



HAL
open science

Somme coloration de graphe

Kaoutar Sghiouer, Yu Li, Corinne Lucet, Aziz Moukrim

► **To cite this version:**

Kaoutar Sghiouer, Yu Li, Corinne Lucet, Aziz Moukrim. Somme coloration de graphe. 8e Conf2rence Internationale de MOd2lisation et SIMulation - MOSIM'10., May 2010, Hammamet, Tunisie. pp.276. hal-00527271

HAL Id: hal-00527271

<https://hal.science/hal-00527271v1>

Submitted on 18 Oct 2010

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.

SOMME COLORATION DE GRAPHE

K. SGHIOUER*, Y. LI**

C. LUCET**, A. MOUKRIM*

* Université de Technologie de Compiègne
Heudiasyc UMR CNRS 6599 BP 20529
Centre de Recherches de Royallieu
60205 Compiègne - France

**Université de Picardie Jules Verne
Laboratoire MIS EA 4290
33 rue St Leu
80000 Amiens - France

kaoutar.sghiouer@utc.fr, aziz.moukrim@utc.fr yu.li@u-picardie.fr, corinne.lucet@u-picardie.fr

RÉSUMÉ : Dans cet article nous étudions le problème de la somme coloration minimale (MSCP) qui consiste à trouver une coloration des sommets d'un graphe, utilisant des entiers naturels, telle que la somme des couleurs est minimale. Nous présentons une synthèse des travaux sur MSCP, ainsi qu'une nouvelle famille d'algorithmes gloutons pour MSCP. Nous proposons une étude de bornes inférieures basée sur l'extraction de graphes partiels. Nous comparons les résultats obtenus avec ceux de la littérature. Les résultats numériques montrent que nos approches améliorent les bornes inférieures et supérieures de la littérature sur les instances utilisées dans (Kokosiński & Kwarcianny (2007)).

MOTS-CLÉS : Optimisation combinatoire, problème de coloration de graphe, bornes inférieures.

1 Introduction

Le problème de la coloration de graphe (GCP) est un problème NP-complet dans le cas général (Garey & Johnson (1990)). Il existe de nombreux problèmes liés à la coloration des sommets d'un graphe qui constituent des outils intéressants dans la modélisation de problèmes d'ordonnancement (Marx (2004)) tels que le problème de la somme chromatique de coût optimal (OCCP), le problème de la somme multicoloration d'un graphe (SMP) ainsi que le problème de la somme coloration minimale (MSCP). Nous nous intéressons dans cet article au MSCP. Nous avons tout d'abord procédé à une synthèse des travaux sur MSCP, ainsi qu'à une étude des algorithmes gloutons utilisés pour GCP. Ces derniers jouent un rôle important dans la pratique : d'une part, ils peuvent être utilisés pour obtenir une solution dans le cas où l'exigence du temps de calcul est très importante et d'autre part, ils peuvent être intégrés dans des méthodes exactes ou approchées. Nous présentons ici une nouvelle famille d'algorithmes gloutons dédiés au MSCP, ainsi que de nouvelles bornes inférieures. Dans la section 2, nous rappelons la définition des problèmes de coloration de graphe. Dans la section 3, nous présentons quelques résultats théoriques sur certaines familles de graphes pour MSCP. Ensuite dans la section 4, nous étudions une nouvelle variante de l'heuristique gloutonne RLF (Leighton (1979)) plus spécifiquement dédiée à la résolution de MSCP. Enfin, nous proposons dans la section 5 une méthode permettant le calcul de bornes inférieures pour des graphes quelconques, basée sur

l'extraction de graphes partiels.

2 Définitions et préliminaires

Nous rappelons ici quelques notions concernant les graphes avant de procéder aux définitions des problèmes.

2.1 Préliminaires

Un graphe *non-orienté* est un couple, $G = (V, E)$, composé d'un ensemble de sommets (ou nœuds) V et d'un ensemble d'arêtes E . Dans la suite $|V| = n$ et $|E| = m$. G est *connexe*, si pour toute paire de sommets $\{u, v\}$ dans V avec $u \neq v$, il existe une chaîne reliant u à v . Sans perte de généralité, les graphes considérés dans la suite seront tous non-orientés et connexes. G est *complet* si et seulement si pour toute paire de sommets $\{u, v\}$ dans V avec $u \neq v$, u et v sont adjacents.

Le *graphe complémentaire* de G est $\bar{G} = (V, \bar{E})$ tel que pour toute paire de sommets $\{u, v\}$ dans V avec $u \neq v$, $[u, v] \in \bar{E}$ si et seulement si $[u, v] \notin E$. Un *graphe partiel* $G' = (V, E')$ de G est un graphe tel que $E' \subset E$.

On appelle *voisinage* d'un sommet v , noté $N(v)$, l'ensemble des sommets qui sont adjacents à v . Les sommets appartenant au voisinage de v dans G sont appelés *voisins* de v .

Le *degré* d'un sommet v du graphe G est le nombre de voisins de ce sommet. Soit $\Delta(G)$ le degré de G , il a pour valeur le degré maximal parmi tous les sommets

de G .

Un ensemble *stable* ou *indépendant* de G est un ensemble de sommets deux à deux non adjacents.

Une *clique* C est un sous-graphe complet (i.e tous les sommets de C sont reliés deux à deux). Si cette clique ne peut être étendue en ajoutant un autre sommet de $V \setminus C$, alors C est une *clique maximale*. Une clique est dite *clique maximum* de G si sa cardinalité, notée $\omega(G)$, est maximale dans G .

Un graphe G est un *arbre* s'il est connexe sans cycle. G est *biparti* si l'ensemble de ses sommets peut être divisé en deux ensembles disjoints V_1 et V_2 tels que chaque arête ait une extrémité dans V_1 et l'autre dans V_2 . G est appelé *graphe d'intervalles* s'il existe une bijection entre les sommets du graphe et un ensemble de segments d'une droite telle que deux sommets u et v sont reliés par une arête si et seulement si les segments correspondant à u et v se chevauchent. G est *planaire* si on peut le dessiner dans le plan de sorte que ses arêtes ne se croisent pas. G est un *split-graphe* si et seulement si l'ensemble de ses sommets peut être divisé en une clique C et un ensemble indépendant I . Un *couplage* est un ensemble d'arêtes deux à deux disjointes. Un couplage est un *couplage maximum* si parmi tous les couplages du graphe, il possède le plus grand nombre possible d'arêtes.

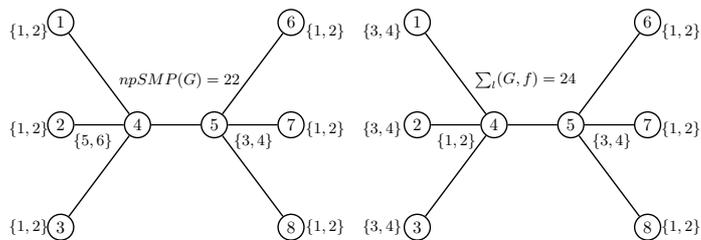
2.2 Problème de coloration de graphe

Une *coloration valide* d'un graphe $G = (V, E)$ est une fonction $c : v \mapsto c(v)$ associant à tout sommet $v \in V$ une couleur $c(v)$, avec $c(v)$ un entier naturel entre 1 et $|V|$, en s'assurant que $c(v) \neq c(u)$ pour toute arête $[u, v] \in E$. Le *nombre chromatique* $\chi(G)$ d'un graphe G est le nombre minimum de couleurs à utiliser pour colorier G . La résolution du problème de coloration de graphe (GCP) consiste à trouver une coloration valide utilisant $\chi(G)$ couleurs. Une coloration valide, peut être vue comme une partition des sommets en k sous-ensembles indépendants, appelés *classes de couleur* : X_1, \dots, X_k , où tout sommet de X_i est de couleur i . On note x_i la cardinalité de X_i .

2.3 Définition du problème OCCP

Soit $G = (V, E)$ un graphe et w_1, \dots, w_k les coûts associés aux classes de couleur $1, \dots, k$. Le problème de la partition chromatique de coût optimal (OCCP) consiste à trouver une coloration valide telle que $\sum w_i |X_i|$ est minimale. Cette somme est appelée *coût optimal chromatique*. Le plus petit nombre de couleurs utilisées pour obtenir le coût optimal chromatique est appelé *force de OCCP*.

OCCP a été introduit dans le cadre des VLSI (Supowit (1987)). Il existe d'autres applications de OCCP présentées dans (Kroon *et al.* (1997)). OCCP est NP-Difficile dans le cas général, et le reste pour



(a) Solution optimale pour SMP (b) Solution valide pour SMP

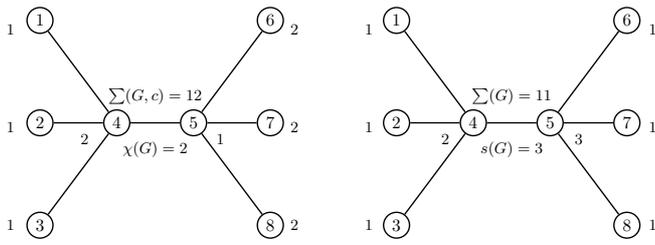
Figure 1: Comparaison entre une solution optimale et une solution valide pour SMP

des graphes particuliers tels que les graphes bipartis (Jansen (1997)) et les graphes d'intervalles (Kroon *et al.* (1997)). Néanmoins, il se résout en temps polynomial pour les arbres (Kroon *et al.* (1997)). Un schéma d'approximation pour les graphes bipartis et les graphes d'intervalles de ratio $O(n^{0.5})$ est présenté dans (Jansen (2000)). Ce même auteur démontre dans (Jansen (1997)) qu'il n'existe pas de schéma d'approximation polynomial de ratio $O(n^{1-\epsilon})$ pour les split-graphes.

2.4 Définition du problème SMP

Soit $G = (V, E)$ un graphe. A chaque sommet v de V , on associe un nombre de couleurs requis appelé *longueur* de v et noté $l(v)$. Une multicoloration des sommets de G est une fonction $f : V \rightarrow 2^{\mathbb{N}}$, telle que $|f(v)| = l(v)$ et pour toute paire de sommets $\{u, v\}$ dans V avec $u \neq v$ $f(u) \cap f(v) = \emptyset$ si u et v sont adjacents. Le but de SMP est de minimiser $\sum_l(G, f) = \sum_{v \in V} C_f^1(v)$, où $C_f^1(v)$ est la plus grande couleur affectée à v par f . Notons par $C_f^0(v)$ la plus petite couleur affectée à v par f . On appelle multicoloration *contigüe* ou *non preemptive*, une multicoloration telle que pour tout sommet v de V on a $C_f^1(v) = C_f^0(v) + (l(v) - 1)$. La plus petite somme multicoloration est appelée *somme multicoloration chromatique* et est notée $pSMP(G)$ pour le cas préemptif et $npSMP(G)$ pour le cas non préemptif. L'exemple de la figure 1 illustre ce problème sur un arbre où tous les sommets ont une longueur de 2. L'application courante de SMP est le problème d'ordonnancement où les temps d'exécution sont quelconques.

Il existe des schémas d'approximation pour des graphes arbitraires (Bar-Noy *et al.* (2000)) ainsi que des schémas d'approximation polynomiaux pour des graphes particuliers tels que les arbres (Halldórsson *et al.* (2003)) et les graphes planaires (Halldórsson & Kortsarz (1999)). Pour les chaînes, il existe un algorithme dans le cas préemptif (Kovács (2004)) en $O(n^3 p)$ où p est la plus grande longueur des sommets du graphe.



(a) Solution optimale pour GCP (b) Solution optimale pour MSCP

Figure 2: Comparaison de GCP et MSCP sur les arbres.

2.5 Problème de la somme coloration d'un graphe

Le problème de la somme coloration d'un graphe (MSCP) correspond au OCCP lorsque la valeur des coûts w_i de la couleur i vaut i lui même. Il correspond également au SMP lorsque la longueur des sommets vaut 1. Plus précisément nous définissons MSCP comme suit.

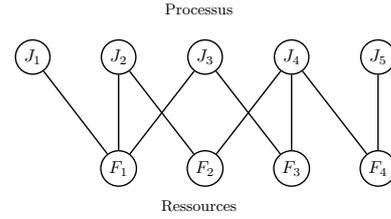
Soit $G = (V, E)$ un graphe, le problème de la somme coloration minimale de G (MSCP) consiste à trouver une coloration valide c telle que $\sum(G, c) = \sum_{v \in V} c(v)$ est minimale. Cette somme coloration minimale est appelée *somme chromatique* et est notée $\sum(G)$. Le plus petit nombre de couleurs utilisées pour colorier le graphe G dans une solution optimale pour MSCP est appelé *force* du graphe, et noté $s(G)$. Si on a une coloration valide de G , avec X_1, \dots, X_k ses classes de couleur, la somme coloration associée est la suivante :

$$\sum(G) = 1.x_1 + 2.x_2 + \dots + k.x_k$$

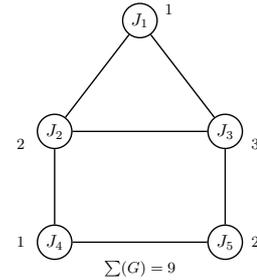
avec $x_1 \geq \dots x_i \dots \geq x_k$.

Trouver une solution optimale pour MSCP, ne revient pas à trouver parmi toutes les solutions optimales pour GCP celle pour laquelle la réorganisation des classes de couleur par cardinalité décroissante donne la plus petite somme. Il est noté dans (Kubicka & Schwenk (1989)) que même pour les arbres dont le nombre chromatique est 2, il est parfois nécessaire d'utiliser plus de 2 couleurs pour trouver la solution optimale pour MSCP. L'arbre de la figure 2 illustre bien ce fait.

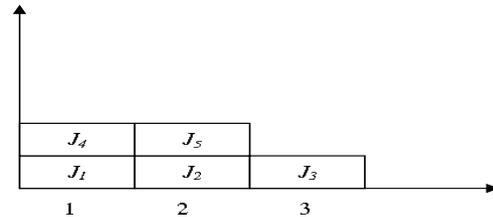
Il existe plusieurs applications pour MSCP, citons par exemple le problème d'allocation de ressources quand le but est de minimiser le temps de réponse moyen des machines, ce qui revient à minimiser la somme des temps d'achèvement des processus (Bar-Noy *et al.* (1998)). On notera J_1, J_2, \dots, J_n les processus à effectuer et F_1, F_2, \dots, F_m les ressources dont ces processus ont besoin pour se réaliser. Chaque processus requiert un accès exclusif à une ressource donnée (J_i et J_k ne peuvent accéder en



(a) Graphe des besoins des processus en ressources



(b) Graphe des conflits



(c) Diagramme de Gantt, Temps moyen = $\frac{9}{5}$

Figure 3: Exemple d'application pour MSCP

même temps à la ressource F_s). L'ensemble de ces contraintes est modélisé par un graphe dont les sommets représentent les processus, et il existe une arête entre deux sommets J_i et J_k , si ces deux processus ont besoin d'accéder à une même ressource F_s . Une somme coloration minimale du graphe des conflits correspondant, fournira un ordonnancement des processus avec un temps de réponse moyen optimal. La figure 3 illustre un exemple de 5 processus et 4 ressources. Le schéma 3(a) montre quels sont les besoins de chaque processus en ressources. On en déduit le graphe des conflits 3(b) et la somme coloration optimale $\sum(G) = 9$. Ainsi, on construit le diagramme de Gantt 3(c) et le temps moyen minimal de réponse des processus vaut 1,8.

3 MSCP pour quelques familles de graphes

Le problème de la somme coloration d'un graphe (MSCP) a été introduit indépendamment par Supowit (Supowit (1987)) et Kubicka (Kubicka (1989)). Kubicka et Schwenk ont démontré dans (Kubicka & Schwenk (1989)) que trouver la somme chromatique est NP-Difficile. Néanmoins, pour certaines familles de graphes la somme chromatique peut être calculée de façon exacte en temps polynomial. Ainsi plusieurs travaux pour MSCP ont été publiés sur des familles de graphes comme des cas particuliers de graphes bipartis, de graphes d'intervalles, ou encore les split-graphes. Nous présentons quelques propriétés qui peuvent être utilisées pour la résolution et/ou l'approximation de MSCP.

3.1 Graphes quelconques

Nous présentons ici quelques bornes pour des graphes quelconques basées sur les paramètres $n, m, \chi(G)$ et $s(G)$. Dans (Thomassen *et al.* (1989)) les auteurs se sont basés sur le nombre d'arêtes m pour fournir des bornes inférieure et supérieure de MSCP. Ils montrent que pour tout m , la borne inférieure proposée ($\lceil \sqrt{8m} \rceil \leq \sum(G)$) peut être atteinte par un graphe biparti colorié avec 2 couleurs.

Concernant la borne supérieure, les auteurs ont démontré que pour une coloration valide c et une seconde coloration valide c' (obtenue par une permutation des deux classes de couleur X_1 et X_2 dans c), on a $\sum(G, c) + \sum(G, c') \leq 3(m+1)$. Donc $\sum(G) \leq \min \{ \sum(G, c), \sum(G, c') \} \leq \frac{3(m+1)}{2}$. On obtient,

$$\lceil \sqrt{8m} \rceil \leq \sum(G) \leq \left\lceil \frac{3(m+1)}{2} \right\rceil$$

Une autre borne inférieure est basée sur $\chi(G)$. Une somme coloration avec $s(G)$ couleurs est au moins égale à : $\sum_{i=1}^{i=s(G)} i + (n - s(G)) = n + \frac{s(G)(s(G)-1)}{2}$. Comme $\chi(G) \leq s(G)$, on obtient,

$$n + \frac{\chi(G)(\chi(G)-1)}{2} \leq \sum(G)$$

Pour toute k -coloration, l'inégalité $\frac{1+\dots+k}{k} \cdot \frac{|X_1|+\dots+|X_k|}{k} \geq \frac{\sum(G,c)}{k}$ est vérifiée. Donc pour $k = \chi(G)$, on obtient la borne supérieure suivante : $\sum(G) \leq \sum(G, c) \leq \frac{n(\chi(G)+1)}{2}$.

La somme coloration pour toute $s(G)$ -coloration est au moins égale à $\frac{s(G)(s(G)+1)}{2}$. On obtient,

$$\frac{s(G)(s(G)+1)}{2} \leq \sum(G) \leq \frac{n(\chi(G)+1)}{2}$$

De plus, Malafiejski (Malafiejski (2004)) montre que pour tout graphe G , la force du graphe $s(G)$ est

bornée comme suit :

$$\chi(G) \leq s(G) \leq \left\lceil \sqrt{n(\chi(G)+1)} \right\rceil$$

Pour GCP, la borne $\chi(G) \leq \Delta(G) + 1$ est immédiatement déduite. Cette borne est atteinte pour un graphe complet et un cycle impair. La même borne reste valable pour la force du graphe $s(G)$.

Théorème 1 (Hajiabolhassan *et al.* (2000)) *Soit G un graphe connexe et $\Delta(G)$ son degré. On a $s(G) \leq \Delta(G) + 1$. En outre, $s(G) = \Delta(G) + 1$ pour un graphe complet et un cycle impair.*

Dans la suite, nous présentons quelques résultats pour MSCP concernant des graphes particuliers.

3.2 Arbres

Pour colorier un arbre T on utilise au moins 2 couleurs, donc $\sum(T) \geq n+1$. Un arbre est un graphe dont on peut diviser l'ensemble de sommets V en deux sous-ensembles V_1 contenant des sommets de couleur 1 et V_2 contenant ceux de couleur 2. Dans le pire des cas $|V_2| = \lfloor \frac{n}{2} \rfloor$. Par conséquent, en utilisant 2 couleurs, la somme chromatique ne peut pas dépasser $\lceil \frac{n}{2} \rceil + 2 \lfloor \frac{n}{2} \rfloor$.

Théorème 2 (Malafiejski (2004)) *Soit $T = (V, E)$ un arbre ayant n sommets avec $n > 1$. Alors $n+1 \leq \sum(T) \leq \lfloor \frac{3n}{2} \rfloor$. De plus, pour tout entier naturel l vérifiant $n+1 \leq l \leq \lfloor \frac{3n}{2} \rfloor$ il existe un arbre avec $|V| = n$ et $\sum(T) = l$.*

Pour cette famille de graphes, il existe un algorithme en temps polynomial pour résoudre MSCP (Kroon *et al.* (1997)).

3.2.1 Algorithme de Kroon

Cet algorithme utilise une technique de programmation dynamique. On enracine l'arbre en un nœud quelconque. L'algorithme garde en mémoire, pour chaque nœud, les informations suivantes :

- $K_1(v)$, représentant le coût minimum de coloration du sous-arbre enraciné au nœud v (première coloration).
- $C(v)$, représentant la couleur utilisée pour le nœud v .
- $K_2(v)$, représentant le coût minimum de coloration du sous-arbre enraciné au nœud v , si celui ci est colorié différemment de $C(v)$

Algorithm 1: Calcul de $K(v, c)$

```

 $K := \sum_{i=1}^d K_1(v_i)$ 
for  $c := 1$  to  $d + 1$  do
   $\delta(c) := K$ 
for  $i := 1$  to  $d$  do
  if  $C(v_i) \leq d + 1$  then
     $\delta(C(v_i)) := \delta(C(v_i)) + K_2(v_i) - K_1(v_i)$ 
for  $c := 1$  to  $d + 1$  do
   $K(v, c) := c + \delta(c)$ 

```

(seconde coloration).

Si v est une feuille, alors $K_1(v) = C(v) = K_2(v) = 1$. Soit $K(v, c)$ le coût de la plus petite coloration du sous-arbre enraciné en v , quand v est colorié avec c et que tous les sous-arbres enracinés aux fils du nœud v sont coloriés avec le plus petit coût de coloration possible. Si v est un nœud interne ayant d fils, v_1, \dots, v_d , les valeurs $K_1(v_i)$, $C(v_i)$, $K_2(v_i)$ sont déterminées récursivement. Ensuite, les coûts de coloration $K(v, c)$ pour $c = 1, \dots, d + 1$ sont déterminés par l'équation :

$$K(v, c) = c + \sum_{i:C(v_i) \neq c} K_1(v_i) + \sum_{i:C(v_i) = c} K_2(v_i) \quad (1)$$

pour $c = 1, \dots, d + 1$

Notons que l'équation (1) peut être calculée en temps linéaire par Algorithm 1.

Étant donnée les coûts de coloration $K(v, c)$ avec $c = 1, \dots, d + 1$, les valeurs $K_1(v)$, $C(v)$ et $K_2(v)$ sont calculées par les formules suivantes :

$$K_1(v) = \min \{K(v, c) \mid c = 1, \dots, d + 1\} \quad (2)$$

$$C(v) = \operatorname{argmin} \{K(v, c) \mid c = 1, \dots, d + 1\} \quad (3)$$

$$K_2(v) = \min \{K(v, c) \mid c = 1, \dots, d + 1; c \neq C(v)\} \quad (4)$$

L'équation (2) détermine le coût minimal de coloration du sous-arbre enraciné en v et l'équation (3) détermine la couleur correspondante. Enfin, l'équation (4) détermine le coût d'une seconde coloration du sous-arbre enraciné en v . La complexité de Algorithm 1 est en $O(d)$ où d est le nombre de fils du nœud v . Donc, la complexité de l'algorithme de Kroon est proportionnelle au nombre d'arêtes m . Comme $m = n - 1$, la complexité de l'algorithme est en $O(n)$.

3.2.2 Algorithme de Tao et al.

Dans (Jiang & West (1999)) les auteurs ont démontré que pour un entier naturel k , il existe un arbre T_k tel

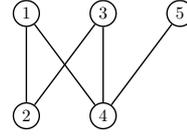


Figure 4: Exemple de graphe biparti chaîné

que $s(T_k) = k$ et $\Delta(T_k) = 2k - 2$. Pour construire un tel arbre, on construit d'abord des sous-arbres T_i^j vérifiant aussi la précédente propriété. Cette construction est basée sur un ordre de paires d'entiers naturels (i, j) suivant la règle : $(h, l) \prec (i, j)$ si et seulement si $(h + l < i + j)$ ou $(h + l = i + j$ et $l < j)$. L'arbre souhaité $T_k = T_k^1$, est obtenu grâce à Algorithm 2.

Algorithm 2: Algorithme de construction des T_i^j

Initialisation : $T_1^1 = \{1\}$ de couleur 1.

Construction de T_i^j :

Considérer u comme la racine de T_i^j de couleur i .

for $k := 1$ **to** $i + j - 1$ **do**

if $i \neq k$ **then**

 Lier deux exemplaires de $T_k^{\lceil \frac{i+j-1}{2} \rceil}$ à la racine u

3.3 Graphes bipartis

MSCP est plus difficile que le problème de coloration classique GCP pour les graphes bipartis. Il se résout en temps polynomial pour des graphes bipartis particuliers tels que ceux de degré $\Delta \leq 3$ (Giaro et al. (2002)), mais reste NP-Difficile pour les graphes planaires bipartis de degré $\Delta \leq 5$ (Giaro et al. (2002)). Il existe plusieurs schémas d'approximation pour MSCP pour cette classe de graphes (Bar-Noy et al. (1998)), (Bar-Noy & Kortsarz (1998))(Giaro et al. (2002)). Cependant, Bar-Noy et Kortsarz (Bar-Noy & Kortsarz (1998)) montrent qu'il existe $\epsilon > 0$ tel qu'on ne peut pas déterminer un schéma d'approximation de ratio $1 + \epsilon$ en temps polynomial pour MSCP pour les graphes bipartis, à moins que $P = NP$.

Définition 1 Un graphe biparti $G = (V_1, V_2; E)$ est appelé graphe biparti chaîné si pour toute paire de sommets $\{x_i, x_j\}$ dans V_1 , on a $N(x_i) \subseteq N(x_j)$, ou $N(x_j) \subseteq N(x_i)$. Autrement dit, il existe un ordre des sommets de V_1 , $x_{\pi_1}, \dots, x_{\pi_{n'}}$, où $|V_1| = n'$, tel que $N(x_{\pi_i}) \subseteq N(x_{\pi_{i+1}})$.

La figure 4 montre un exemple de graphe biparti chaîné. En effet, soit $V_1 = \{1, 3, 5\}$ et $V_2 = \{2, 4\}$, en considérant les sommets de V_1 dans l'ordre $(5, 1, 3)$, on obtient $N(5) \subseteq N(1) \subseteq N(3)$.

Théorème 3 (Salavatipour (2003)) *La somme chromatique d'un graphe biparti chaîné peut être calculée en temps polynomial $O(n + m)$.*

Pour un graphe biparti $G = (V_1, V_2; E)$ avec un couplage M couvrant V_1 , le problème est facile. On considère $V_2^M \subset V_2$ l'ensemble de tous les sommets couverts par M , et c une meilleure coloration utilisant au moins 3 couleurs. Alors :

$$\begin{aligned} \sum(G) &= \sum_{\{v_1, v_2\} \in M} (c(v_1) + c(v_2)) + \sum_{v \in V_2 \setminus V_2^M} c(v) \\ &> 3|V_1| + |V_2| - |V_1| = n + |V_1| \end{aligned} \quad (5)$$

Or il existe une 2-coloration de G avec une somme coloration qui vaut : $n + |V_1|$, ce qui constitue une contradiction. D'où le théorème suivant :

Théorème 4 (Malafiejski (2004)) *Soit $G = (V_1, V_2; E)$ un graphe biparti avec un couplage M couvrant V_1 . Toute coloration c de G donnant la somme chromatique $\sum(G)$ utilise au plus 2 couleurs et $\sum(G) = n + |V_1|$.*

3.4 Split-graphe

Rappelons ici quelques notions sur les split-graphes :

- Un split-graphe $G = (C, I; E)$ est un k_I -split-graphe si le degré de chaque sommet de I est au plus k , de même il est k_C -split-graphe si pour chaque sommet v de C le nombre d'arêtes adjacentes à v avec extrémité dans I est au plus k .
- Un graphe est un k -split-graphe s'il est k_C -split-graphe ou k_I -split-graphe.

Salavatipour (Salavatipour (2003)) montre que MSCP est NP-Difficile pour les split-graphes et donne des algorithmes de résolution pour les k_C -split-graphes et k_I -split-graphes.

Lemme 1 *Pour tout split-graphe $G = (C, I; E)$ où $|C| = n_C$ et $|I| = n_I$, on a :*

- $n_C \leq s(G) \leq n_C + 1$.
- $n_I + \frac{n_C(n_C+1)}{2} \leq \sum(G) \leq \frac{n_C(n_C+1)}{2} + n_C + n_I$.
- Si $s(G) = n_C + 1$ alors $\sum(G) = \frac{n_C(n_C+1)}{2} + n_C + n_I$.

Comme énoncé dans le lemme 1, $\sum(G)$ est connue lorsque $s(G) = n_C + 1$. Nous nous intéressons alors au calcul de $\sum(G)$ pour $s(G) = n_C$. Dans ce cas, la clique C , est coloriée avec les couleurs $1, 2, \dots, n_C$. Pour une coloration fixée de C , une coloration optimale de I est obtenue en $O(n_I)$ en attribuant à chacun de ses sommets la plus petite couleur disponible. Dans le cas où G est un k_I -split-graphe, comme $\Delta(I) \leq k$, les couleurs attribuées aux sommets de I , valent au plus $k + 1$, dans une solution optimale. De plus, seules les couleurs $1, 2, \dots, k$ attribuées aux sommets de C , influencent les couleurs, que pourront prendre les sommets de I . Il suffit donc d'énumérer les $O(n_C^k)$ affectations des couleurs $1, 2, \dots, k$ aux sommets de C pour calculer une coloration optimale de G en $O(n_I n_C^k)$.

Théorème 5 (Salavatipour (2003)) *Soit $G = (C, I; E)$ un k_I -split-graphe, pour un k fixé, alors MSCP se résout en temps polynomial $O(n_I n_C^k)$.*

Supposons maintenant que G soit un k_C -split-graphe. On note v_{c_1} le sommet de C de couleur 1 dans une solution optimale et $N_I(v_{c_1}) = I \cap N(v_{c_1})$. Dans (Salavatipour (2003)), l'auteur montre que les sommets de $N_I(v_{c_1})$ sont tous de couleur au plus égale à $k + 1$. De plus tous les sommets de $I \setminus N_I(v_{c_1})$ sont de couleur 1. On en déduit que pour v_{c_1} fixé, l'optimalité de la solution dépend de la coloration de $N_I(v_{c_1})$ et de $C \setminus \{v_{c_1}\}$. Or, comme G est un k_C -split-graphe, $|N_I(v_{c_1})| \leq k$ et les couleurs possibles pour $N_I(v_{c_1})$ sont $2, 3, \dots, k + 1$. Trouver la solution optimale revient alors à fixer v_{c_1} , énumérer les colorations possibles de $N_I(v_{c_1})$ et pour chacune d'entre elles résoudre en $O(n_C)$ un problème d'affectation simple, des couleurs $2, 3, \dots, n_C$ aux sommets de $C \setminus \{v_{c_1}\}$. Il y a n_C possibilités pour fixer v_{c_1} , la solution optimale est alors calculée en $O(n_C^2)$.

Théorème 6 (Salavatipour (2003)) *Soit $G = (C, I; E)$ un k_C - split - graphe, pour un k fixé, alors MSCP se résout en temps polynomial $O(n_C^2)$.*

Corollaire 1 *MSCP se résout en temps polynomial $O(n_I n_C^k)$ pour les k -split-graphes.*

4 Algorithmes gloutons pour MSCP

Nous avons vu dans la section 3, un grand nombre de propriétés utiles à la résolution de MSCP pour des familles de graphes spécifiques ainsi que des bornes théoriques larges pour des graphes quelconques. Nous aborderons maintenant la résolution de MSCP pour des graphes quelconques d'un point de vue algorithmique. Dans ce but, nous proposons des

algorithmes gloutons permettant de calculer à la fois une borne supérieure et une borne inférieure.

La présentation générale d'un algorithme glouton comprend quatre éléments de base : une fonction objectif f , une configuration C , un ensemble de candidats L et une règle de sélection r . Nous allons définir ces éléments dans le cadre de la coloration de graphe.

- *Fonction objectif* : pour GCP la fonction objectif consiste à minimiser le nombre de couleurs, alors que pour MSCP il s'agit de minimiser la somme des couleurs.
- *Configuration* : une configuration représente une coloration valide partielle ou complète.
- *Candidat* : un candidat est un sommet non colorié.
- *Règle de sélection* : la règle de sélection est une stratégie de choix du meilleur candidat local suivant un certain critère.

Nous nous sommes intéressés à l'étude et l'amélioration des critères et règles de sélection utilisés par les algorithmes gloutons de la littérature, DSATUR (Bréaz (1979)) et RLF (Leighton (1979)). Une version révisée de RLF est publiée par Chiarandini (Chiarandini (2005)).

Nous rappelons les deux algorithmes gloutons DSATUR et RLF et présentons la nouvelle variante MRLF.

4.1 Dsat

Soit C une configuration et v un sommet non colorié. On note par $nbV\text{oisin}NC(v)$ le nombre de voisins non coloriés de v . Le nombre de couleurs différentes utilisées par les voisins de v est appelé le *degré de saturation* de v et est noté $dsat(v)$.

DSATUR utilise $dsat$ comme critère de sélection. La règle de sélection de DSATUR consiste à choisir un sommet non colorié v qui a le plus grand $dsat(v)$, et en cas d'ex æquo sera choisi celui qui a le plus grand $nbV\text{oisin}NC(v)$. La complexité de DSATUR est en $O(n^2)$.

4.2 De RLF à MRLF

L'idée de RLF est de colorier le maximum de sommets avec une couleur c avant d'en utiliser une nouvelle.

Soit X_1, \dots, X_i où X_i est la classe de couleur i en cours de construction, et les candidats sont les sommets v non coloriés dont la valeur $cmin(v)$ est égale à i . Pour chaque candidat v , l'ensemble de ses voisins w non coloriés se divise en deux; l'ensemble

γ^{cminC} qui contient les sommets w dont la valeur $cmin(w)$ est égale à i , (on note par $nbCminC(v)$ le cardinal de γ^{cminC}) et l'ensemble γ^{cminNC} qui contient les sommets dont la valeur $cmin(w)$ est différente de i (on note par $nbCminNC(v)$ le cardinal de γ^{cminNC}). La coloration du sommet v par son $cmin(v) = i$ influencera tous les sommets de γ^{cminC} et aucun sommet de γ^{cminNC} . RLF utilise $nbCminNC$ comme premier critère de sélection et $nbCminC$ comme second critère de sélection. Sa règle de sélection est basée sur le choix du sommet v qui a le plus grand $nbCminNC$ et en cas d'ex æquo sera choisi celui qui a le plus petit $nbCminC$. La complexité de cet algorithme est en $O(n^3)$.

Cette nouvelle variante cherche à minimiser l'influence de la coloration d'un sommet avant une coloration définitive. En fait cette variante est une extension de RLF, son critère de sélection est le même que celui de RLF mais la règle de sélection est une nouvelle combinaison entre les deux critères $nbCminC$ et $nbCminNC$ (Li et al. (2009a)). La complexité de cette nouvelle variante est en $O(n^3)$.

5 Borne inférieure pour MSCP (Li et al. (2009b))

Comme nous l'avons vu dans la section 3, on trouve dans la littérature des bornes inférieures théoriques pour MSCP, concernant certaines familles de graphes, mais à notre connaissance il n'existe aucune étude sur une approche algorithmique permettant de calculer une borne inférieure pour un graphe quelconque. Dans cette section nous présentons quelques nouveaux résultats sur le calcul de bornes inférieures pour MSCP.

Soit $G = (V, E)$ un graphe et $G' = (V, E')$ un graphe partiel de G . Toute coloration valide de G est une coloration valide de G' , alors Proposition 1 est immédiatement déduite.

Proposition 1 *Si G' est un graphe partiel de G , alors la somme chromatique de G' est une borne inférieure de G .*

Pour calculer une borne inférieure de MSCP, il suffit de déterminer des graphes partiels pour lesquels nous pouvons facilement calculer la somme chromatique. Pour cela nous nous sommes intéressés à deux familles de graphes pour lesquelles la somme chromatique est calculée en temps polynomial, les arbres et les cliques.

5.1 Borne inférieures par extraction d'arbre

Comme mentionné précédemment (Section 3), la résolution du MSCP pour les arbres se fait en temps polynomial (Kroon et al. (1997)). Tout arbre

couvrant T extrait de G est un graphe partiel de G . Donc $\sum(T)$ est une borne inférieure de la somme chromatique de G . Trouver la meilleure somme chromatique en utilisant l'extraction d'arbres consiste à trouver un arbre couvrant T extrait de G tel que $\sum(T)$ est la plus grande de toutes les extractions d'arbres couvrants possibles. On note par $\sum_{tree}(T)$ cette meilleure solution. D'après Théorème 2 on a $n \leq \sum_{tree}(G) \leq \overline{LB}_b$ où $\overline{LB}_b = \lfloor \frac{3n}{2} \rfloor$.

5.2 Borne inférieure par décomposition en cliques

Nous considérons une clique de taille k . Il existe une façon unique de la colorier qui nécessite k couleurs et la somme associée est $\sum_1^k i = k(k+1)/2$. Nous nous sommes basés sur cette observation pour fournir une borne inférieure issue de la décomposition du graphe G en cliques. Une décomposition du graphe en cliques constitue un graphe partiel dont la somme chromatique est calculée en temps polynomial. La détermination d'un tel graphe avec la meilleure évaluation possible est un problème d'optimisation combinatoire appelé CDMSCP (Li *et al.* (2009b)). Il s'agit d'une extension du problème NP-Complet de partition en cliques (Garey & Johnson (1990)). Résoudre CDMSCP revient à trouver une partition V_1, \dots, V_l de V , telle que pour tout $i \in 1, \dots, l$ le sous-graphe induit par V_i est une clique et $\sum \frac{|V_i|(|V_i|+1)}{2}$ est maximale. La solution de CDMSCP est notée $\sum_{clique}(G)$.

On considère le graphe complémentaire \overline{G} . Toute clique dans G est un stable dans \overline{G} . Donc toute coloration de \overline{G} fournit une décomposition de \overline{G} en stables (classes de couleur) X_1, X_2, \dots, X_l et donc une décomposition en cliques de G . Soit K_i la clique de taille k_i dans G associée au stable X_i dans \overline{G} alors $k_i = x_i$. Suivant ce principe, trouver une borne inférieure pour MSCP revient à trouver une partition V_1, \dots, V_l de V telle que pour tout $i \in \{1, \dots, l\}$ V_i est un stable (une classe couleur i) et $\sum \frac{|V_i|(|V_i|+1)}{2}$ est maximale. Pour obtenir les différents stables de \overline{G} , nous utilisons la nouvelle variante MRLF. L'algorithme est exécuté n fois et on retient la plus grande solution comme borne inférieure LB_c de G .

Si G possède une clique maximum de taille ω , toute décomposition Λ en cliques a au plus $\lfloor \frac{n}{\omega} \rfloor$ cliques de taille ω . La somme chromatique de chacune de ces cliques est égale à $\frac{\omega(\omega+1)}{2}$. Les r sommets restants forment dans le meilleur cas une clique dont la somme chromatique est égale à $\frac{r(r+1)}{2}$. On en déduit la proposition suivante :

Proposition 2 *Soit ω la taille de la clique maximum de G , $r = n - \omega \lfloor \frac{n}{\omega} \rfloor$ alors $\sum_{clique}(G)$ est majorée par $\overline{LB}_c = \frac{\omega(\omega+1)}{2} \lfloor \frac{n}{\omega} \rfloor + \frac{r(r+1)}{2}$.*

6 Résultats numériques

Tableau 1 résume les résultats numériques sur quelques instances de la librairie DIMACS présentées dans (Kokosiński & Kwarciány (2007)). Pour chaque instance dans Tableau 1, on note par \overline{LB}_b l'évaluation par excès de la borne inférieure définie dans la section 5.1, $LB_{th} = \max\{\lfloor \sqrt{8m} \rfloor, n + \frac{\chi(G)(\chi(G)-1)}{2}\}$ la borne théorique mentionnée dans la section 3, LB_c la borne inférieure obtenue par la décomposition en cliques, \overline{LB}_c l'évaluation par excès de la borne inférieure définie dans Proposition 2. On note par UB la meilleure borne supérieure obtenue par MRLF, et par UB_k la meilleure borne supérieure dans (Kokosiński & Kwarciány (2007)). Finalement, on note par $Gap = \frac{UB-LB_c}{LB_c}$, le gap entre la meilleure borne inférieure obtenue par la décomposition en cliques et la borne supérieure obtenue par MRLF.

Ces résultats montrent que la borne LB_c , est meilleure que la borne inférieure théorique LB_{th} . Cette dernière se base juste sur le nombre de sommets, le nombre d'arêtes, et le nombre chromatique, alors que LB_c est obtenue par notre algorithme glouton MRLF. De plus, notre borne supérieure obtenue également par MRLF est de meilleure qualité que celle présentée dans (Kokosiński & Kwarciány (2007)) utilisant un algorithme génétique basé sur les croisements GPX et CEX, avec un nombre d'itérations de 10000. Enfin, nous proposons un meilleur intervalle de la solution optimale pour MSCP. L'optimum est atteint pour 3 instances parmi celles présentées (huck, queen5-5, queen7-7).

7 Conclusion

Dans cet article, nous avons présenté une synthèse des travaux publiés sur MSCP. Ensuite nous avons présenté une nouvelle famille d'algorithmes gloutons MRLF basée sur de nouveaux critères et règles de sélection. Ces algorithmes nous ont permis de calculer une borne supérieure mais aussi une borne inférieure par coloration du graphe complémentaire. Les résultats numériques montrent que notre approche améliore les résultats de la plupart des instances mentionnées dans littérature.

REMERCIEMENTS

Ce travail de recherche est financé par le conseil régional de Picardie dans le cadre du projet SCOOP.
*Bibliography

Bar-Noy, A., & Kortsarz, G. 1998. Minimum color

$G(V, E)$	n	m	ω	$\chi(G)$	\overline{LB}_b	LB_{th}	LB_c	\overline{LB}_c	UB_k	UB	Gap
<i>anna</i>	138	493	11	11	207	193	272	813	281	277	0,02
<i>david</i>	87	406	11	11	130	142	234	517	243	241	0,03
<i>huck*</i>	74	301	11	11	111	129	243	432	243	244	0,01
<i>jean</i>	80	254	10	10	120	125	216	440	218	217	0,01
<i>queen5 - 5*</i>	25	160	5	5	37	36	75	75	75	75	0
<i>queen6 - 6</i>	36	290	6	7	54	57	126	126	138	138	0,1
<i>queen7 - 7*</i>	49	476	7	7	73	70	196	196	196	196	0
<i>queen8 - 8</i>	64	728	8	9	96	100	288	288	302	303	0,06
<i>games120</i>	120	638	9	9	180	156	442	591	460	446	0,01
<i>miles250</i>	128	387	8	8	192	156	316	576	347	334	0,06
<i>miles500</i>	128	1170	20	20	192	318	677	1296	762	715	0,06
<i>myciel3</i>	11	20	2	4	16	17	16	16	21	21	0,32
<i>myciel4</i>	23	71	2	5	34	33	34	34	45	45	0,33
<i>myciel5</i>	47	236	2	6	70	62	70	70	93	93	0,33
<i>myciel6</i>	95	755	2	7	142	116	142	142	189	189	0,34
<i>myciel7</i>	191	2360	2	8	286	219	286	286	382	381	0,34

Tableau 1: Comparaison avec la littérature

- sum of bipartite graphs. *J. Algorithms*, **28**(2), 339–365.
- Bar-Noy, A., Bellare, M., Halldórsson, M., Shachnai, H., & Tamir, T. 1998. On chromatic sums and distributed resource allocation. *Inf. Comput.*, **140**(2), 183–202.
- Bar-Noy, A., Halldórsson, M., Kortsarz, G., Salman, R., & Shachnai, H. 2000. Sum Multicoloring of Graphs. *Journal of Algorithms*, **37**, 422–450.
- Brélaz, D. 1979. New methods to color the vertices of a graph. *Commun. ACM*, **22**(4), 251–256.
- Chiarandini, M. 2005. *Stochastic local search methods for highly constrained combinatorial optimisation problems*. Ph.D. thesis, Darmstadt University of Technology, Darmstadt, Germany.
- Garey, R., & Johnson, S. 1990. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co.
- Giario, K., Janczewski, R., Kubale, M., & Malafiejski, M. 2002. A 27/26-Approximation Algorithm for the Chromatic Sum Coloring of Bipartite Graphs. *Pages 135–145 of: APPROX '02*.
- Hajiabolhassan, H., Mehrabadi, M. L., & Tusserkani, R. 2000. Minimal coloring and strength of graphs. *Discrete Math.*, **215**(1-3), 265–270.
- Halldórsson, M., & Kortsarz, G. 1999. Multicoloring Planar Graphs and Partial k-Trees. *Pages 73–84 of: RANDOM-APPROX '99: Proceedings of the Third International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*.
- Halldórsson, M., Kortsarz, G., Proskurowski, A., Salman, R., Shachnai, H., & Telle, J.A. 2003. Multicoloring trees. *Inf. Comput.*, **180**(2), 113–129.
- Jansen, K. 1997. The Optimum Cost Chromatic Partition Problem. *Pages 25–36 of: CIAC*.
- Jansen, K. 2000. Approximation results for the optimum cost chromatic partition problem. *J. Algorithms*, **34**(1), 54–89.
- Jiang, T., & West, D.B. 1999. Coloring of trees with minimum sum of colors. *J. Graph Theory*, **32**(4), 354–358.
- Kokosiński, K., & Kwarciany, K. 2007. On Sum Coloring of Graphs with Parallel Genetic Algorithms. *Pages 211–219 of: ICANNGA '07*.
- Kovács, A. 2004. Sum-Multicoloring on Paths. *Pages 68–80 of: STACS*.
- Kroon, L.G., Sen, A., Deng, H., & Roy, A. 1997. The Optimal Cost Chromatic Partition Problem for Trees and Interval Graphs. *Pages 279–292 of: WG '96*.
- Kubicka, E. 1989. *The chromatic sum of a graph*. Ph.D. thesis, Western Michigan University, USA.
- Kubicka, E., & Schwenk, A. J. 1989. An introduction to chromatic sums. *Pages 39–45 of: CSC '89: Proceedings of the 17th conference on ACM Annual Computer Science Conference*.
- Leighton, F.T. 1979. A graph Coloring Algorithm for Large Scheduling Problems. *Journal of research of the national institute of standards and technology*, **84**(6), 251–256.
- Li, Y., Lucet, C., Moukrim, A., & Sghiouer, K. 2009a. Greedy Algorithms for the Minimum Sum Coloring Problem. *In: International Workshop: Logistics and transport*.

- Li, Y., Lucet, C., Moukrim, A., & Sghiouer, K. 2009b. Lower bounds for the minimum sum coloring problem, *Rapport interne, Université de technologie de Compiègne, France*.
- Malafiejski, M. 2004. *Sum coloring of graphs, in Graph Colorings (M.Kubale, Ed.)*.
- Marx, D. 2004. Graph Coloring Problems and Their Applications in Scheduling. *Pages 1–2 of: in Proc. John von Neumann PhD Students Conference*.
- Salavatipour, M.R. 2003. On sum coloring of graphs. *Discrete Appl. Math.*, **127**(3), 477–488.
- Supowit, K. J. 1987. 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.
- Thomassen, C., Erdős, P., Alavi, Y., Malde, P.J., & Schwenk, A.J. 1989. Tight bounds on the chromatic sum of a connected graph. *J. Graph Theory*, **13**, 353–357.