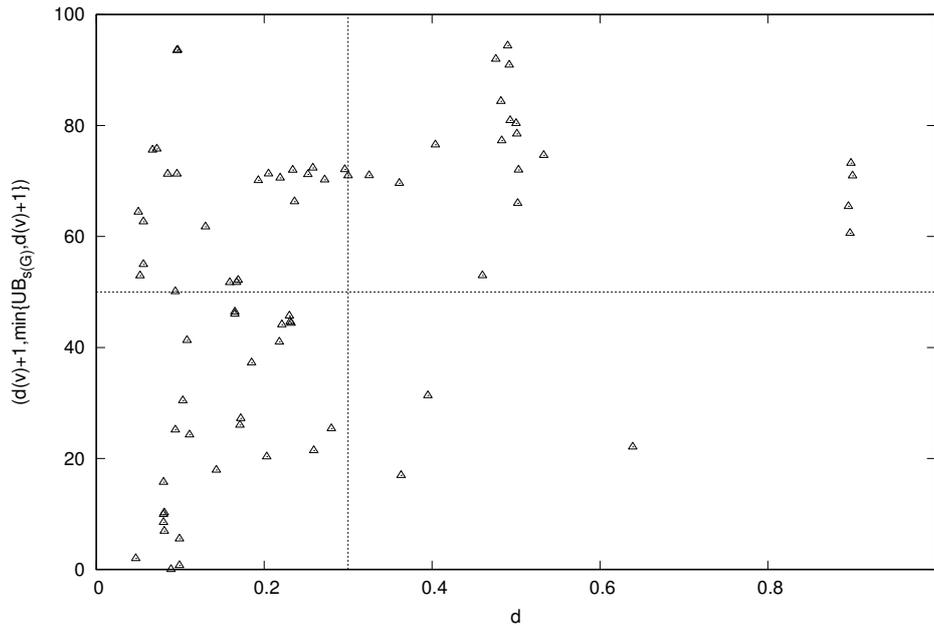


FIGURE 4.11 – Nuage de points comparant le taux de réduction entre un encodage respectant la propriété de Minot (Propriété 24) avec un encodage augmenté (Propriété 25) et la densité d'un graphe.

#### 4.4 Étude des performances des encodages MaxSAT et MinSAT

Dans cette section, nous étudions expérimentalement nos différentes approches, afin de caractériser le comportement des différents encodages sur le problème *weighted partial MaxSAT* et le problème *weighted partial MinSAT*. Ces encodages utilisent la Propriété 25 pour réduire la taille des instances *MaxSAT* et *MinSAT* obtenues. L'ensemble des expérimentations a été mené sur la plateforme MatriCS équipée de processeurs Intel Xeon CPUs E5-2680 v4@2.40GHz sous Linux avec 128GB de mémoire. Pour la résolution du problème *weighted partial MaxSAT*, nous avons considéré la version de 2013 du solveur *ISAC* Ansótegui *et al.* (2016) et pour la résolution du problème *weighted partial MinSAT*, nous avons considéré la version de 2013 du solveur *MinSatz* Li *et al.* (2011).

Le Tableau 4.2 reporte les résultats obtenus sur les instances *COLOR* (COLOR) et *DIMACS* (Dimacs). Dans cette expérimentation, nous avons fixé la limite de temps à 4 heures pour chaque solveur et pour chaque graphe. Un processus ayant dépassé cette limite de temps est stoppé (N/A). Pour chaque graphe  $G$ , sont représentés dans les différentes colonnes, le nombre de sommets  $|V|$ , la densité du graphe  $d$  et les temps en secondes pour les différents encodages du problème *weighted partial MaxSAT* et du problème *weighted partial MinSAT*.

| Graphes   | $ V $ | $d$   | MaxSAT     |            | MinSAT     |            |
|-----------|-------|-------|------------|------------|------------|------------|
|           |       |       | Encodage 1 | Encodage 2 | Encodage 1 | Encodage 2 |
| le450_5c  | 450   | 0.097 | <b>111</b> | 143        | N/A        | N/A        |
| le450_5d  | 450   | 0.096 | 122        | <b>90</b>  | N/A        | N/A        |
| queen5_5  | 25    | 0.533 | <b>0</b>   | <b>0</b>   | <b>0</b>   | <b>0</b>   |
| queen6_6  | 36    | 0.460 | <b>192</b> | 355        | N/A        | N/A        |
| queen7_7  | 49    | 0.404 | <b>0</b>   | <b>0</b>   | N/A        | N/A        |
| queen8_12 | 96    | 0.300 | <b>1</b>   | <b>1</b>   | N/A        | N/A        |
| myciel3   | 11    | 0.363 | <b>0</b>   | <b>0</b>   | <b>0</b>   | <b>0</b>   |
| myciel4   | 23    | 0.280 | <b>0</b>   | <b>0</b>   | <b>0</b>   | 7684       |

Continue sur la page suivante...

| Graphes    | V   | $d$   | MaxSAT     |            | MinSAT     |            |
|------------|-----|-------|------------|------------|------------|------------|
|            |     |       | Encodage 1 | Encodage 2 | Encodage 1 | Encodage 2 |
| myciel5    | 47  | 0.218 | 1          | <b>0</b>   | 77         | N/A        |
| myciel6    | 95  | 0.169 | <b>40</b>  | 41         | N/A        | N/A        |
| fpsol2_i_1 | 496 | 0.094 | <b>11</b>  | 12         | N/A        | N/A        |
| fpsol2_i_2 | 451 | 0.085 | <b>5</b>   | <b>5</b>   | N/A        | N/A        |
| fpsol2_i_3 | 425 | 0.096 | <b>5</b>   | <b>5</b>   | N/A        | N/A        |
| inithx.i.1 | 864 | 0.050 | <b>16</b>  | 18         | N/A        | N/A        |
| inithx.i.2 | 645 | 0.067 | <b>8</b>   | <b>8</b>   | N/A        | N/A        |
| inithx.i.3 | 621 | 0.072 | <b>9</b>   | <b>9</b>   | N/A        | N/A        |
| mulsol.i.1 | 197 | 0.203 | <b>2</b>   | <b>2</b>   | N/A        | N/A        |
| mulsol.i.2 | 188 | 0.221 | <b>1</b>   | <b>1</b>   | N/A        | N/A        |
| mulsol.i.3 | 184 | 0.232 | <b>1</b>   | <b>1</b>   | N/A        | N/A        |
| mulsol.i.4 | 185 | 0.231 | <b>1</b>   | <b>1</b>   | N/A        | N/A        |
| mulsol.i.5 | 186 | 0.230 | <b>1</b>   | <b>1</b>   | N/A        | N/A        |
| zeroin.i.1 | 211 | 0.185 | <b>3</b>   | <b>3</b>   | N/A        | N/A        |
| zeroin.i.2 | 211 | 0.159 | <b>1</b>   | <b>1</b>   | N/A        | N/A        |
| zeroin.i.3 | 206 | 0.167 | <b>1</b>   | <b>1</b>   | N/A        | N/A        |
| anna       | 138 | 0.052 | <b>0</b>   | <b>0</b>   | N/A        | N/A        |
| david      | 87  | 0.108 | <b>0</b>   | <b>0</b>   | N/A        | N/A        |
| homer      | 561 | 0.020 | <b>3</b>   | 16         | N/A        | N/A        |
| huck       | 74  | 0.111 | <b>0</b>   | <b>0</b>   | N/A        | N/A        |
| jean       | 80  | 0.080 | <b>0</b>   | <b>0</b>   | N/A        | N/A        |
| games120   | 120 | 0.089 | <b>0</b>   | 10         | N/A        | N/A        |
| miles250   | 128 | 0.047 | <b>0</b>   | <b>0</b>   | N/A        | N/A        |
| miles500   | 128 | 0.143 | 12         | <b>5</b>   | N/A        | N/A        |
| miles750   | 128 | 0.259 | 117        | <b>99</b>  | N/A        | N/A        |
| miles1000  | 128 | 0.395 | 188        | <b>177</b> | N/A        | N/A        |
| miles1500  | 128 | 0.639 | <b>50</b>  | 56         | N/A        | N/A        |

TABEAU 4.2 – Temps de résolution en seconde des différents encodages sur les instances *COLOR* (COLOR) et *DIMACS* (Dimacs).

Cette première expérimentation met en évidence l'efficacité du solveur *ISAC* pour la résolution du problème *weighted partial MaxSAT*. En effet, sur les 76 instances testées, 34 ont été résolues par *ISAC* contre 3 pour le solveur *MinSatz*. Nous remarquons également que l'encodage semble avoir un impact sur les performances des différents solveurs. En effet, si nous considérons *weighted partial MinSAT*, nous constatons que l'instance *myciel4* a un temps de résolution inférieur à 1 seconde pour l'encodage 1 tandis que le temps de résolution est de 7684 secondes pour l'encodage 2. De plus, l'encodage 1 permet la résolution de l'instance *myciel5* contrairement à l'encodage 2. Pour le problème *weighted partial MaxSAT*, nous remarquons que l'encodage 2 semble plus intéressant que l'encodage 1 pour tout graphe ayant une densité supérieure à 0.1 à l'exception de l'instance *queen6\_6*. Dans le cas où la densité est faible, aucune tendance ne se dégage. En effet, l'encodage 1 est meilleur que l'encodage 2 pour l'instance *le450\_c* de densité 0.096. Cependant, pour l'instance de la même famille *le450\_d* de densité 0.096, l'encodage 2 est meilleur que le 1.

Afin de confirmer cette tendance, nous proposons un nouveau jeu de tests reporté dans le Tableau 4.3 portant sur un ensemble de graphes aléatoires  $rn_d$  où  $n$  représente le nombre de sommets du graphe et  $d$  sa densité. Pour chaque famille de graphes  $rn_d$ , 10 graphes ont été générés. Nous reportons dans le Tableau 4.3 le temps moyen en seconde pour déterminer une solution. Entre parenthèses est reporté le nombre d'instances résolues. Nous avons fixé la limite de temps à 4 heures pour chaque solveur et pour chaque graphe. Si après 4 heures un solveur n'a résolu aucun graphe pour une famille donnée alors nous reportons dans le Tableau 4.3 *N/A*. Les solveurs ayant résolu le plus d'instances pour une même famille sont colorés. Parmi ces solveurs, ceux possédant

le plus petit temps moyen de résolution sont considérés comme meilleurs et colorés en foncés.

| Graphes | MaxSAT      |            | MinSAT      |            |
|---------|-------------|------------|-------------|------------|
|         | Encodage 1  | Encodage 2 | Encodage 1  | Encodage 2 |
| r30_0.1 | 0(10)       | 0(10)      | 0(10)       | 2176.33(3) |
| r30_0.5 | 5.30(10)    | 5.00(10)   | N/A         | N/A        |
| r30_0.9 | 2.90(10)    | 1.50(10)   | N/A         | N/A        |
| r35_0.1 | 0(10)       | 0(10)      | 0(10)       | 5725.00(2) |
| r35_0.5 | 47.20(10)   | 48.10(10)  | N/A         | N/A        |
| r35_0.9 | 569.33(9)   | 83.30(10)  | N/A         | N/A        |
| r40_0.1 | 0(10)       | 0(10)      | 2.70(10)    | N/A        |
| r40_0.5 | 1024.10(10) | 981.40(10) | N/A         | N/A        |
| r40_0.9 | 679.00(4)   | 381.60(5)  | N/A         | N/A        |
| r45_0.1 | 0(10)       | 0(10)      | 15.50(10)   | N/A        |
| r45_0.5 | 3151.25(8)  | 2699.62(8) | N/A         | N/A        |
| r45_0.9 | 464.75(4)   | 205.33(6)  | N/A         | N/A        |
| r50_0.1 | 0.20(10)    | 0.70(10)   | 1039.00(10) | N/A        |
| r50_0.5 | N/A         | N/A        | N/A         | N/A        |
| r50_0.9 | N/A         | 1465.00(2) | N/A         | N/A        |
| r55_0.1 | 3.20(10)    | 3.10(10)   | 1183.14(7)  | N/A        |
| r55_0.5 | N/A         | N/A        | N/A         | N/A        |
| r55_0.9 | N/A         | 543.00(1)  | N/A         | N/A        |
| r60_0.1 | 12.20(10)   | 13.90(10)  | 1762.80(5)  | N/A        |
| r60_0.5 | N/A         | N/A        | N/A         | N/A        |
| r60_0.9 | N/A         | N/A        | N/A         | N/A        |
| r70_0.1 | 14.90(10)   | 14.80(10)  | 6862.33(3)  | N/A        |
| r70_0.5 | N/A         | N/A        | N/A         | N/A        |
| r70_0.9 | N/A         | N/A        | N/A         | N/A        |
| r80_0.1 | 299.70(10)  | 281.00(10) | N/A         | N/A        |
| r80_0.5 | N/A         | N/A        | N/A         | N/A        |
| r80_0.9 | N/A         | N/A        | N/A         | N/A        |
| r90_0.1 | 2990.42(7)  | 3364.71(7) | N/A         | N/A        |
| r90_0.5 | N/A         | N/A        | N/A         | N/A        |
| r90_0.9 | N/A         | N/A        | N/A         | N/A        |

TABLEAU 4.3 – Moyenne des temps de résolution en seconde des différents encodages sur un ensemble d'instances aléatoires.

Nous pouvons observer dans le Tableau 4.3 que pour le problème *weighted partial MinSAT* l'encodage 1 est plus efficace que l'encodage 2. En ce qui concerne le problème *weighted partial MaxSAT*, nous remarquons l'efficacité de l'encodage 2 pour tout graphe de densité supérieure à 0.1. En effet, elle permet la résolution d'un plus grand nombre d'instances en un temps moyen inférieur à celui de l'encodage 1. Cependant, dans le cas où la densité moyenne est inférieure à 0.1, nous ne pouvons déterminer lequel des deux encodages est à sélectionner.

Le Tableau 4.4 compare les temps de résolution des encodages 1 et 2 du problème *weighted partial MaxSAT* au meilleur temps de résolution connu pour *MSCP* [Minot et al. \(2016a,b, 2017\)](#); [Wang et al. \(2013\)](#). Pour chaque graphe  $G$ , sont représentés dans les différentes colonnes, le nombre de sommets  $|V|$ , la densité du graphe  $d$ , le temps en seconde pour les encodages 1 et 2 du problème *weighted partial MaxSAT* et le meilleur temps de résolution en seconde connu pour le problème *MSCP*  $T_{Best}$ . Les résultats de  $T_{Best}$  ont été obtenus sur une unité de calcul similaire à celle utilisée pour nos tests. Nous remarquons que le benchmark d'instances considérées a été étendu pour cette expérimentation. Le temps de résolution a été fixé à 24 heures. Nous reportons

N/A quand un solveur n'a pas déterminé la somme chromatique.

| Graphes     | V   | $d$   | MaxSAT       |            | $T_{Best}$   |
|-------------|-----|-------|--------------|------------|--------------|
|             |     |       | Encodage 1   | Encodage 2 |              |
| le450_5c    | 450 | 0.097 | <b>111</b>   | 164        | 4614         |
| le450_5d    | 450 | 0.096 | 123          | <b>90</b>  | 11317        |
| queen5_5    | 25  | 0.533 | <b>0</b>     | <b>0</b>   | <b>0</b>     |
| queen6_6    | 36  | 0.460 | <b>205</b>   | 400        | 451          |
| queen7_7    | 49  | 0.404 | <b>0</b>     | <b>0</b>   | 1            |
| queen8_8    | 64  | 0.361 | <b>17141</b> | 45786      | 29286        |
| queen8_12   | 96  | 0.300 | <b>1</b>     | <b>1</b>   | 2            |
| myciel3     | 11  | 0.363 | <b>0</b>     | <b>0</b>   | <b>0</b>     |
| myciel4     | 23  | 0.280 | <b>0</b>     | <b>0</b>   | <b>0</b>     |
| myciel5     | 47  | 0.218 | 1            | <b>0</b>   | 2            |
| myciel6     | 95  | 0.169 | 40           | 40         | <b>16</b>    |
| myciel7     | 191 | 0.130 | 19637        | 68654      | <b>11036</b> |
| fpsol2_i_1  | 496 | 0.094 | 12           | 13         | <b>10</b>    |
| fpsol2_i_2  | 451 | 0.085 | <b>5</b>     | <b>5</b>   | 7            |
| fpsol2_i_3  | 425 | 0.096 | <b>5</b>     | <b>5</b>   | 7            |
| inithx.i.1  | 864 | 0.050 | <b>18</b>    | <b>18</b>  | 23           |
| inithx.i.2  | 645 | 0.067 | <b>8</b>     | <b>8</b>   | 13           |
| inithx.i.3  | 621 | 0.072 | <b>9</b>     | <b>9</b>   | 12           |
| mulsol.i.1  | 197 | 0.203 | 2            | 2          | <b>1</b>     |
| mulsol.i.2  | 188 | 0.221 | <b>1</b>     | <b>1</b>   | <b>1</b>     |
| mulsol.i.3  | 184 | 0.232 | <b>1</b>     | <b>1</b>   | <b>1</b>     |
| mulsol.i.4  | 185 | 0.231 | <b>1</b>     | <b>1</b>   | <b>1</b>     |
| mulsol.i.5  | 186 | 0.230 | <b>1</b>     | <b>1</b>   | <b>1</b>     |
| zeroin.i.1  | 211 | 0.159 | 3            | 3          | <b>2</b>     |
| zeroin.i.2  | 211 | 0.157 | <b>1</b>     | <b>1</b>   | <b>1</b>     |
| zeroin.i.3  | 206 | 0.167 | <b>1</b>     | <b>1</b>   | <b>1</b>     |
| anna        | 138 | 0.052 | <b>0</b>     | <b>0</b>   | <b>0</b>     |
| david       | 87  | 0.108 | <b>0</b>     | <b>0</b>   | <b>0</b>     |
| homer       | 561 | 0.010 | <b>3</b>     | 16         | 25           |
| huck        | 74  | 0.111 | <b>0</b>     | <b>0</b>   | <b>0</b>     |
| jean        | 80  | 0.080 | <b>0</b>     | <b>0</b>   | <b>0</b>     |
| games120    | 120 | 0.089 | <b>0</b>     | 10         | 1            |
| miles250    | 128 | 0.047 | <b>0</b>     | <b>0</b>   | <b>0</b>     |
| miles500    | 128 | 0.143 | 13           | <b>5</b>   | 24           |
| miles750    | 128 | 0.259 | 119          | <b>94</b>  | 144          |
| miles1000   | 128 | 0.395 | 193          | 183        | <b>113</b>   |
| miles1500   | 128 | 0.639 | 50           | 56         | <b>9</b>     |
| 1-FullIns_3 | 30  | 0.229 | <b>0</b>     | <b>0</b>   | <b>0</b>     |
| 1-FullIns_4 | 93  | 0.138 | <b>1</b>     | <b>1</b>   | 2            |
| 1-FullIns_5 | 282 | 0.081 | <b>213</b>   | 289        | 602          |
| 2-FullIns_3 | 52  | 0.151 | <b>0</b>     | <b>0</b>   | <b>0</b>     |
| 2-FullIns_4 | 212 | 0.072 | 17           | <b>15</b>  | 22           |
| 3-FullIns_3 | 80  | 0.109 | <b>0</b>     | <b>0</b>   | 1            |
| 3-FullIns_4 | 405 | 0.043 | <b>173</b>   | 192        | 178          |
| 4-FullIns_3 | 114 | 0.083 | <b>0</b>     | <b>0</b>   | <b>0</b>     |
| 4-FullIns_4 | 690 | 0.027 | <b>633</b>   | 4224       | 6572         |

Continue sur la page suivante...

| Graphes        | V    | $d$   | MaxSAT      |              | $T_{Best}$ |
|----------------|------|-------|-------------|--------------|------------|
|                |      |       | Encodage 1  | Encodage 2   |            |
| 5-FullIns_3    | 154  | 0.067 | <b>0</b>    | 1            | 2          |
| 5-FullIns_4    | 1085 | 0.019 | <b>4022</b> | N/A          | 48434      |
| 1-Insertions_4 | 67   | 0.104 | <b>0</b>    | 1            | 1          |
| 1-Insertions_5 | 202  | 0.060 | 430         | 154          | <b>109</b> |
| 2-Insertions_3 | 37   | 0.108 | <b>0</b>    | <b>0</b>     | <b>0</b>   |
| 2-Insertions_4 | 149  | 0.049 | <b>5</b>    | <b>5</b>     | 7          |
| 2-Insertions_5 | 597  | 0.071 | N/A         | <b>71310</b> | N/A        |
| 3-Insertions_3 | 56   | 0.068 | <b>0</b>    | <b>0</b>     | <b>0</b>   |
| 3-Insertions_4 | 281  | 0.026 | 75          | 114          | <b>54</b>  |
| 4-Insertions_3 | 79   | 0.050 | <b>0</b>    | <b>0</b>     | <b>0</b>   |
| 4-Insertions_4 | 475  | 0.015 | 978         | 798          | <b>373</b> |
| mug88_1        | 88   | 0.038 | <b>1</b>    | <b>1</b>     | <b>1</b>   |
| mug88_25       | 88   | 0.038 | <b>0</b>    | 1            | 1          |
| mug100_1       | 100  | 0.033 | <b>1</b>    | <b>1</b>     | 2          |
| mug100_25      | 100  | 0.033 | <b>1</b>    | <b>1</b>     | 5          |
| qg.order30     | 900  | 0.064 | 268         | 210          | <b>0</b>   |
| qg.order40     | 1600 | 0.048 | 12030       | 1574         | <b>5</b>   |

TABEAU 4.4 – Comparaison des temps de résolution en seconde des différents encodages du problème *weighted partial MaxSAT* face aux meilleurs temps connus de la littérature pour un jeu d'instances.

Les résultats reportés dans le Tableau 4.4 montrent que nos encodages de *MSCP* en un problème *weighted partial MaxSAT* couplés au solveur *ISAC* sont compétitifs. Les instances résolues par l'une des méthodes présentées dans la chapitre 2, le sont également par notre approche. Cependant, l'inverse n'est pas vrai, puisque notre approche a permis de résoudre l'instance *2-Insertions\_5*, que d'autres méthodes de la littérature ne sont pas capables de résoudre. En ce qui concerne les temps de résolution, les résultats sont globalement similaires aux meilleurs résultats connus. Toutefois, pour les instances *queen8\_8*, *4-FullIns\_4* et *5-FullIns\_4*, notre approche a permis de diminuer respectivement les temps de résolutions de 29286, 6572 et 48434 secondes à 17141, 633 et 4022 secondes.

## 4.5 Conclusions

À travers ce chapitre, nous avons présenté quatre encodages pour la transformation de *MSCP* en un problème *weighted partial MaxSAT* et en un problème *weighted partial MinSAT*. Afin de réduire la taille des instances générées, nous avons étudié la propriété proposée dans [Minot et al. \(2017\)](#). Comme nous avons pu le remarquer, le fait de considérer le degré du sommet d'un graphe a permis une réduction significative de la taille des instances sur les graphes de faible densité. Afin d'améliorer ces résultats sur tout type de graphe, nous avons proposé de renforcer cette propriété en considérant une borne supérieure de la force en plus du degré du sommet. En effet, si la borne supérieure de la force est inférieure au degré du sommet alors nous considérons cette dernière. Cette approche a permis sur certaines instances d'obtenir un taux de réduction supplémentaire, en termes de nombre de clauses, de plus de 80%, comparé à un codage considérant uniquement le degré d'un sommet.

La résolution des instances *weighted partial MaxSAT* par le solveur *ISAC* ([Ansótegui et al. \(2016\)](#)) a montré que ce formalisme était particulièrement adapté à la résolution de *MSCP*. Cependant, les résultats obtenus pour les instances *weighted partial MinSAT* par le solveur *MinSatz* ([Li et al. \(2011\)](#)) ne sont pas compétitifs. Afin d'améliorer ces résultats, il serait intéressant de mettre à jour le solveur *MinSatz* et d'orienter les techniques de résolution de ce dernier aux spécificités des instances de type *MSCP*.



## Chapitre 5

# Deux Approches de type Branch-and-Bound pour le Problème de la Somme Coloration Minimum d'un Graphe

### Sommaire

---

|  |           |
|--|-----------|
| <b>5.1 Branch-and-Bound</b> . . . . .  | <b>88</b> |
| 5.1.1 Séparation . . . . .   | 88        |
| 5.1.2 Évaluation . . . . .   | 88        |
| <b>5.2 BBMSCP</b> . . . . .  | <b>89</b> |
| 5.2.1 Algorithme général . . . . .   | 89        |
| 5.2.2 Évaluation : borne inférieure simple . . . . .                           | 90        |
| 5.2.3 Évaluation : borne inférieure par une décomposition en cliques . . . . . | 90        |
| 5.2.4 Expérimentations . . . . .   | 92        |
| <b>5.3 3LMSCP</b> . . . . .  | <b>93</b> |
| 5.3.1 Sous-ensemble de colorations majeures . . . . .                          | 93        |
| 5.3.2 Résultats expérimentaux . . . . .  | 94        |
| <b>5.4 Conclusions</b> . . . . .   | <b>97</b> |

---

Dans ce chapitre, nous présentons deux approches complètes *BBMSCP* et *3LMSCP* dédiées à la résolution de *MSCP*. Ces approches sont basées sur une méthode de résolution de type Branch-and-Bound. *BBMSCP* est une méthode énumérative dont l'efficacité repose principalement sur la qualité des bornes appelées  $LB_{B\Sigma}$  et  $LBP_{B\Sigma}$ . Ces bornes permettent l'exclusion de sous-ensembles de colorations de l'espace de recherche en faisant une estimation par défaut des solutions impliquées. Des bornes efficaces sont donc primordiales pour réduire de manière significative l'espace de recherche. *3LMSCP* est une méthode énumérative qui évite la redondance de traitement des colorations symétriques  $\Psi(X)$  d'une coloration  $X$ . Pour cela, *3LMSCP* considère uniquement les colorations majeures. Tout comme *BBMSCP*, *3LMSCP* exploite également une borne  $LB_{3\Sigma}$  afin de réduire l'espace de recherche.

Ce chapitre est organisé de la façon suivante. Dans la section 1, nous présentons la technique de Branch-and-Bound. Dans la section 2, nous introduisons et étudions expérimentalement notre solveur *BBMSCP* ainsi que les bornes  $LB_{B\Sigma}$  et  $LBP_{B\Sigma}$ . Dans la section 3, après avoir présenté notre solveur *3LMSCP*, nous étudions expérimentalement notre solveur. Dans la section 4, nous concluons ce chapitre.

Une partie de ces travaux a donné lieu aux publications suivantes : [Lecat et al. \(2015a,b\)](#).

## 5.1 Branch-and-Bound

Dans cette section, nous présentons le principe de fonctionnement d'une méthode de résolution se basant sur une technique de Branch-and-Bound. Une telle méthode a pour objectif de *séparer* l'espace de recherche et d'*évaluer* la pertinence des sous-ensembles générés.

### 5.1.1 Séparation

La phase de séparation a pour objectif de diviser l'espace de recherche en plusieurs sous-ensembles de solutions. Ainsi, la phase de séparation génère un arbre de recherche où chaque noeud de l'arbre représente un sous-ensemble de solutions. L'application récursive du principe de séparation sur chacun des sous-ensembles de solutions forme un arbre de recherche (Figure 5.2).

Dans le cas d'un problème de coloration, généralement, la séparation de l'espace de recherche se fait par l'attribution d'une couleur à un sommet du graphe. Par conséquent, un noeud de l'arbre de recherche représente un sommet du graphe et une branche représente l'affectation d'une couleur à ce sommet.

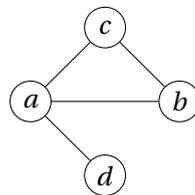


FIGURE 5.1 – Graphe  $G = (V, E)$ .

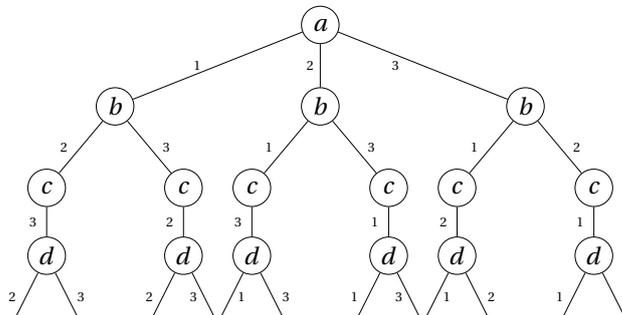


FIGURE 5.2 – Arbre de recherche associé à une méthode énumérative pour un problème de 3-coloration.

**Exemple 51** Soit  $G$  le graphe de la Figure 5.1. L'arbre de recherche associé à la résolution d'un problème de 3-coloration de  $G$  est donné par la Figure 5.2.

### 5.1.2 Évaluation

La phase d'évaluation permet de déterminer l'intérêt du sous-ensemble généré. Pour cela, une évaluation de chaque sous-ensemble de solutions est effectuée afin de déterminer si une solution optimale du problème peut être présente ou non dans ce sous-ensemble. Si l'évaluation prouve que tel n'est pas le cas, l'exploration du sous-ensemble est abandonnée.

Pour un problème de minimisation, les méthodes de résolution maintiennent à jour une borne supérieure globale  $UB$  et une borne inférieure locale  $LB$  durant l'exploration de l'arbre de recherche. La borne  $UB$  correspond à la meilleure solution rencontrée durant la recherche. Cette dernière est mise à jour si une branche complète de l'arbre de recherche a pu être développée. La borne  $LB$  correspond à une sous-estimation de la solution optimale d'un sous-ensemble à explorer. Si lors de l'évaluation d'une branche  $LB \geq UB$ , alors cette dernière est coupée.

## 5.2 BBMSCP

### 5.2.1 Algorithme général

Notre algorithme *BBMSCP* est un algorithme complet de type *Branch-and-Bound* dédié à la résolution de *MSCP*. Le processus d'exploration de *BBMSCP* est similaire à celui décrit précédemment. La phase de séparation consiste en la sélection d'un sommet  $v$  d'un graphe  $G$  et en l'assignation récursive de chacune des couleurs disponibles de  $v$ . La borne supérieure  $UB$  considérée par notre approche correspond à la meilleure somme coloration rencontrée durant l'exploration de l'espace des solutions. Cette dernière est mise à jour quand une coloration  $X$ , telle que  $UB \geq \Sigma(X)$ , est générée. La borne inférieure  $LB$  utilisée par *BBMSCP* est une sous-estimation de la somme coloration des colorations possibles pour  $G_c \cup G_{\bar{c}}$ , où  $G_c$  est le sous-graphe de  $G$  constitué des sommets colorés et  $G_{\bar{c}}$  est le sous-graphe de  $G$  constitué des sommets non colorés. Si lors de la phase d'évaluation  $LB \geq UB$  alors il n'est pas pertinent de continuer la recherche dans cette partie de l'espace des solutions. Dans le cas contraire, le processus est à nouveau appliqué.

L'Algorithme 11 présente en détail notre approche. Il prend en paramètre un sous-graphe  $G_{\bar{c}}$  de  $G$  constitué des sommets non colorés, la somme coloration  $\Sigma_S$  associée au sous-graphe  $G_c$  de  $G$  composé des sommets colorés et  $UB_{\Sigma}$  une borne supérieure de la somme chromatique du graphe  $G$ . *BBMSCP* retourne la somme chromatique associée au graphe donné en paramètre. Notre méthode commence par vérifier si le graphe  $G_{\bar{c}}(V_{\bar{c}}, E_{\bar{c}})$  comporte des sommets, c'est-à-dire si  $V_{\bar{c}} \neq \emptyset$  (Ligne 2). Dans le cas où  $V_{\bar{c}} = \emptyset$ , cela signifie qu'une couleur a été assignée à chaque sommet du graphe. Par conséquent, la somme coloration associé au graphe  $G_c$  est la meilleure rencontrée à ce moment de la recherche. Sinon, si  $V_{\bar{c}} \neq \emptyset$  alors un sommet  $v$  du graphe  $G_{\bar{c}}$  est sélectionné. Pour chaque couleur  $c$  disponible de  $v$ , nous assignons  $c$  à  $v$  afin de générer le sous-graphe  $G_{\bar{c}} \setminus \{v\}$ . Ensuite, une sous-estimation de la meilleure somme coloration de  $G_{\bar{c}} \setminus \{v\} \cup G_c \cup \{v\}$  est calculée (Ligne 6). Nous remarquons que le premier appel de notre algorithme est  $BBMSCP(G, \Sigma(X), \Sigma(X))$  où  $X$  est une coloration  $G$  obtenue à l'aide d'un algorithme glouton tel que *MDSAT*, *MRLF* (Li *et al.* (2009)).

---

**Algorithme 11 :** Méthode de fonctionnement de  $BBMSCP(G, \Sigma_S, UB_{\Sigma})$ .

---

**Entrées :**  $G_{\bar{c}} = (V_{\bar{c}}, E_{\bar{c}})$  un sous-graphe de  $G$  constitué des sommets non colorés,  $\Sigma_S$  la somme coloration associée sous-graphe  $G_c$  de  $G$  constitué des sommets colorés et  $UB_{\Sigma}$  la meilleure somme coloration rencontrée durant la recherche.

**Sortie :**  $\Sigma(G)$  la somme chromatique.

```

1  début
2  si  $V_{\bar{c}} = \emptyset$  alors
3  | retourner  $UB_{\Sigma}$ ;
4  Choisir un sommet  $v$  de  $V_{\bar{c}}$ ;
5  pour chaque couleur  $c$  disponible pour  $v$  faire
6  | si  $LB_{\Sigma}(G \setminus v) + c + \Sigma_S < UB_{\Sigma}$  alors
7  | | pour chaque sommet  $u$  voisin de  $v$  faire
8  | | | Supprimer  $c$  des couleurs disponibles pour  $u$ ;
9  | | |  $UB_{\Sigma} \leftarrow BBMSCP(G_{\bar{c}} \setminus v, \Sigma_S + c, UB_{\Sigma})$ ;
10 | | | pour chaque sommet  $u$  voisin de  $v$  faire
11 | | | | Ajouter  $c$  aux couleurs disponibles pour  $u$ ;
12 | retourner  $UB_{\Sigma}$ ;

```

---

Comme nous le remarquons dans l'Algorithme 11, la phase d'évaluation est un élément central de notre méthode. En effet, une borne de bonne qualité permet l'abandon rapide des sous-ensembles de solutions non pertinents. Dans la section suivante, nous proposons deux méthodes permettant de calculer une borne inférieure.

### 5.2.2 Évaluation : borne inférieure simple

Le principe de la première borne inférieure que nous proposons repose sur l'impact des sommets colorés sur les sommets non colorés. Chaque sommet  $v$  de  $G_{\bar{c}}$  dispose d'une plus petite couleur disponible, noté  $\min_c(v)$ , c'est-à-dire, pour chaque sommet  $u$  voisin de  $v$  tel que  $u \in G_c$ ,  $c(u) \neq \min_c(v)$ . Par conséquent, la somme des couleurs des sommets de  $G_c$  additionnée à la somme des plus petites couleurs des sommets de  $G_{\bar{c}}$  est une borne inférieure pour la somme chromatique du sous-ensemble de solutions considéré. Nous notons cette borne  $LB_{B\Sigma}$ . La valeur de cette borne est donnée par l'Équation 5.1 et calculée par l'Algorithme 12.

$$LB_{B\Sigma}(G_{\bar{c}}) = \sum_{v \in G_c} c(v) + \sum_{v \in G_{\bar{c}}} \min_c(v) \quad (5.1)$$

---

**Algorithme 12 :**  $LB_{B\Sigma}(G)$ .

---

**Entrée :**  $G$  un graphe.

**Sortie :**  $lb$  une borne inférieure de la somme coloration du graphe  $G$  en cours de coloration.

```

1 début
2    $lb \leftarrow 0$ ;
3   pour chaque sommet  $v$  de  $G$  faire
4     si  $v \in G_c$  alors
5        $lb \leftarrow lb + c(v)$ ;
6     sinon
7        $lb \leftarrow lb + \min_c(v)$ ;
8   retourner  $lb$ ;

```

---

**Exemple 52** Soit le graphe  $G$  de la Figure 5.3 tel que le sommet  $b$  est coloré avec la couleur 2. Pour  $k = 3$ ,  $LB_{B\Sigma}(G_{\bar{c}}) = 2 + 1 + 1 + 1 = 5$ .

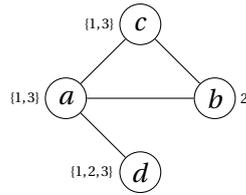


FIGURE 5.3 – Modélisation des couleurs disponibles d'un graphe  $G = (V, E)$  pour  $k = 3$  tel que  $c(b) = 2$ .

L'intérêt de  $LB_{B\Sigma}$  est sa faible complexité en  $O(|V|)$ . Cependant, la principale faiblesse de cette approche est la relaxation de l'ensemble des contraintes de voisinage entre les sommets non colorés.

### 5.2.3 Évaluation : borne inférieure par une décomposition en cliques

La seconde borne que nous proposons préserve d'avantage les contraintes de voisinage du problème. Pour cela, nous décomposons le sous-graphe  $G_{\bar{c}}$  d'un graphe  $G$  en une partition de cliques  $P_1, P_2, \dots, P_n$ , tel que  $\forall i, j \leq n$ ,  $P_i \cap P_j = \emptyset$ . Comme vu dans le Chapitre 3.4.2, plusieurs méthodes de la littérature, telles que [Wu et Hao \(2013b\)](#); [Moukrim et al. \(2014\)](#), ont développé une approche similaire afin d'extraire une borne inférieure de la somme chromatique d'un graphe (Propriété 21). Comme précisé dans la Propriété 21, le calcul de la somme chromatique d'une clique  $P_i$  est  $\frac{|P_i| \times (|P_i| + 1)}{2}$ . Cela signifie que les sommets de la clique sont assignés aux couleurs allant de 1 à  $|P_i|$ . Or, la méthode que nous proposons s'applique aux sommets du sous-graphe  $G_{\bar{c}}$  pour

lesquels certains sommets voisins sont colorés. Pour tenir compte de ces contraintes de voisinage, notre borne extrait les  $|P_i|$  plus petites couleurs disponibles d'une clique  $P_i$  pour déterminer la somme coloration associée à  $P_i$ . Cette borne que nous appelons  $LBP_{B\Sigma}$  est présentée en détail dans l'Algorithme 13.  $LBP_{B\Sigma}$  prend en paramètre un graphe et effectue une décomposition en cliques de ce dernier (Ligne 3). Quand la décomposition est obtenue, l'ensemble des couleurs disponibles d'une clique  $P_i$  est stocké dans un ensemble  $C_i$  (Ligne 5). Ainsi, la somme des  $|P_i|$  plus petites couleurs de  $C_i$  représentent une borne inférieure de la somme chromatique de la clique  $P_i$ .

---

**Algorithme 13 :**  $LBP_{B\Sigma}(G)$ .

---

**Entrée :**  $G = (V, E)$  un graphe.  
**Sortie :**  $lb$  une borne inférieure d'une somme coloration.

```

1 début
2    $lb \leftarrow 0$ ;
3    $P \leftarrow CliquePart(G)$ ;
4   pour chaque clique  $P_i$  de  $P$  faire
5      $C_i \leftarrow \bigcup_{v \in P_i} \{c \mid c \text{ est l'ensemble des couleurs disponibles de } v \text{ et } c \notin C_i\}$ ;
6     pour  $j$  allant de 1 à  $|P_i|$  faire
7        $c_{min} \leftarrow \min(C_i)$ ;
8        $lb \leftarrow lb + c_{min}$ ;
9        $C_i \leftarrow C_i \setminus \{c_{min}\}$ ;
10  retourner  $lb$ ;

```

---

L'algorithme *CliquePart* (Algorithme 14) présente une méthode basique de partitionnement en cliques d'un graphe. Cette méthode gloutonne sélectionne les sommets à traiter suivant un ordre fixe (Ligne 4). Après avoir expérimenté différents critères de sélection, le choix que nous avons sélectionné est un ordonnancement croissant selon le degré des sommets. La complexité de cet algorithme est en  $O(|V|^2)$ .

---

**Algorithme 14 :**  $CliquePart(G)$ .

---

**Entrée :**  $G = (V, E)$  un graphe.  
**Sortie :** une partition de cliques  $P$ .

```

1 début
2    $P \leftarrow \{\emptyset\}$ ;
3   tant que  $V$  n'est pas vide faire
4     Sélectionner un sommet  $v$  de  $V$ ;
5      $V \leftarrow V \setminus \{v\}$ ;
6     si  $\exists P_i \in P$  tel que  $\forall u \in P_i, (u, v) \in E$  alors
7        $P_i \leftarrow P_i \cup \{v\}$ ;
8     sinon
9        $P_{new} \leftarrow \{v\}$ ;
10       $P \leftarrow P \cup \{P_{new}\}$ ;
11  retourner  $P$ ;

```

---

**Exemple 53** Soit le graphe  $G$  de la Figure 5.3 tel que le sommet  $b$  est coloré avec la couleur 2 et la décomposition en cliques de ce dernier suivant une approche *CliquePart* modélisée dans la Figure 5.4. Pour  $k = 3$ ,  $C_a = C_c = \{1, 3\}$  et  $C_d = \{1\}$ . Par conséquent,  $LBP_{B\Sigma}(G) = c(b) + LB_{\Sigma}(\{a, c\}) + LB_{\Sigma}(\{d\}) = 2 + 4 + 1 = 7$ .

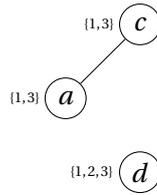


FIGURE 5.4 – Modélisation d’une décomposition en cliques de la Figure 5.3 après coloration du sommet  $b$  suivant une approche *CliquePart*.

## 5.2.4 Expérimentations

Dans cette section, nous comparons expérimentalement notre solveur *BBMSCP* avec les résultats obtenus par le solveur *ISAC* (Ansótegui *et al.* (2016)). L’ensemble des expérimentations a été mené sur la plateforme MatriCS équipée de processeurs Intel Xeon CPUs E5-2680 v4@2.40GHz sous Linux avec 128GB de mémoire.

Le Tableau 5.1 reporte les résultats obtenus par notre solveur *BBMSCP* sur quelques instances *COLOR* (**COLOR**) et *DIMACS* (**Dimacs**). Dans cette expérimentation, nous avons fixé la limite de temps à 4 heures. Pour chaque graphe  $G$ , sont représentés dans les différentes colonnes, le nombre de sommets  $|V|$ , la densité du graphe  $d$ , les temps en secondes pour la détermination de la somme coloration pour le solveur *BBMSCP* suivant l’utilisation de la borne  $LB_{B\Sigma}$  et  $LBP_{B\Sigma}$ , et les temps en secondes pour la détermination de la somme coloration pour le solveur *ISAC* suivant la considération de l’encodage 1 et l’encodage 2 présenté dans le Chapitre 4.1.2 (Définitions 106 et 107). Nous notons que seules les instances résolues par notre solveur *BBMSCP* ont été reportées dans le Tableau 5.1.

| Graphes  | $ V $ | $d$   | BBMSCP         |                 | ISAC       |            |
|----------|-------|-------|----------------|-----------------|------------|------------|
|          |       |       | $LB_{B\Sigma}$ | $LBP_{B\Sigma}$ | Encodage 1 | Encodage 2 |
| queen5_5 | 25    | 0.533 | 0              | 0               | 0          | 0          |
| myciel3  | 11    | 0.363 | 0              | 0               | 0          | 0          |
| myciel4  | 23    | 0.280 | 0              | 0               | 0          | 0          |
| myciel5  | 47    | 0.218 | 7834           | 1663            | 1          | 0          |

TABLEAU 5.1 – Comparaison des temps de résolution en secondes du solveur *BBMSCP* et du solveur *ISAC* pour les instances *COLOR* (**COLOR**) et *DIMACS* (**Dimacs**).

Les résultats reportés dans le Tableau 5.1 montrent la difficulté d’une résolution de type Branch-and-Bound pour la résolution du problème *MSCP*. En effet, parmi l’ensemble des instances testées, seulement 4 instances ont pu être résolues.

Le Tableau 5.2 reporte les résultats obtenus par notre solveur *BBMSCP* sur l’ensemble de benchmarks aléatoires. Dans cette expérimentation, nous avons fixé la limite de temps à 4 heures pour chaque solveur et pour chaque instance. Pour chaque graphe  $m\_d$  où  $n$  représente le nombre de sommets du graphe et  $d$  sa densité, 10 graphes ont été générés. Nous reportons dans les différentes colonnes, le temps moyen en secondes pour la détermination de la somme coloration pour le solveur *BBMSCP* suivant l’utilisation de la borne  $LB_{B\Sigma}$  et  $LBP_{B\Sigma}$ , et le temps moyen en secondes pour la détermination de la somme coloration pour le solveur *ISAC* suivant la considération de l’encodage 1 et l’encodage 2. Entre parenthèses est reporté le nombre d’instances résolues pour

les différentes méthodes. Les solveurs ayant résolu le plus d'instances pour une même famille sont colorés. Parmi ces solveurs, ceux possédant le plus petit temps moyen de résolution sont considérés comme meilleurs et colorés en foncés.

| Graphes | BBMSCP         |                 | MaxSAT      |            |
|---------|----------------|-----------------|-------------|------------|
|         | $LB_{B\Sigma}$ | $LBP_{B\Sigma}$ | Encodage 1  | Encodage 2 |
| r30_0.1 | 1.60(10)       | 0.10(10)        | 0(10)       | 0(10)      |
| r30_0.5 | 3704(2)        | 5582(6)         | 5.30(10)    | 5(10)      |
| r30_0.9 | N/A            | 2030(1)         | 2.90(10)    | 1.50(10)   |
| r35_0.1 | 99.20(10)      | 6.90(10)        | 0(10)       | 0(10)      |
| r35_0.5 | N/A            | N/A             | 47.20(10)   | 48.10(10)  |
| r35_0.9 | N/A            | N/A             | 569.33(9)   | 83.30(10)  |
| r40_0.1 | 2051.50(8)     | 1028.10(10)     | 0(10)       | 0(10)      |
| r40_0.5 | N/A            | N/A             | 1024.10(10) | 981.40(10) |
| r40_0.9 | N/A            | N/A             | 679(4)      | 381.60(5)  |
| r45_0.1 | 324.25(4)      | 2378.57(7)      | 0(10)       | 0(10)      |
| r45_0.5 | N/A            | N/A             | 3151.25(8)  | 2699.62(8) |
| r45_0.9 | N/A            | N/A             | 464.75(4)   | 205.33(6)  |
| r50_0.1 | 2668.50(2)     | 612(4)          | 0.20(10)    | 0.70(10)   |
| r50_0.5 | N/A            | N/A             | N/A         | N/A        |
| r50_0.9 | N/A            | N/A             | N/A         | 1465(2)    |
| r55_0.1 | N/A            | 4684.25(4)      | 3.20(10)    | 3.10(10)   |
| r55_0.5 | N/A            | N/A             | N/A         | N/A        |
| r55_0.9 | N/A            | N/A             | N/A         | 543(1)     |
| r60_0.1 | N/A            | 3496(1)         | 12.20(10)   | 13.90(10)  |
| r60_0.5 | N/A            | N/A             | N/A         | N/A        |
| r60_0.9 | N/A            | N/A             | N/A         | N/A        |

TABLEAU 5.2 – Comparaison du temps moyen de résolution en secondes du solveur *BBMSCP* et du solveur *ISAC* pour un ensemble d'instances aléatoires.

Tout comme le Tableau 5.1, le Tableau 5.2 montre la difficulté d'une approche exacte pour les problèmes *NP*-difficiles comme *MSCP*. Cependant, nous pouvons observer que la borne considérée a un impact sur la résolution de *MSCP*. En effet, la borne  $LBP_{B\Sigma}$  a résolu 53 instances tandis que la borne  $LB_{B\Sigma}$  a résolu 36 instances. L'une des raisons de la difficulté de résolution est le grand nombre de symétries rencontrées durant l'exploration de l'espace de recherche. Afin de réduire l'énumération des colorations symétriques, nous proposons dans la section suivante le solveur *3LMSCP*.

## 5.3 3LMSCP

### 5.3.1 Sous-ensemble de colorations majeures

Comme nous l'avons remarqué dans le Chapitre 3.1, pour une coloration  $X$  d'un graphe, il existe un ensemble de colorations  $\Psi(X)$  symétriques résultant de la permutation des classes couleurs de  $X$  (Définition 93). Cependant, si nous considérons l'espace de recherche développé par notre solveur *BBMSCP* (Figure 5.2) alors nous constatons que ce dernier n'est pas en mesure de détecter deux colorations symétriques telles que les colorations  $\{\{a\}_1, \{b, d\}_2, \{c\}_3\}$  et  $\{\{b, d\}_1, \{a\}_2, \{c\}_3\}$ . De ce fait, nous proposons une nouvelle approche pour la résolution de *MSCP*, appelée *3LMSCP*, se basant sur la notion de coloration majeure (Définition 94). L'objectif de notre approche est de parcourir l'espace de recherche afin de ne considérer que les colorations majeures. L'intérêt de cette approche est de réduire l'espace de recherche de manière significative en éliminant l'ensemble des colorations symétriques de  $\Psi(X)$  par la représentation d'une coloration  $X$  sous forme

de partition en un ensemble de stables. Cette approche est similaire à celle étudiée dans Mehrotra et Trick (1996); Gualandi et Malucelli (2012) pour le problème de coloration de graphe. La Figure 5.5 illustre l'arbre de recherche associé à ce type de méthode.

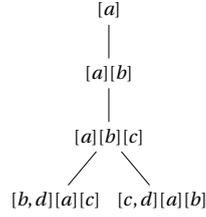


FIGURE 5.5 – Arbre de recherche associé à 3LMSCP pour un problème de 3-coloration (Figure 5.1).

Le principe de 3LMSCP est d'énumérer l'ensemble des colorations majeures valides du graphe. Pour cela, 3LMSCP maintient à chaque noeud de l'arbre de recherche une partition en un ensemble de stables  $P = P_1, P_2, \dots, P_n$  du sous-graphe  $G_c$ . La phase de séparation de 3LMSCP consiste en la sélection d'un sommet  $v$  du sous-graphe  $G_{\bar{c}}$  et en la création d'une branche pour chaque stable  $P_i$  de  $P$  tel que  $P_i \cup \{v\}$  forme un stable. De plus, si  $n \leq UB_S(G)$ , où  $UB_S(G)$  est une borne supérieure de la force, une branche est créée où le stable singleton  $\{v\}$  est ajouté à la partition  $P$ . Par conséquent, chaque chemin de la racine à une feuille du graphe forme une partition en un ensemble de stables du graphe  $G$ . Comme montré dans la Figure 5.5, 3LMSCP développe 2 colorations alors que BBMSCP en considère 12 (Figure 5.2).

L'Algorithme 12 présente le fonctionnement de 3LMSCP. Les paramètres sont un graphe  $G$ ,  $G_{\bar{c}}$  un sous-graphe de  $G$  composé des sommets non colorés,  $P$  un partitionnement en stable de  $G_c$ ,  $UB_{\Sigma}$  la meilleure somme coloration rencontrée et  $UB_{\alpha(G)}$  une borne supérieure de la cardinalité du stable maximum de  $G$ . Le premier appel de 3LMSCP est  $3LMSCP(G, G, \{\emptyset\}, \Sigma(X), UB_{\alpha(G)})$  où  $X$  est une coloration valide obtenue à l'aide d'un algorithme glouton. Si le graphe  $G_{\bar{c}}$  ne contient pas de sommets (Ligne 2), cela signifie que tous les sommets du graphe ont été traités. Une feuille de l'arbre de recherche de 3LMSCP a donc été atteinte. L'algorithme retourne la somme coloration associée au partitionnement de  $P$ . Pour cela, il est cependant nécessaire de réorganiser  $P$  par ordre décroissant des cardinalités des stables. Si tous les sommets de  $G$  n'ont pas été considérés une borne inférieure de la somme coloration de l'état courant du graphe est calculée (Ligne 5). Dans le cas où la borne inférieure est supérieure à la meilleure somme coloration connue, la recherche est stoppée. Sinon, un sommet  $v$  de  $G_{\bar{c}}$  est sélectionné et 3LMSCP est appelé de façon récursive.

La borne inférieure  $LB_{\Sigma}$  (Ligne 5) considérée par notre algorithme pour la phase d'évaluation est présentée en détail dans l'Algorithme 16 ( $LB_{3\Sigma}$ ). Afin d'obtenir une coloration majeure partielle,  $LB_{3\Sigma}$  commence par ordonner les stables  $P_i$  de  $P$  par ordre décroissant de leur cardinalité. Puis, pour chaque sommet  $v$  de  $G_{\bar{c}}$ ,  $v$  est inséré dans le premier stable  $P_i$  tel que  $|P_i| < \alpha(G)$ . Si un tel stable ne peut être déterminé dans  $P$  alors le singleton  $\{v\}$  est ajouté à  $P$ . Nous remarquons que  $LB_{3\Sigma}$  ne considère aucune contrainte de voisinages pour les sommets de  $G_{\bar{c}}$ . La somme coloration associée à la coloration majeure construite par  $LB_{3\Sigma}$  est une sous-estimation de  $G_c \cup G_{\bar{c}}$ .

**Exemple 54** Soit le graphe  $G$  de la Figure 5.1 tel que le sommet  $b$  est coloré ( $P = [b]_1$ ) et  $UB_{\alpha(G)} = 2$ . La coloration majeure construite par  $LB_{3\Sigma}$  est  $P = [b, a]_1 [c, d]_2$ . Par conséquent,  $LB_{3\Sigma} = 2 \times 1 + 2 \times 2 = 6$ .

### 5.3.2 Résultats expérimentaux

Dans cette section, nous comparons expérimentalement notre solveur 3LMSCP avec les résultats obtenus par le solveur ISAC. L'ensemble des expérimentations a été mené sur la plateforme MatriCS équipée de processeurs Intel Xeon CPUs E5-2680 v4@2.40GHz sous Linux avec 128GB de mémoire.

Le Tableau 5.3 reporte les résultats obtenus sur quelques instances COLOR (COLOR) et DIMACS (Dimacs) par notre solveur 3LMSCP. Dans cette expérimentation, nous avons fixé la limite

**Algorithme 15 :**  $3LMSCP(G, P, UB_{\Sigma}, UB_{\alpha(G)})$ .

**Entrées :**  $G = (V, E)$  un graphe,  $G_{\bar{c}}$  un sous-graphe de  $G$  composé des sommets non colorés,  $P = \{P_1, P_2, \dots, P_{|P|}\}$  une décomposition en stable du sous-graphe  $G_{\bar{c}}$  de  $G$  composé des sommets colorés,  $UB_{\Sigma}$  la meilleure somme coloration rencontrée durant la recherche et  $UB_{\alpha(G)}$  une borne supérieure de la cardinalité du stable maximum de  $G$ .

**Sortie :**  $\Sigma(G)$  la somme chromatique.

```

1 début
2   si  $G_{\bar{c}} = \emptyset$  alors
3     Trier  $P$  par ordre décroissant des  $|P_i|$ ;
4     retourner  $LB_{\Sigma}(G_{\bar{c}}, P, UB_{\alpha(G)})$ ;
5   si  $LB_{\Sigma}(G_{\bar{c}}, P, UB_{\alpha(G)}) \geq UB_{\Sigma}$  alors
6     retourner  $UB_{\Sigma}$ ;
7   Sélectionner un sommet  $v$  de  $G_{\bar{c}}$ ;
8   pour chaque  $P_i \in P$  tel que  $\mathcal{N}(v) \cap P_i = \emptyset$  et  $|P_i| < UB_{\alpha(G)}$  faire
9      $UB_{\Sigma} \leftarrow 3LMSCP(G, G_{\bar{c}} \setminus \{v\}, \{P_1, \dots, P_{i-1}, P_i \cup \{v\}, P_{i+1}, \dots, P_{|P|}\}, UB_{\Sigma}, UB_{\alpha(G)})$ ;
10  si  $|P| < s(G)$  alors
11     $UB_{\Sigma} \leftarrow 3LMSCP(G, G_{\bar{c}} \setminus \{v\}, \{P_1, P_2, \dots, P_{|P|}, \{v\}\}, UB_{\Sigma}, UB_{\alpha(G)})$ ;
12  retourner  $UB_{\Sigma}$ ;

```

de temps à 4 heures pour chaque solveur et pour chaque instance. Pour chaque graphe  $G$ , sont représentés dans les différentes colonnes, le nombre de sommets  $|V|$ , la densité du graphe  $d$ , les temps en secondes pour la détermination de la somme coloration par le solveur  $3LMSCP$ , les temps en secondes pour la détermination de la somme coloration par le solveur  $BBMSCP$  suivant l'utilisation de la borne  $LB_{B\Sigma}$  et  $LBP_{B\Sigma}$ , et le temps en secondes pour la détermination de la somme coloration par le solveur  $ISAC$  suivant la considération de l'encodage 1 et l'encodage 2. Nous notons que seules les instances résolues par notre solveur  $3LMSCP$  ont été reportées dans le Tableau 5.3.

| Graphes   | $ V $ | $d$   | 3LMSCP         | BBMSCP         |                 | ISAC       |            |
|-----------|-------|-------|----------------|----------------|-----------------|------------|------------|
|           |       |       | $LB_{3\Sigma}$ | $LB_{B\Sigma}$ | $LBP_{B\Sigma}$ | Encodage 1 | Encodage 2 |
| le450_5c  | 450   | 0.097 | <b>0</b>       | N/A            | N/A             | 111        | 164        |
| le450_5d  | 450   | 0.096 | <b>0</b>       | N/A            | N/A             | 123        | 90         |
| queen5_5  | 25    | 0.533 | 0              | 0              | 0               | 0          | 0          |
| queen6_6  | 36    | 0.460 | 937            | N/A            | N/A             | <b>205</b> | 400        |
| queen7_7  | 49    | 0.404 | 0              | N/A            | N/A             | 0          | 0          |
| queen8_8  | 64    | 0.361 | <b>11406</b>   | N/A            | N/A             | N/A        | N/A        |
| queen8_12 | 96    | 0.300 | <b>0</b>       | N/A            | N/A             | 1          | 1          |
| myciel3   | 11    | 0.363 | 0              | 0              | 0               | 0          | 0          |
| myciel4   | 23    | 0.280 | 0              | 0              | 0               | 0          | 0          |
| myciel5   | 47    | 0.218 | 36             | 7834           | 1663            | 1          | <b>0</b>   |

TABLEAU 5.3 – Comparaison des temps de résolution en secondes du solveur  $3LMSCP$ , du solveur  $BBMSCP$  et du solveur  $ISAC$  pour les instances *COLOR* (COLOR) et *DIMACS* (Dimacs).

Le Tableau 5.3 montre que  $3LMSCP$  a permis de diminuer fortement les temps de résolutions des instances  $le450_c$  et  $le450_d$ . De plus, nous remarquons que seul  $3LMSCP$  a pu résoudre l'instance  $queen8_8$  en moins de 4 heures. Cependant, le nombre d'instances résolues reste très faible.

Le Tableau 5.4 reporte les résultats obtenus par notre solveur  $3LMSCP$  sur l'ensemble des benchmarks aléatoires. Dans cette expérimentation, nous avons fixé la limite de temps à 4 heures. Pour chaque graphe  $rn_d$  où  $n$  représente le nombre de sommets du graphe et  $d$  sa densité, 10

---

**Algorithme 16 :**  $LB_{3\Sigma}(G_{\bar{c}}, P, UB_{\alpha(G)})$ .

---

**Entrées :**  $G_{\bar{c}}$  un sous-graphe de  $G$  composé des sommets non colorés,  $P = P_1, \dots, P_n$  un partitionnement en clique du sous-graphe  $G_c$  de  $G$  composé des sommets colorés et  $UB_{\alpha(G)}$  une borne supérieure de la cardinalité du stable maximum de  $G$ .

**Sortie :**  $lb$  une borne inférieure de la somme coloration du graphe  $G$  en cours de coloration.

```

1 début
2    $lb \leftarrow 0$ ;
3   Ordonner  $P$  par ordre décroissant des  $|P_i|$ ;
4   pour chaque sommet  $v$  de  $G_{\bar{c}}$  faire
5     si  $\exists P_i \in P$  tel que  $|P_i| < UB_{\alpha(G)}$  et  $\forall j < i, |P_j| = UB_{\alpha(G)}$  alors
6        $P_i \leftarrow P_i \cup \{v\}$ ;
7     sinon
8        $P \leftarrow P \cup \{v\}$ ;
9   pour chaque  $P_i$  de  $P$  faire
10     $lb \leftarrow lb + i \times |P_i|$ ;
11  retourner  $lb$ ;

```

---

graphes ont été générés. Nous reportons dans les différentes colonnes, le temps moyen en secondes pour la détermination de la somme coloration par le solveur *3LMSCP*, les temps en secondes pour la détermination de la somme coloration par le solveur *BBMSCP* suivant l'utilisation de la borne  $LB_{B\Sigma}$  et  $LBP_{B\Sigma}$ , et le temps moyen en secondes pour la détermination de la somme coloration par le solveur *ISAC* suivant la considération de l'encodage 1 et l'encodage 2. Entre parenthèses est reporté le nombre d'instances résolues pour les différentes méthodes. Les solveurs ayant résolu le plus d'instances pour une même famille sont colorés. Parmi ces solveurs, ceux possédant le plus petit temps moyen de résolution sont considérés comme meilleurs et colorés en foncés.

| Graphes | 3LMSCP         | BBMSCP         |                 | ISAC        |            |
|---------|----------------|----------------|-----------------|-------------|------------|
|         | $LB_{3\Sigma}$ | $LB_{B\Sigma}$ | $LBP_{B\Sigma}$ | Encodage 1  | Encodage 2 |
| r30_0.1 | 0(10)          | 1.60(10)       | 0.10(10)        | 0(10)       | 0(10)      |
| r30_0.5 | 1(10)          | 3704(2)        | 5582(6)         | 5.30(10)    | 5(10)      |
| r30_0.9 | 0(10)          | N/A            | 2030(1)         | 2.90(10)    | 1.50(10)   |
| r35_0.1 | 0(10)          | 99.20(10)      | 6.90(10)        | 0(10)       | 0(10)      |
| r35_0.5 | 53(10)         | N/A            | N/A             | 47.20(10)   | 48.10(10)  |
| r35_0.9 | 0(10)          | N/A            | N/A             | 569.33(9)   | 83.30(10)  |
| r40_0.1 | 1.40(10)       | 2051.50(8)     | 1028.10(10)     | 0(10)       | 0(10)      |
| r40_0.5 | 995.30(10)     | N/A            | N/A             | 1024.10(10) | 981.40(10) |
| r40_0.9 | 1(10)          | N/A            | N/A             | 679(4)      | 381.60(5)  |
| r45_0.1 | 22.20(10)      | 324.25(4)      | 2378.57(7)      | 0(10)       | 0(10)      |
| r45_0.5 | 9326(4)        | N/A            | N/A             | 3151.25(8)  | 2699.62(8) |
| r45_0.9 | 28.60(10)      | N/A            | N/A             | 464.75(4)   | 205.33(6)  |
| r50_0.1 | 1050.70(10)    | 2668.50(2)     | 612(4)          | 0.20(10)    | 0.70(10)   |
| r50_0.5 | 4004(1)        | N/A            | N/A             | N/A         | N/A        |
| r50_0.9 | 297(10)        | N/A            | N/A             | N/A         | 1465(2)    |
| r55_0.1 | 3740.11(9)     | N/A            | 4684.25(4)      | 3.20(10)    | 3.10(10)   |
| r55_0.5 | N/A            | N/A            | N/A             | N/A         | N/A        |
| r55_0.9 | 529.62(8)      | N/A            | N/A             | N/A         | 543(1)     |
| r60_0.1 | 719.75(4)      | N/A            | 3496(1)         | 12.20(10)   | 13.90(10)  |
| r60_0.5 | N/A            | N/A            | N/A             | N/A         | N/A        |

Continue sur la page suivante...

|         | 3LMSCP         | BBMSCP         |                 | ISAC       |            |
|---------|----------------|----------------|-----------------|------------|------------|
| Graphes | $LB_{3\Sigma}$ | $LB_{B\Sigma}$ | $LBP_{B\Sigma}$ | Encodage 1 | Encodage 2 |
| r60_0.9 | 4054.20(5)     | N/A            | N/A             | N/A        | N/A        |

TABLEAU 5.4 – Comparaison du temps moyen de résolution en secondes du solveur 3LMSCP, du solveur BBMSCP et du solveur ISAC pour un ensemble d'instances aléatoires.

Les résultats reportés dans le Tableau 5.4 montrent que notre approche obtient des résultats compétitifs face au solveur ISAC sur l'ensemble de graphes aléatoires. En effet, lorsque la densité est faible, une approche de type *MaxSAT* permet une meilleure résolution du problème. Cependant, lorsque la densité augmente une approche de type 3LMSCP est appropriée. De plus, nous remarquons que notre approche a permis la résolution de 161 instances contre 139 pour l'encodage 1 et 146 pour l'encodage 2 résolus par ISAC. Toutefois, 3LMSCP est efficace uniquement sur des graphes de petites tailles. L'une des raisons est le temps de calculs nécessaire à la réorganisation des classes couleurs afin de pouvoir calculer la somme coloration associée.

## 5.4 Conclusions

Dans ce chapitre, nous avons présenté deux nouveaux algorithmes basés sur une méthode de Branch-and-Bound, appelés BBMSCP et 3LMSCP, pour la résolution exacte de MSCP. Nous avons également présenté deux bornes  $LB_{B\Sigma}$  et  $LBP_{B\Sigma}$  pour BBMSCP, afin de sous-estimer la somme coloration durant la phase d'évaluation. Le principe de  $LB_{B\Sigma}$  repose sur l'impact des sommets colorés sur les sommets non colorés, en considérant les plus petites couleurs disponibles. Cependant, une telle approche relâche trop de contraintes de voisinage. Le principe de la borne  $LBP_{B\Sigma}$  est de considérer un partitionnement du sous-graphe composé des sommets non colorés en une partition de cliques afin de calculer à partir de celle-ci une sous-estimation de la meilleure somme coloration du sous-ensemble de solutions associé.

L'approche 3LMSCP est différente de celle proposée par BBMSCP. En effet, un noeud de l'arbre de recherche développé par BBMSCP correspond à un sommet du graphe et une branche à l'assignation du sommet par une couleur. Pour 3LMSCP, un noeud de l'arbre de recherche correspond à une partition du sous-graphe composé des sommets déjà traités en un ensemble de stables et une branche à l'assignation d'un sommet non traité à un stable de la partition en cours. L'intérêt d'une telle approche est qu'elle permet la suppression d'un ensemble de colorations symétriques en considérant uniquement les colorations majeures.

Les résultats expérimentaux ont montré que notre approche BBMSCP n'était pas compétitive dans sa version actuelle alors que les résultats de 3LMSCP sont prometteurs. Ces derniers ont montré sur de petits graphes aléatoires de densité allant 0.5 à 0.9 que 3LMSCP est plus efficace qu'une approche utilisant le paradigme *weighted partial MaxSAT*. Mais, pour tout graphe de plus de 60 sommets, 3LMSCP n'est plus compétitif. L'une des raisons à cela est la mauvaise qualité de la borne  $LB_{3\Sigma}$  considérée pendant la phase d'évaluation de 3LMSCP. Améliorer ces bornes est une perspective majeure pour espérer traiter plus d'instances.



# Conclusion Générale

Le Problème de la Somme Coloration Minimum (*MSCP*) d'un graphe est un problème complexe. Comme nous avons pu le remarquer, la force du graphe peut être très éloignée du nombre chromatique. Par conséquent, l'espace de recherche associé à *MSCP* est plus compliqué que celui d'un Problème de Coloration de Graphe (*GCP*).

Le premier objectif des travaux de cette thèse a été d'étudier l'espace de recherche de *MSCP* pour permettre une réduction significative de ce dernier. Cette étude a mené à la notion de coloration majeure et de motif. Pour une coloration donnée, la permutation de deux classes couleurs n'impacte pas la validité de la solution. Cependant, la somme coloration associée à ces deux colorations est différente. *MSCP* ayant pour objectif de minimiser la somme coloration, seule la coloration parmi l'ensemble des colorations symétriques possédant ces classes couleurs triées par ordre décroissant de leur cardinalité est susceptible d'être solution du problème. Cette coloration est dite majeure. De plus, nous remarquons qu'une coloration majeure peut également être considérée comme une séquence d'entiers décroissante où chaque entier représente la cardinalité d'une classe couleur. Une telle représentation est appelée motif. Un même motif permet de représenter un ensemble de colorations majeures.

Après avoir adapté la relation de dominance existante entre un ensemble de motifs, nous avons introduit de nouvelles bornes supérieures  $UB_A$  et  $UB_S$  pour la force d'un graphe.  $UB_A$  est une borne algébrique considérant un motif et une coloration valide du graphe. Cependant,  $UB_A$  n'exploite aucune propriété structurelle du graphe hormis une coloration valide. Afin de renforcer l'idée générale derrière  $UB_A$ , nous avons introduit la borne algorithmique  $UB_S$  qui, en plus d'une coloration valide du graphe, considère la cardinalité du stable maximum du graphe (ou une borne supérieure).

Les résultats expérimentaux ont montré que notre première borne  $UB_A$  a permis d'améliorer les meilleures bornes connues de la littérature pour les graphes de densité faible. En ce qui concerne la borne  $UB_S$ , les résultats ont montré que notre approche améliore considérablement les meilleurs résultats connus.

Les motifs nous ont également permis d'apporter une contribution sur la borne inférieure de la somme chromatique.  $LBM\Sigma$  est une borne algébrique qui considère en plus d'une coloration valide du graphe la cardinalité du stable maximum d'un graphe (ou une borne supérieure). Les résultats ont montré que pour certaines familles de graphes  $LBM\Sigma$  est capable d'améliorer significativement les résultats. De plus, notre borne  $LBM\Sigma$  a prouvé pour la première fois dans la littérature l'optimalité de la meilleure somme chromatique connue pour quatre instances de la littérature.

La suite de nos travaux a naturellement porté sur la résolution exacte de *MSCP*. Pour cela, nous avons proposé quatre encodages permettant une modélisation de *MSCP* en un problème *weighted partial MaxSAT* et en un problème *weighted partial MinSAT*. Après une étude basée sur notre borne de la force d'un graphe,  $UB_S$ , permettant une réduction significative des instances générées par nos encodages, nous avons exploité l'un des meilleurs solveurs de résolution du problème *weighted partial MaxSAT* afin d'éprouver la puissance du formalisme *MaxSAT*. Les résultats ont

montré qu'une telle approche permettait de concurrencer les meilleurs techniques de résolution exacte connues pour *MSCP*. En ce qui concerne le problème *weighted partial MinSAT* peu de solveurs ont été développés pour résoudre le problème. De ce fait, les résultats sont inférieurs à ceux observés pour le problème *weighted partial MaxSAT*.

Les derniers travaux présentés dans ce manuscrit introduisent des méthodes de résolution de type *Branch-and-Bound*. Nos contributions sur ce domaine consistent en l'élaboration de deux solveurs *BBMSCP* et *3LMSCP* pour la résolution de *MSCP*. Pour *BBMSCP*, un noeud de l'arbre de recherche développé correspond à la sélection d'un sommet du graphe alors qu'une branche correspond à l'assignation d'une couleur à ce sommet. Afin d'évaluer la pertinence des différentes branches générées, nous avons mis en place les bornes  $LB_{B\Sigma}$  et  $LBP_{B\Sigma}$ . La borne  $LB_{B\Sigma}$  est une borne prenant en considération les plus petites couleurs des sommets disponibles en relâchant toute contrainte de voisinage. L'intérêt d'une telle approche est sa faible complexité. Pour  $LBP_{B\Sigma}$ , nous exploitons une décomposition en cliques des sommets non colorés afin de ne pas relâcher l'ensemble des contraintes de voisinages.

Le solveur *3LMSCP* est une approche différente de celle proposée par *BBMSCP*. Un noeud de l'arbre de recherche développé par *3LMSCP* correspond à une partition en stables des sommets traités du graphe tandis qu'une branche correspond à l'assignation d'un sommet non traité à une partie de la partition. Une telle approche parcourt l'espace des solutions de *MSCP* en considérant uniquement les colorations majeures. Par conséquent, cela implique une réduction significative de l'espace de recherche de *MSCP*. Afin d'évaluer la pertinence d'une branche, nous avons développé la borne inférieure de la somme chromatique  $LB_3\Sigma$ . Le principe de  $LB_3\Sigma$  est d'estimer la meilleure partition en stable du graphe en relâchant l'ensemble des contraintes de voisinage des sommets disponibles du graphe.

En ce qui concerne *BBMSCP* les résultats expérimentaux ont montré que notre approche *BBMSCP* n'était pas compétitive dans sa version actuelle. Les expérimentations ont montré que pour de petits graphes de densité allant de 0.5 à 0.9, *3LMSCP* permet d'obtenir de meilleurs résultats qu'une approche de type *MaxSAT*.

Les travaux présentés dans cette thèse ouvrent un vaste champ de perspectives. Tout d'abord, il serait intéressant de continuer à étudier la relation entre l'abstraction de l'espace des solutions par la considération des motifs et la structure du graphe. Cela permettrait de mettre en avant un ensemble de motifs non pertinents et d'affiner les résultats obtenus pour les bornes  $UB_A, UB_S$  et  $LBM\Sigma$ .

Nous remarquons également que le solveur *weighted partial MaxSAT* considéré dans ce manuscrit n'est pas le plus récent. Une étude des dernières avancées est envisagée afin d'utiliser les versions améliorées. De plus, nous envisageons également de développer un nouveau solveur pour la résolution du problème *weighted partial MinSAT* spécifique aux propriétés structurelles des instances *MSCP*. Pour cela une étude de ces instances et des dernières méthodes de résolution du problème est à effectuer.

Nos travaux concernant *BBMSCP* et *3LMSCP* nécessitent une étude approfondie des bornes existantes pour les méthodes de type *Branch-and-Bound* afin de pouvoir développer de nouvelles méthodes de coupe, plus efficaces, notamment pour notre approche *3LMSCP* dont les résultats sont extrêmement prometteurs.

# Index

| <b>Symboles</b>             |    |
|-----------------------------|----|
| $F(v)$ .....                | 20 |
| $O(t(n))$ .....             | 5  |
| $X^m$ .....                 | 42 |
| $X_i$ .....                 | 17 |
| $\Delta(G)$ .....           | 11 |
| $\Omega$ .....              | 24 |
| $\Omega(t(n))$ .....        | 5  |
| $\Pi$ .....                 | 4  |
| $\Psi(X)$ .....             | 42 |
| $\Sigma(G)$ .....           | 21 |
| $\alpha(G)$ .....           | 13 |
| $\beta$ .....               | 51 |
| $\lambda$ .....             | 51 |
| $\mathcal{I}$ .....         | 8  |
| $\mathcal{N}(v)$ .....      | 10 |
| $\mathcal{N}^+(v)$ .....    | 17 |
| $\mathcal{N}^-(v)$ .....    | 17 |
| $\mathcal{V}(S)$ .....      | 26 |
| $\omega(G)$ .....           | 13 |
| $\phi(n)$ .....             | 43 |
| $\phi(n, k)$ .....          | 43 |
| $\alpha$ .....              | 6  |
| $\geq$ .....                | 44 |
| $Disp(v)$ .....             | 17 |
| $TIME(t(n))$ .....          | 5  |
| $c(v)$ .....                | 16 |
| $d(v)$ .....                | 11 |
| $d_{N_{dmin}^-}^-(v)$ ..... | 26 |
| $d_{N_{dmin}^-}^-(v)$ ..... | 26 |
| $d_{N_{dsat}^-}^-(v)$ ..... | 25 |
| $d_{N_{dsat}^-}^-(v)$ ..... | 25 |
| $s(G)$ .....                | 21 |
| $x_{v,j}$ .....             | 65 |
| $NTIME(t(n))$ .....         | 5  |
| 3LMSCP .....                | 95 |
| <b>B</b>                    |    |
| BBMSCP .....                | 89 |
| <b>C</b>                    |    |
| classe couleur .....        | 17 |
| clause .....                | 9  |
| clique .....                | 12 |
| clique maximale .....       | 12 |
| clique maximum .....        | 12 |
| CliquePart .....            | 91 |
| CNF .....                   | 9  |
| coloration .....            | 16 |
| majeure .....               | 42 |
| partielle .....             | 17 |
| valide .....                | 16 |
| COP .....                   | 34 |
| CSP .....                   | 33 |
| <b>F</b>                    |    |
| $F_{dMSCP}$ .....           | 65 |
| $F_{sMSCP}$ .....           | 65 |
| force .....                 | 21 |
| <b>G</b>                    |    |
| $G_c$ .....                 | 89 |
| $G_{\bar{c}}$ .....         | 89 |
| GCP .....                   | 18 |
| graphe .....                | 10 |
| arbre .....                 | 13 |
| biparti .....               | 13 |
| complémentaire .....        | 12 |
| complet .....               | 11 |
| intervalles .....           | 14 |
| k-arbre .....               | 13 |
| k-arbre partiel .....       | 14 |
| $P_k$ .....                 | 15 |
| $P_k$ -sparse .....         | 15 |
| parfait .....               | 18 |
| partiel .....               | 11 |
| planaire .....              | 15 |
| sous-graphe .....           | 11 |
| split-graphe .....          | 15 |
| <b>I</b>                    |    |
| interprétation .....        | 8  |
| <b>K</b>                    |    |
| k-coloration .....          | 17 |
| <b>L</b>                    |    |
| $LB_3\Sigma$ .....          | 96 |
| $LB_B\Sigma$ .....          | 90 |

|                                 |    |                       |    |
|---------------------------------|----|-----------------------|----|
| LBM $\Sigma$ .....              | 58 | UB <sub>S</sub> ..... | 53 |
| LBP <sub>B</sub> $\Sigma$ ..... | 91 |                       |    |
| littéral.....                   | 9  |                       |    |
| <b>M</b>                        |    |                       |    |
| machine de Turing.....          | 4  |                       |    |
| déterministe.....               | 4  |                       |    |
| non-déterministe.....           | 4  |                       |    |
| MaxSAT.....                     | 64 |                       |    |
| MDSAT.....                      | 26 |                       |    |
| MinSAT.....                     | 68 |                       |    |
| modèle.....                     | 9  |                       |    |
| motif.....                      | 43 |                       |    |
| majeur.....                     | 51 |                       |    |
| MRLF.....                       | 26 |                       |    |
| MSCP.....                       | 21 |                       |    |
| <b>N</b>                        |    |                       |    |
| NP.....                         | 6  |                       |    |
| NP-complet.....                 | 6  |                       |    |
| NP-difficile.....               | 6  |                       |    |
| npSMC.....                      | 20 |                       |    |
| <b>O</b>                        |    |                       |    |
| OCCP.....                       | 19 |                       |    |
| optimum                         |    |                       |    |
| global.....                     | 28 |                       |    |
| local.....                      | 28 |                       |    |
| ordre total.....                | 46 |                       |    |
| <b>P</b>                        |    |                       |    |
| P.....                          | 5  |                       |    |
| partial                         |    |                       |    |
| MaxSAT.....                     | 64 |                       |    |
| MinSAT.....                     | 68 |                       |    |
| PLNE.....                       | 31 |                       |    |
| problème                        |    |                       |    |
| d'optimisation.....             | 5  |                       |    |
| décidable.....                  | 4  |                       |    |
| indécidable.....                | 4  |                       |    |
| pSMC.....                       | 20 |                       |    |
| <b>R</b>                        |    |                       |    |
| réduction polynomiale.....      | 6  |                       |    |
| réseau de contraintes.....      | 33 |                       |    |
| pondérées.....                  | 35 |                       |    |
| <b>S</b>                        |    |                       |    |
| SAT.....                        | 10 |                       |    |
| stable.....                     | 12 |                       |    |
| stable maximal.....             | 13 |                       |    |
| stable maximum.....             | 13 |                       |    |
| <b>U</b>                        |    |                       |    |
| UB <sub>A</sub> .....           | 50 |                       |    |

# Bibliographie

- Carlos ANSÓTEGUI, Joel GABAS, Yuri MALITSKY et Meinolf SELLMANN : Maxsat by improved instance-specific algorithm configuration. *Artificial Intelligence*, 235:26–39, 2016. [2](#), [81](#), [85](#), [92](#)
- Kenneth I APPEL et Wolfgang HAKEN : *Every Planar Map is Four Colorable*, volume 98. American Mathematical Soc., 1989. [18](#)
- Amotz BAR-NOY, Mihir BELLARE, Magnús M HALLDÓRSSON, Hadas SHACHNAI et Tami TAMIR : On chromatic sums and distributed resource allocation. *Information and Computation*, 140(2):183–202, 1998. [16](#)
- Amotz BAR-NOY, Magnús M HALLDÓRSSON, Guy KORTSARZ, Ravit SALMAN et Hadas SHACHNAI : Sum multicoloring of graphs. *Journal of Algorithms*, 37(2):422–450, 2000. [20](#)
- Amotz BAR-NOY et Guy KORTSARZ : Minimum color sum of bipartite graphs. *Journal of Algorithms*, 28(2):339–365, 1998. [47](#)
- Nicolas BARNIER et Pascal BRISSET : Graph coloring for air traffic flow management. *Annals of operations research*, 130(1):163–178, 2004. [16](#)
- Pierre BÉJIAN : Matroïdes et algorithmes gloutons : une introduction. [24](#)
- Una BENLIC et Jin-Kao HAO : A study of breakout local search for the minimum sum coloring problem. In *SEAL*, pages 128–137. Springer, 2012. [1](#), [23](#), [29](#)
- Flavia BONOMO et Mario VALENCIA-PABON : Minimum sum coloring of p4-sparse graphs. *Electronic Notes in Discrete Mathematics*, 35:293–298, 2009. [2](#), [44](#), [45](#), [62](#)
- Flavia BONOMO et Mario VALENCIA-PABON : On the minimum sum coloring of p 4-sparse graphs. *Graphs and Combinatorics*, 30(2):303–314, 2014. [2](#), [44](#), [45](#), [47](#), [62](#)
- Hend BOUZIRI et Mouna JOUINI : A tabu search approach for the sum coloring problem. *Electronic Notes in Discrete Mathematics*, 36:915–922, 2010. [23](#)
- Daniel BRÉLAZ : New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979. [25](#)
- Gregory J CHAITIN : Register allocation & spilling via graph coloring. 17(6):98–105, 1982. [16](#)
- Fred C CHOW et John L HENNESSY : The priority-based coloring approach to register allocation. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 12(4):501–536, 1990. [16](#)
- COLOR : <http://mat.gsia.cmu.edu/COLOR/instances.html>. [ix](#), [37](#), [54](#), [59](#), [62](#), [71](#), [81](#), [82](#), [92](#), [94](#), [95](#)
- Stephen A COOK : The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971. [6](#), [7](#)
- Joseph CULBERSON : Iterated greedy graph coloring and the difficulty landscape. 1992. [40](#)
- Charles DICKENS : *David copperfield*. Bradbury & Evans, 1850. [37](#)

- DIMACS : <ftp://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/color/>. [ix](#), [37](#), [54](#), [59](#), [62](#), [71](#), [81](#), [82](#), [92](#), [94](#), [95](#)
- Philippe GALINIER et Jin-Kao HAO : Hybrid evolutionary algorithms for graph coloring. *Journal of combinatorial optimization*, 3(4):379–397, 1999. [31](#)
- Michael R GAREY et David S JOHNSON : *Computers and intractability*. Freeman, 1979. [7](#)
- Krzysztof GIARO, Robert JANCZEWSKI, Marek KUBALE et Michal MALAFIEJSKI : A 27/26-approximation algorithm for the chromatic sum coloring of bipartite graphs. *Lecture notes in computer science*, pages 135–145, 2002. [22](#)
- Fred GLOVER : Tabu search—part i. *ORSA Journal on computing*, 1(3):190–206, 1989. [28](#)
- Stefano GUALANDI et Federico MALUCELLI : Exact solution of graph coloring problems via constraint programming and column generation. *INFORMS Journal on Computing*, 24(1):81–100, 2012. [37](#), [54](#), [59](#), [94](#)
- Nebojša GVOZDENOVIĆ et Monique LAURENT : Computing semidefinite programming lower bounds for the (fractional) chromatic number via block-diagonalization. *SIAM Journal on Optimization*, 19(2):592–615, 2008. [59](#)
- Hossein HAJIABOLHASSAN, Mojtaba L MEHRABADI et Ruzbeh TUSSEKANI : Minimal coloring and strength of graphs. *Discrete Mathematics*, 215(1-3):265–270, 2000. [ix](#), [47](#), [54](#), [55](#), [62](#)
- William K HALE : Frequency assignment : Theory and applications. *Proceedings of the IEEE*, 68(12):1497–1514, 1980. [16](#)
- Magnús HALLDÓRSSON, Guy KORTSARZ et Hadas SHACHNAI : Minimizing average completion of dedicated tasks and interval graphs. *Approximation, Randomization, and Combinatorial Optimization : Algorithms and Techniques*, pages 114–126, 2001. [1](#), [48](#)
- Magnús M HALLDÓRSSON et Guy KORTSARZ : Multicoloring : Problems and techniques. In *MFCS*, pages 25–41. Springer, 2004. [48](#)
- Godfrey H HARDY et Srinivasa RAMANUJAN : Asymptotic formulæ in combinatory analysis. *Proceedings of the London Mathematical Society*, 2(1):75–115, 1918. [44](#)
- Anders HELMAR et Marco CHIARANDINI : A local search heuristic for chromatic sum. In *Proceedings of the 9th metaheuristics international conference*, volume 1101, pages 161–170, 2011. [1](#), [23](#)
- David HILBERT et Wilhelm ACKERMANN : *Principles of mathematical logic*, volume 69. American Mathematical Soc., 1950. [3](#)
- HOMÈRE : *Iliade*. [37](#)
- Victor HUGO : *Les misérables*. Albert Lacroix et Cie, 1862. [37](#)
- Klaus JANSEN : The optimum cost chromatic partition problem. *Algorithms and complexity*, pages 25–36, 1997. [19](#), [48](#)
- Klaus JANSEN : Approximation results for the optimum cost chromatic partition problem. *Journal of Algorithms*, 34(1):54–89, 2000. [47](#)
- Yan JIN et Jin-Kao HAO : Hybrid evolutionary search for the minimum sum coloring problem of graphs. *Information Sciences*, 352:15–34, 2016. [23](#), [37](#), [54](#), [59](#)
- Yan JIN, Jin-Kao HAO et Jean-Philippe HAMIEZ : A memetic algorithm for the minimum sum coloring problem. *Computers & Operations Research*, 43:318–327, 2014. [1](#), [23](#), [37](#), [54](#), [59](#)

- David S JOHNSON, Cecilia R ARAGON, Lyle A MCGEOCH et Catherine SCHEVON : Optimization by simulated annealing : an experimental evaluation ; part ii, graph coloring and number partitioning. *Operations research*, 39(3):378–406, 1991. [38](#)
- Richard M KARP : Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972. [6](#), [13](#), [17](#), [18](#)
- Scott KIRKPATRICK, C Daniel GELATT, Mario P VECCHI *et al.* : Optimization by simulated annealing. *science*, 220(4598):671–680, 1983. [28](#)
- Donald Ervin KNUTH : *The Stanford GraphBase : a platform for combinatorial computing*, volume 37. Addison-Wesley Reading, 1993. [37](#)
- Rajeev KOHLI, Ramesh KRISHNAMURTI et Prakash MIRCHANDANI : The minimum satisfiability problem. *SIAM Journal on Discrete Mathematics*, 7(2):275–283, 1994. [68](#)
- Zbigniew KOKOSIŃSKI et Krzysztof KWARCIAŃY : On sum coloring of graphs with parallel genetic algorithms. *Adaptive and Natural Computing Algorithms*, pages 211–219, 2007. [23](#), [58](#), [59](#), [62](#)
- Leo G KROON, Arunabha SEN, Haiyong DENG et Asim ROY : The optimal cost chromatic partition problem for trees and interval graphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 279–292. Springer, 1996. [16](#), [19](#), [22](#)
- Ewa KUBICKA et Allen J SCHWENK : An introduction to chromatic sums. In *Proceedings of the 17th conference on ACM Annual Computer Science Conference*, pages 39–45. ACM, 1989. [1](#), [21](#), [22](#), [48](#)
- Benoit LAURENT et Jin-Kao HAO : List-graph colouring for multiple depot vehicle scheduling. *International Journal of Mathematics in Operational Research*, 1(1-2):228–245, 2009. [16](#)
- Clément LECAT, C Min LI, Corinne LUCET et Yu LI : Comparaison de méthodes de résolution pour le probleme de somme coloration. In *JFPC-2015*, pages 204–213, 2015a. [2](#), [33](#), [35](#), [63](#), [87](#)
- Clément LECAT, Chu-Min LI, Corinne LUCET et Yu LI : Exact methods for the minimum sum coloring problem. In *Doctoral Program of Constraint Programming (CP-DP'15)*, pages 61–69, 2015b. [2](#), [33](#), [35](#), [63](#), [87](#)
- Clément LECAT, Corinne LUCET et Chu-Min LI : Bornes supérieures de la force et borne inférieure de la somme chromatique d'un graphe. *Bulletin de la Société Française de Recherche Opérationnelle et d'Aide à la Décision*, (38):10–13, Juillet 2017a. [2](#), [42](#)
- Clément LECAT, Corinne LUCET et Chu-Min LI : Considération des motifs dans la détermination d'une borne supérieure de la force d'un graphe. In *JFPC-2017*, 2017b. [2](#), [42](#)
- Clément LECAT, Corinne LUCET et Chu-Min LI : Minimum sum coloring problem : Upper bounds for the chromatic strength. *Discrete Applied Mathematics*, 233:71–82, 2017c. [2](#), [42](#), [45](#)
- Clément LECAT, Corinne LUCET et Chu-Min LI : New lower bound for the minimum sum coloring problem. In *AAAI*, pages 853–859, 2017d. [42](#), [45](#)
- Clément LECAT, Corinne LUCET et Chu-Min LI : Problème de somme coloration minimale : Borne inférieure de la somme chromatique et borne supérieure de la force du graphe. In *ROADEF 2017*, 2017e. [2](#), [42](#)
- Frank Thomson LEIGHTON : A graph coloring algorithm for large scheduling problems. *Journal of research of the national bureau of standards*, 84(6):489–506, 1979. [16](#), [25](#), [38](#)
- Gary LEWANDOWSKI et Anne CONDON : Experiments with parallel graph coloring heuristics and applications of graph coloring. *DIMACS Series in Discrete Mathematics*, 1994. [39](#)

- Chu-Min LI, Zhiwen FANG et Ke XU : Combining maxsat reasoning and incremental upper bound for the maximum clique problem. *In Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, pages 939–946. IEEE, 2013. [53](#), [54](#), [62](#)
- Chu-Min LI, Zhu ZHU, Felip MANYÀ et Laurent SIMON : Minimum satisfiability and its applications. *In IJCAI*, pages 605–610, 2011. [81](#), [85](#)
- Yu LI, Corinne LUCET, Aziz MOUKRIM et Kaoutar SGHIOUER : Greedy algorithms for the minimum sum coloring problem. *In Logistique et transports*, pages LT-027, 2009. [23](#), [25](#), [26](#), [89](#)
- Enrico MALAGUTI, Michele MONACI et Paolo TOTH : An exact approach for the vertex coloring problem. *Discrete Optimization*, 8(2):174–190, 2011. [37](#), [54](#)
- Anuj MEHROTRA et Michael A TRICK : A column generation approach for graph coloring. *informs Journal on Computing*, 8(4):344–354, 1996. [94](#)
- Maël MINOT, Samba Ndojh NDIAYE et Christine SOLNON : Évaluation d’approches complètes pour le problème de somme coloration. *In Douzièmes Journées Francophones de Programmation par Contraintes (JFPC 2016)*, 2016a. [1](#), [35](#), [83](#)
- Maël MINOT, Samba Ndojh NDIAYE et Christine SOLNON : Using cp and ilp with tree decomposition to solve the sum colouring problem. *In Doctoral program of CP 2016*, 2016b. [32](#), [35](#), [83](#)
- Maël MINOT, Samba Ndojh NDIAYE et Christine SOLNON : Combining cp and ilp in a tree decomposition of bounded height for the sum colouring problem. *In International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 359–375. Springer, 2017. [2](#), [32](#), [35](#), [63](#), [70](#), [83](#), [85](#)
- John MITCHEM et Patrick MORRIS : On the cost-chromatic number of graphs. *Discrete Mathematics*, 171(1-3):201–211, 1997. [47](#), [48](#)
- Ugo MONTANARI : Networks of constraints : Fundamental properties and applications to picture processing. *Information sciences*, 7:95–132, 1974. [33](#)
- Aziz MOUKRIM, Kaoutar SGHIOUER, Corinne LUCET et Yu LI : Upper and lower bounds for the minimum sum coloring problem, submitted for publication, 2014. [1](#), [23](#), [29](#), [59](#), [62](#), [90](#)
- Jan MYCIELSKI : Sur le coloriage des graphes. *In Colloq. Math*, volume 3, page 9, 1955. [37](#)
- Sara NICOLOSO : Sum coloring and interval graphs : a tight upper bound for the minimum number of colors. *Discrete Mathematics*, 280(1):251–257, 2004. [48](#)
- Christos H PAPANITRIOU : *Computational complexity*. John Wiley and Sons Ltd., 2003. [7](#), [64](#)
- Neil ROBERTSON, Daniel SANDERS, Paul SEYMOUR et Robin THOMAS : A new proof of the four-colour theorem. *Electronic research announcements of the American Mathematical Society*, 2(1):17–25, 1996. [18](#)
- Mohammad R SALAVATIPOUR : On sum coloring of graphs. *Discrete Applied Mathematics*, 127(3):477–488, 2003. [22](#), [47](#)
- Arunabha SEN, Haiyong DENG et Sumanta GUHA : On a graph partition problem with application to vlsi layout. *Information processing letters*, 43(2):87–94, 1992. [1](#), [31](#)
- Kaoutar SGHIOUER : *Probleme de la somme coloration : approches heuristiques et bornes inférieures*. Thèse de doctorat, Compiègne, 2011. [23](#)
- Kenneth J SUPOWIT : Finding a maximum planar subset of a set of nets in a channel. *IEEE transactions on computer-aided design of integrated circuits and systems*, 6(1):93–94, 1987. [16](#), [18](#)

- Leo TOLSTOY : *Anna karenina*. Rousky vestnik, 1877. [37](#)
- Alan Mathison TURING : On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London mathematical society*, 2(1):230–265, 1937. [3](#)
- Mark TWAIN : *Les aventures de Huckleberry Finn*. Chatto & Windus, 1884. [37](#)
- Yang WANG, Jin-Kao HAO, Fred GLOVER et Zhipeng LÜ : Solving the minimum sum coloring problem via binary quadratic programming. *arXiv preprint arXiv :1304.5876*, 2013. [32](#), [83](#)
- Tai-Kuo WOO, Stanley Y.W. SU et Richard NEWMAN-WOLFE : Resource allocation in a dynamically partitionable bus network using a graph coloring algorithm. *IEEE Transactions on Communications*, 39(12):1794–1801, 1991. [16](#)
- Qinghua WU et Jin-Kao HAO : Coloring large graphs based on independent set extraction. *Computers & Operations Research*, 39(2):283–290, 2012a. [26](#)
- Qinghua WU et Jin-Kao HAO : An effective heuristic algorithm for sum coloring of graphs. *Computers & Operations Research*, 39(7):1593–1600, 2012b. [23](#), [26](#)
- Qinghua WU et Jin-Kao HAO : An adaptive multistart tabu search approach to solve the maximum clique problem. *Journal of Combinatorial Optimization*, 26(1):86–108, 2013a. [26](#)
- Qinghua WU et Jin-Kao HAO : Improved lower bounds for sum coloring via clique decomposition. *arXiv preprint arXiv :1303.6761*, 2013b. [59](#), [62](#), [90](#)





# Résumé

Le Problème de la Somme Coloration Minimum (MSCP) d'un graphe est un problème d'optimisation combinatoire dont l'objectif est de déterminer une coloration valide minimisant la somme des poids associés aux couleurs utilisées. Le nombre minimum de couleurs dans une solution optimale de MSCP est appelé la force du graphe, et la somme des poids des couleurs utilisées est appelée la somme chromatique du graphe. L'objectif de cette thèse a été d'étudier MSCP afin de proposer de nouvelles approches permettant sa résolution. La contribution de cette thèse est double. Premièrement, nous avons introduit deux nouvelles bornes supérieures de la force d'un graphe et une nouvelle borne inférieure de la somme chromatique, basées sur la notion de motif que nous avons adapté de la littérature. L'intérêt majeur de ces travaux est la réduction de l'espace des solutions de MSCP. Deuxièmement, nous avons proposé plusieurs approches de résolution exacte de MSCP. La première méthode consiste en une modélisation de MSCP en un formalisme MaxSAT partiel pondéré ou MinSAT partiel pondéré afin d'utiliser les solveurs MaxSAT/MinSAT de l'état de l'art pour résoudre MSCP. Nous avons montré que notre borne de la force du graphe permet de réduire grandement la taille des instances MaxSAT/MinSAT obtenues et de rendre les solveurs MaxSAT compétitifs pour résoudre MSCP. Les deux autres méthodes de résolution proposées sont des approches de type Branch-and-Bound appelées BBMSCP et 3LMSCP. La différence entre BBMSCP et 3LMSCP est que 3LMSCP exploite la partition du graphe en stables afin de ne pas considérer les colorations symétriques.

Cette thèse bénéficie d'un financement du ministère français de l'Enseignement supérieur, de la Recherche et de l'Innovation.

# Abstract

The Minimum Sum Coloring Problem (MSCP) of a graph is an optimization problem whose the aim is to find a valid coloring such that the sum of weights associated to the used color is minimum. The minimum number of colors needed in an optimal solution is called the strength of the graph, and the sum of weights of the colors used in an optimal solution is called the chromatic sum of the graph. The aim of this thesis was to study the MSCP in order to propose new approaches for its resolution. The contribution of the thesis is twofold. First, we have introduced two upper bounds of the strength and one lower bound of the chromatic sum of a graph, based on a notion called motif adapted from the literature. These bounds allow to reduce significantly the search space of the MSCP. Second, we have proposed several exact resolution methods for the MSCP. The first method consists in modelling the MSCP to a weighted partial MaxSAT or a weighted partial MinSAT and solves the MSCP using state-of-the-art MaxSAT/MinSAT solvers. We have showed that our upper bound of the strength allows to improve substantially the MaxSAT/MinSAT encodings, and consequently, the MaxSAT solvers are competitive to solve the MSCP. The two other resolution methods are based on the Branch-and-Bound scheme and are called BBMSCP and 3LMSCP respectively. The difference between BBMSCP and 3LMSCP is that BBMSCP explores many symmetric colorings but 3LMSCP exploits the partitions of the graph into stables to avoid symmetric colorings.

This thesis was financed by the Ministry of Higher Education, Research and Innovation of France.