



HAL
open science

Modélisation des trois contraintes majeures d'ordonnement pour une résolution optimum

Xavier Cousin

► **To cite this version:**

Xavier Cousin. Modélisation des trois contraintes majeures d'ordonnement pour une résolution optimum. 2023. hal-04112310

HAL Id: hal-04112310

<https://hal.science/hal-04112310>

Preprint submitted on 31 May 2023

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.



Distributed under a Creative Commons Attribution 4.0 International License

Modélisation des trois contraintes majeures d'ordonnement pour une résolution optimum.

Résumé : Cet article introduit une modélisation permettant de poser d'une façon conjointe les trois contraintes majeures de problème de recherche opérationnelle telles que la disjonction temporelle de tâches, l'ordonnement entre tâches ou l'affectation cumulée de ressources avec un plafond ou un plancher. Jusqu'à ce jour, la résolution conjointe des trois types de contraintes n'est pas solvable dans un même modèle de résolution. Cette modélisation permet d'appliquer les méthodes décrites dans mes articles précédents pour transformer le problème d'ordonnement vers un système de résolution se reposant sur un problème SAT.

Mots-clefs : Modélisation, Ordonnement, Contraintes, problème NP-complet, théorie de graphes, problème SAT

Summary : This article introduces a model allowing to set in a joint way the three major constraints of operational research problem such as the temporal disjunction of tasks, the scheduling between tasks or the cumulative allocation of resources with a ceiling or a floor. Until now, the joint resolution of the three types of constraints is not solvable in the same resolution model. This modeling makes it possible to apply the methods described in my previous articles to transform the scheduling problem into a resolution system based on a SAT problem.

Keywords: Modeling, Scheduling, Constraints, NP-complete problem, Graph theory, SAT problem

1. Introduction

Il existe plusieurs modélisations pour des problèmes de recherche opérationnelle. Cela est vu comme un système de contraintes diverses posées sur des activités et ressources dont leur utilisation est à optimiser. En réalité, aujourd'hui toutes les résolutions essaient de s'approcher de la meilleure solution sans forcément l'atteindre. La problématique vient de l'aspect discret et non continu des problèmes.

Dans cet article, issu de réflexions passées [4], [5], nous repartons de la théorie des graphes où on peut modéliser ces problèmes en mathématique discrète en recherchant la meilleure solution quitte à mettre un système de priorité de contraintes. Pour cela, nous fournissons un ensemble d'outils représentant trois contraintes majeures pour appliquer la résolution présentée dans les articles [6], [7], [8].

La notion d'ordonnement, d'affectation ou de recherche opérationnelle permet de poser des contraintes temporelles où deux tâches par exemple ne peuvent pas être soit sur le même créneau temporel, soit ordonnée l'une après l'autre. On ajoute la possibilité de poser une contrainte de cumul de tâches ou de ressources pouvant être sur le même créneau temporel dans une limite maximale ou minimale. C'est l'exemple que nous trouvons sur des problèmes d'emploi du temps comme vu en [3].

Une première modélisation est faite par la pose d'un graphe G et résolue par la coloration de sommets permettant d'introduire les contraintes de disjonction. En un second temps, nous ajoutons les contraintes d'ordonnement par le mécanisme de somme cartésienne du graphe G et d'une clique K_q . La modélisation finale aboutit à transformer le graphe précédent vers un problème SAT ajoutant les contraintes de cumul.

Avant tout, nous rappelons quelques conventions de nommage pour pouvoir décrire rapidement ce qu'est un problème d'ordonnement modélisé en coloration de sommets puis nous indiquons comment sont modélisées les trois types de contraintes en gardant cette possibilité de résoudre le problème sur un seul système de résolution. Nous supposons que le lecteur connaît un minimum de théorie des graphes.

2. Rappels et conventions

- $G = (X, A)$ est le graphe qui a comme ensemble de sommets X et comme ensemble d'arêtes A .
- $|Y|$ est le cardinal d'un ensemble Y .
- On note $d_G(x)$ le degré du sommet x dans le graphe G .
- La clique est le nombre d'entités liées entre elles deux à deux. Une clique de sommets est un ensemble de sommets liés entre eux deux à deux par des arêtes.
- K_q représente la clique d'ordre q .
- On nomme N le nombre de sommets de G (ou $|X|$) et M le nombre d'arêtes (ou $|A|$).

- L'indice chromatique de G correspondant au nombre minimum de couleurs pour colorier fortement les arêtes.
- Le nombre chromatique de G correspond au nombre minimum de couleurs pour colorier fortement les sommets.
- \overline{G} est noté comme le complémentaire de G .
- $G_1 + G_2$ dénote la somme cartésienne de deux graphes (cf. [1] [2]).
- P , un ensemble ordonné de couleurs $(1, \dots, q)$, $q > 1$.

3. Ordonnement et affectation optimisée de ressources par la théorie des graphes

Nous pouvons trouver dans la littérature de nombreuses définitions d'un problème d'ordonnement. Ici ce problème et celui de sa résolution en recherche opérationnelle est un objectif d'affectation de ressources à des tâches ou activités en fonction de contraintes et se rapproche de celui défini dans le livre de Gondran et Minoux [11].

La résolution peut simplement indiquer la cohérence d'un système de tâches, contraintes et affectations possibles mais nous pouvons aussi y trouver une recherche d'optimisation sur ces affectations. L'exemple auquel nous pouvons se référer est celui d'emploi du temps dont celui scolaire [3]. Par rapport à ce qui a été fait jusqu'à aujourd'hui l'approche décrite permet une résolution sur une solution modélisant conjointement toutes les contraintes.

a. Modélisation et principe de base

Pour permettre de présenter ces trois types de contraintes et comment les poser, nous allons définir un exemple d'emploi du temps illustrant à chaque fois l'utilisation de ces types de contraintes. Nous passons par une modélisation qui introduit les éléments de base concrétisant les tâches, affectations et relation basique entre ces tâches en deux étapes : celle de simple graphe puis de graphe éclaté et étendu. Un passage facile ensuite en problème SAT autorise une résolution unique des problèmes posés en graphe.

i. *Illustration d'un simple exemple de problème d'emploi du temps*

Prenons un cas simple du problème d'emploi du temps suivant :

Soit les sept cours, illustrant des tâches suivants :

- cours 1 = (prof1, groupe1, matière1, salle TP) d'une durée d'une heure,
- cours 2 = (prof1, groupe2, matière1, salle TP) d'une durée d'une heure,
- cours 3 = (prof1, groupe3, matière1, salle TP) d'une durée d'une heure,
- cours 4 = (prof1, groupe4, matière1, salle TP) d'une durée d'une heure,
- cours 5 = (prof1, groupe5, matière1, salle TP) d'une durée d'une heure,
- cours 6 = (prof2, groupe5 & groupe6, matière2, amphi1) d'une durée de deux heures,
- cours 7 = (prof2, groupe6, matière2, amphi2) d'une durée de deux heures.

Associées nous trouvons les contraintes :

Contraintes de non ubiquité : un même professeur ne peut assurer un cours avec une même classe ou groupe au même moment. Nous typons ce genre de contrainte de disjonction.

Deux cours avec une même ressource (professeur, classe) auront deux configurations possibles : cours 1 puis cours 2 ou cours 2 puis cours 1 en tenant compte de leur durée respective.

Posons que le cours 7 doit se faire après le cours 6. Nous avons une contrainte de succession ou d'ordre.

Posons pour une raison de capacité d'accueil de la salle de TP, que nous n'ayons que trois groupes en même temps. Cela se traduit par trois cours sur cinq pour les cours de 1 à 5 peuvent se dérouler au même moment. Nous typons cette contrainte de contrainte cumulative en plafond ou en plancher.

En jouant sur ces cumuls, nous pouvons aussi introduire une fréquence comme le cours 7 se déroule une fois par jour.

L'allocation de ressources est celles des heures de début des cours sur une semaine divisée en créneaux d'une heure où chaque journée commence à 8h pour finir à 18h du lundi au vendredi.

ii. *Graphe originel G et palette de couleurs P*

Une première modélisation passe par un graphe G donnant les éléments de base où par exemple on doit colorier fortement les sommets définis dans une palette P finie et discrète tels que nous pouvons trouver dans les documents [3], [4], [5]. Cela donne des possibilités de modéliser des contraintes de disjonction temporelle. Nous y trouvons alors les questions :

- Que représentent les sommets ?
- Que représentent les arêtes ?
- Que représentent les couleurs ?

Pour un graphe de départ, les sommets sont les entités-tâches auxquelles on doit allouer un créneau temporel soit à une heure, à un jour ou autre. Ces sommets peuvent être des compositions, agrégats de ressources aboutissant à définir des tâches comme un cours qui est un agrégat de professeur, salle, groupe d'élèves et matière enseignée.

Les arêtes ici représentent les disjonctions entre ces sommets pour le choix de l'allocation.

Cela donne dans notre exemple que les cours sont des sommets, les couleurs à allouer, les heures possibles de début de cours et les arêtes les disjonctions entre deux débuts de cours.

Pour être plus concret pour notre palette P de couleurs, les couleurs possibles des sommets de G représentent les numéros d'heure correspondant aux nombres d'heures écoulées depuis le lundi 0h soit 0 à 23 pour lundi, de 24 à 47 pour mardi, etc. en ne gardant que ceux sélectionnable et éligible pour le problème posé.

On peut même avoir une multi-allocation pour une tâche répétitive (une fois par jour) donc une multi-coloration.

iii. *Graphe étendu G^e: éclatement (G x P) résultat du croisement du graphe G et de la palette P des couleurs*

De ce graphe, nous allons en créer un autre G^e par duplicata de graphe G autant de fois qu'il y a de couleurs dans P. Nous allons ajouter des arêtes inter-copies pour indiquer des disjonctions entre le couple généré (sommets, couleur) et par conséquent entre créneau de début des tâches pour les cours associés.

Dans le même esprit du document [7] liant le nombre chromatique de G et le nombre de stabilité du graphe étendu tel que la somme cartésienne G+Kq où q est le cardinal de la palette de couleurs, nous cherchons l'ensemble stable maximum dans G^e qui correspondra à une affectation de couleurs aux sommets de G, l'affectation pouvant même être multiple en cas de périodicité à trouver.

Dans notre exemple d'emploi du temps, nous avons les sommets {(cours, heure de début)} où on cherche l'ensemble maximum de ces associations qui n'ont pas de lien entre elles donnant un ensemble stable au sens de la théorie des graphes. Ces liens sont les arêtes traduisant les contraintes et posées de la manière que l'on aborde par la suite.

iv. *Problème SAT sous forme CNF*

Décrivons et rappelons comme illustré en [8], la transformation d'un graphe étendu en problème SAT sous sa forme normale conjonctive (CNF). On introduira des clauses suites aux arêtes et des clauses d'exclusivité pour les cliques connues.

$$\begin{aligned}
 \text{SAT}_i = & \{ \\
 & \bigwedge_{x \in X} \left(\bigwedge_{\substack{x_i \in G_i \subset X+Kq \\ x_j \in G_j \subset X+Kq \\ x_i, x_j \in Kq(x)}} (\neg x_i \vee \neg x_j) \right) \wedge \\
 & \bigwedge_{A_i \subset G+Kq} \left(\bigwedge_{(x_i, y_i) \in A_i} (\neg x_i \vee \neg y_i) \right) \wedge \\
 & \bigwedge_{Kq(x) \subset X+Kq} (x_1 \vee \dots \vee x_q) \\
 & \}
 \end{aligned}$$

Les diverses transformations à faire en arêtes sur G^e sont exposées dans les paragraphes suivants.

Les problèmes d'ordonnancement et de recherche opérationnelle peuvent être vus à travers la typologie suivante de contraintes donnant un kit de classes de contraintes. Cela rejoint la description faite dans le livre de Gondran et Minoux [12] paragraphe 3.3 p58 :

- Contraintes disjonctives
- Contraintes d'ordre ou de succession
- Contraintes de cumul

Détaillons leur pose et traduction dans les paragraphes ci-dessous.

b. Contraintes d'ordre et de succession

Les contraintes d'ordre sont aujourd'hui bien résolues à travers diverses méthodes dont la méthode PERT. Cependant dès que l'on mixte ces contraintes avec celles disjonctives, avec les méthodes classiques on retombe dans un problème NP-Complexe de résolution exponentielle en espace ou en temps en fonction de ses données.

On peut se ramener à une méthode pure de recherche d'un ensemble stable au sens fort en modélisant ces contraintes à travers un graphe étendu ou « éclaté » comme vu ci-dessus et présenté dans l'article [7] en les transformant en contraintes disjonctives. Cela nous permet de cumuler contraintes de succession et contraintes de disjonction même entre tâches qui ont une durée.

Transformation de contrainte d'ordre en disjonctives

Par exemple, on pose la contrainte $a \geq b + 2$, a et b sont représentés par des sommets dans un graphe G . Le graphe étendu dupliquera ce graphe G comme indiqué précédemment ; chaque instance correspond à une couleur, la palette de couleur étant totalement ordonnée.

Les instances d'affectation possibles de la palette étant ordonnées, $0, 1, \dots, i, \dots, q$, on transforme cette contrainte de succession en contraintes disjonctives ($a \neq b-k, \dots, a \neq b-1, a \neq b, a \neq b+1$) et en arêtes soit dans la même instance i de graphe G soit entre sommets de différentes instances j de G . Si G_i est l'instance i de G (ou la couleur i) et les sommets a_i et b_i les représentants de a et b de G pour la couleur i . Il y aura une arête entre les sommets a_i et b_i et entre les sommets a_i et b_{i+1} ainsi que a_i et b_{i-1} et sur les instance précédente (ici jusqu'à $b-k$). Bien entendu, on suppose que la $(i+1)^{i\text{ème}}$ instance correspond à une couleur-temps plus grand que la couleur-temps i .

Soit en résumé, $a \geq b+2$ se transforme en ($a \neq b-k, \dots, a \neq b-1, a \neq b, a \neq b+1$) à traduire en arêtes sur tout duplicata de G_i et G_j , deux à deux soit avec a_i instance de a sur $G_i : \bigcup_{i=1}^{|P|} \bigcup_{j=1}^{\min(i+1, |P|)}$ arête (a_i, b_j)

c. Contraintes disjonctives

Les contraintes les plus simples sont celles disjonctives et se posent simplement en termes d'arêtes entre sommets. S'il n'y a que ces contraintes, la résolution correspond à une coloration forte sur un graphe G avec une palette de couleur ordonnée ou non.

Contraintes de disjonction de non ubiquité.

Reprenons notre graphe G^e qui est constitué d'autant de duplicata de G que d'éléments de la palette. Entre chaque sommet x_i dupliqué, x_i représentant une tâche x au temps i , nous ne voulons que x n'apparaisse qu'à un seul moment. Pour cela nous créons des cliques entre les sommets x_i . Cela résulte à la somme cartésienne du graphe G et de la clique $K_{|P|}$. Par contre, G^e peut ne pas être cette somme cartésienne si nous voulons que cette tâche apparaisse à plusieurs moments comme sur plusieurs jours mais restreint à une seule fois par jour.

Contraintes de disjonction temporelle

En introduisant une palette ordonnée de couleurs de cardinal q avec le graphe G dont les sommets sont les tâches et les arêtes de G les disjonctions entre tâches, on crée un graphe G^e duplicata de q graphes G , on peut poser des contraintes de disjonction temporelles.

Illustrons ce problème sur une tâche x , sommet de G , qui ne doit apparaître qu'une fois par jour. Soit G_1 correspondant à la première de lundi, 8h par exemple, G_9 la dernière heure du lundi, G_{10} la première heure de mardi, etc. On suppose qu'il n'y a que 9 heures chaque jour.

Si on veut une disjonction par jour, on peut poser des cliques $K_9(x)$ sur les sommets des duplicata de x dans G_1 à G_9 soit une clique (x_1, x_2, \dots, x_9) , puis sur les duplicata de x de G_{10} à G_{18} soit une clique $(x_{10}, x_{11}, \dots, x_{18})$.

On veut qu'obligatoirement la tâche soit allouée chaque jour à un de ces neuf horaires. On verra dans le paragraphe (d) pour la transformation en problème SAT en forme normal conjonctive (CNF) que chaque telle clique doit avoir une clause indiquant qu'obligatoirement un unique sommet de cette clique doit avoir une affectation (valeur égale à vrai). On les appellera clause d'exclusivité.

En introduisant le graphe G^e , on peut aussi introduire des contraintes disjonctives sur des tâches avec durée. Ce cas de durée des tâches rejoint les contraintes de succession ou d'ordre. Décrivons les plus précisément dans le paragraphe ci-après.

Cas des contraintes disjonctives sur des tâches avec durée

Revenons sur notre problème de disjonction de tâches avec durée. Nous le transformons en contraintes de succession puis en disjonction élémentaires comme ci-dessus.

Prenons le problème de la durée des tâches. Soit deux tâches a et b , la contrainte $a \neq b$ avec durée se transforme en $a \geq b + \text{durée } b$ ou $b \geq a + \text{durée } a$ (qui équivaut à $a \leq b - \text{durée } a$). Par conséquent, on crée les contraintes de disjonction : $\{\forall i, a_i \neq b_i, a_i \neq b_{i+1}, \dots, a_i \neq b_{(i+\text{durée } b-1)}\} \cup \{a_i \neq b_{i-1}, \dots, a_i \neq b_{(i-\text{durée } a-1)}\}$ en ajoutant les arêtes entre les sommets illustrant les différentes instances des tâches a et b issues des duplicata de G (G_i).

d. Contraintes cumulatives (parfois nommées de capacité)

Il reste les contraintes additionnelles de cumul ou de capacité. On peut en distinguer de 2 types : de limite à minima et de limite à maxima.

Limite d'affectation a maxima des ressources

Rappel de c-couplage et de couplage

Un couplage est un ensemble stable d'arêtes, c'est-à-dire que les arêtes, deux à deux, sont disjointes et n'ont aucun sommet en commun. A un sommet, nous avons alors au maximum une seule arête appartenant au couplage. Le sommet fait office de cliques entre ses arêtes incidentes.

Dans le c-couplage, on émet à chaque sommet x une constante c où on autorise à avoir c arêtes dans le même couplage, c étant inférieur au degré de x .

On applique alors la transformation décrite dans [1] au chapitre 8 :

- on ajoute $d_G(x) - c$ sommets y_i ;
- chaque arête est dissociée du sommet x en dupliquant ce sommet x en x_i sommets ;
- chacun des nouveaux sommets y_i et x_i sont reliés entre eux.

La recherche du c-couplage maximum dans G est la recherche du couplage maximum dans ce graphe transformé et par conséquent un ensemble stable maximum.

Par exemple, sur un sommet de degré 6, on veut autoriser au maximum 4 arêtes dans le même couplage. Cela donne le schéma ci-dessous sur la transformation en recherche de couplage parfait (cf. Berge [1] p. 145 ch. 8).

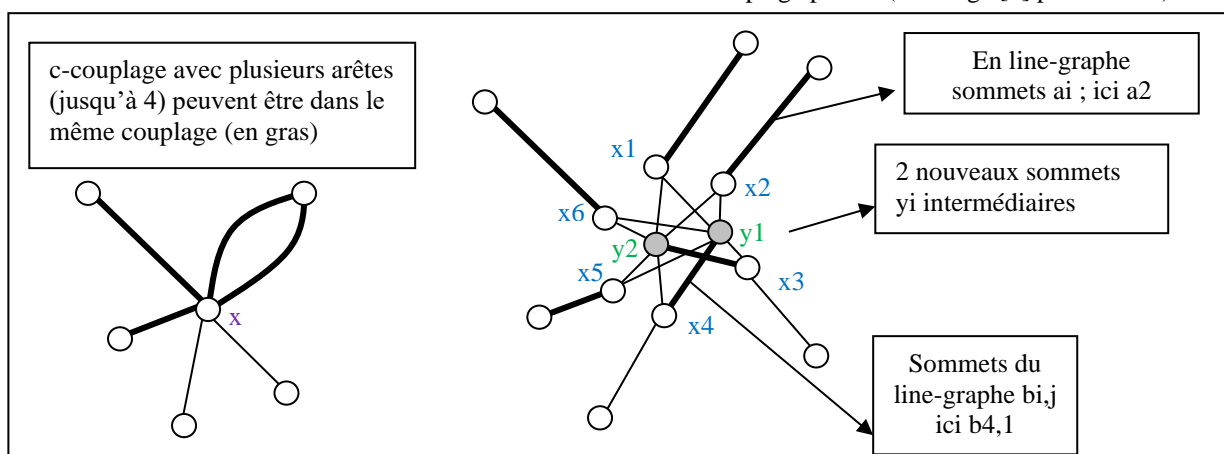


Figure 2. Exemple de transformation d'un 4-couplage maximum en un couplage maximum

Parallèle pour un c -stable

En faisant le parallèle, dans le c -couplage, les arêtes concernées a_i forment une clique que concrétise le sommet x et sa constante c . Dans ce parallèle, nous travaillons sur une clique de sommets, c'est à dire une disjonction entre eux deux à deux, où nous voulons au plus c sommets de cette clique pouvant être dans le même stable. C'est typiquement le cas quand on veut allouer un créneau horaire à un cours de TP constitué en groupes. Les sommets sont l'agrégat (groupe d'élèves-prof-matière) avec une même salle d'accueil –TP) qui a une capacité de c groupes. Nous pouvons alors avoir au même horaire au maximum c cours évidemment. L'intérêt est d'ajouter les autres contraintes disjonctives sur les cours de même prof, élèves, etc..

En prenant le line-graphe du c -couplage, le principe est :

- les p sommets en question sont nommés a_1 à a_p correspondants aux arêtes éclatées ;
- créons $(p-c)$ cliques sur p sommets intermédiaires b_{ij} tel que qu'il y ait une clique par j . Pour un même j , i de 1 à $(p-c)$, les arêtes relient les b_{ij} , i étant de 1 à p . Dans le diagramme précédent, elle représentent les arêtes (x_i, y_j) autour d'un y_j ;
- créons p cliques, chacune reliant un sommet a_k et un sommet b_{ij} des cliques précédemment créées soit sur les sommets $(a_k, b_{k1}, \dots, b_{k(p-c)})$.

Le parallèle sera complet avec la transformation en couplage parfait de la figure 2 si notre c -stable rencontre toutes ses cliques créées. Ce cas est possible lors de la transformation en problème SAT comme il sera vu à la section suivante. En forme CNF, on ajouterait une clause d'exclusivité où chaque sommet des cliques ajoutées est représenté par un littéral et la clause pose un « OU » entre chacun de ces littéraux de la clause concernée. Par exemple, pour les p dernières cliques, on ajouterait $(a_k \vee b_{k1} \vee \dots \vee b_{k(p-c)})$ en plus de ET $(a_k \vee \text{non } b_{ki})$ $i=1$ à $p-c$ ET la clause d'exclusivité suivante : $(\text{non } b_{ki} \vee \text{non } b_{kj})$ pour tout $i=1$ à $p-c$ et $j=1$ à $p-c$.

Reprenons l'exemple des cours 1 à 5 qui sont représentés par 5 sommets a_1 à a_5 . Nous autorisons un stable maximum de 3 sommets parmi les 5. Nous avons $(5-3)$ soit 2 cliques de 5 sommets b_{ij} représentées par des arêtes noires et 5 cliques reliant des triplets (a_k, b_{i1}, b_{i2}) , i allant de 1 à 5 représentées par des arêtes de couleur. Illustrons le résultat sur le schéma 3 ci-dessous :

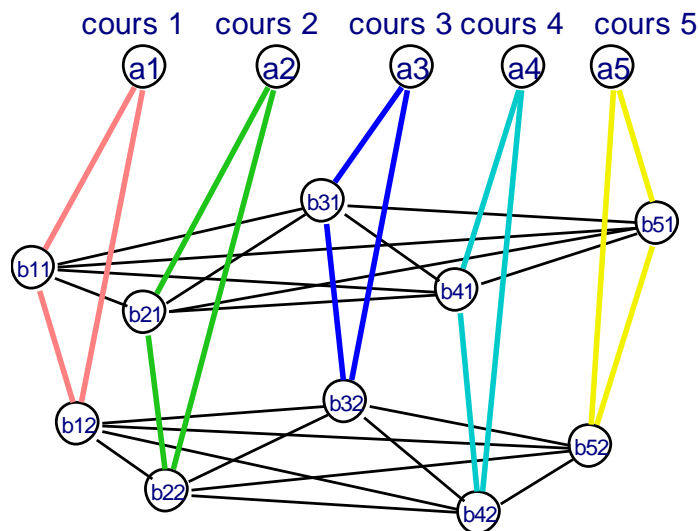


Figure 3. Exemple de transformation d'un 3-stable de sommets en stable

Ainsi en revenant à notre exemple de groupe de TP ci-dessus, et en revenant au graphe G_e , les sommets de chaque instance G_i sont l'association (cours de groupe, créneau horaire) et les arêtes les contraintes de disjonction ou d'ordre. A chaque instance G_i , on pose sur ses sommets représentant tous les groupes de TP physique, une contrainte de cumul comme vu précédemment. Comme vu en [7], le stable à chercher dans le graphe G^e correspond à la bonne affectation des couleurs en chaque sommet permettant de limiter à un même créneau horaire le nombre de groupes.

Limite d'affectation de ressources a minima

Si on poursuit la méthode pour la recherche de nombre chromatique et stable maximum, rappelons-nous [7], [8] que pour résoudre ce problème on passe itérativement sur une transformation de G en G_e puis en un problème SAT. Cela les transforme en variables logiques x_i et pose des littéraux sur ces variables ainsi que des clauses sur ces littéraux.

En reprenant le même principe que poser une limite maximale sur un ensemble de sommets, la limite minimale se traduit par une limite maximale sur les négations des variables x_i représentant les sommets voulues. Ainsi si nous voulons mettre une limite minima que parmi les p sommets x_i , qu'en chaque couleur, nous voulons au moins k sommets dans cette couleur, on va poser une contrainte a minima.

En développant sur G_e dans notre méthode de résolution, on peut les poser sur les couleurs qui nous intéressent. Si les couleurs représentent des créneaux horaires, on peut choisir de les mettre sur les instances G_i correspondant aux créneaux voulues. Par exemple, on peut demander d'avoir a minima deux sommets à toute heure.

Comment procède-t-on ?

On crée des sommets $\text{non_}x_i$, autant que les sommets x_i concernés, soit p sommets. On pose une contrainte de cumul maximale sur les $\text{non_}x_i$. C'est-à-dire, on veut a minima k sommets, on pose alors une contrainte cumulative au maximum $p-k$ sur les $\text{non_}x_i$ sommets :

- créons $(p-k)$ cliques sur p sommets intermédiaires z_{ij} , i étant de 1 à p , j de 1 à $(p-c)$,
- créons p cliques, chacune reliant un sommet $\text{non_}x_k$ et un sommet des cliques précédemment créées soit sur les sommets $(\text{non_}x_k, z_{k1}, \dots, z_{k(p-c)})$

On ajouterait en plus dans un problème SAT transformé les clauses d'exclusivité traduisant que le stable rencontre ces cliques.

Un des exemples d'utilisation est la contrainte de fréquence où on utiliserait ces limites a minima et a maxima. Si on veut avoir une ressource utilisée exactement une fois par semaine, on pose une contrainte au maximum de 1 et de minimum de 1 d'une façon cumulative sur tous les x_i (et $\text{non } x_i$), i représentant les instances des créneaux horaires une semaine.

e. Une résolution unique : passage du graphe étendu posé vers un problème SAT

Le problème est le maintien entre x_i et $\text{non } x_i$, l'arête traduisant un simple « ou » logique. Ici il faut un ou exclusif. Du coup, la modélisation en graphe étendu permet de poser ou de concrétiser cette contrainte mais pas de la résoudre. Pour cela, il faut passer à une modélisation en problème SAT.

Lors de la transformation de G^e en problème SAT [8],[12] au lieu de créer une variable $\text{non_}x_i$ correspondant au sommet $\text{non_}x_i$, on prend la négation des variables x_i , soit $\neg x_i$. Les clauses induites par les contraintes maximum sur les sommets $\text{non_}x_i$ et z_{ij} seront posées sur les variables $(\neg x_i)$ et z_{ij} .

Chaque copie de graphe indique la présence et des contraintes sur un même horaire ou ressource. Le fait d'avoir l'entité dupliquée par horaire donne la possibilité de mettre des contraintes entre ces entités et leurs horaires possibles. Cela permet aussi d'augmenter ces possibilités en travaillant sur les absences d'une entité à un horaire précis. C'est ce qu'autorise aussi la transformation vers un problème SAT en travaillant sur les variables mais aussi sur leurs négations, une variable représentant le couple (entité/tâche, horaire). La négation de la variable représente que la tâche n'est pas présente à cet horaire.

4. Conclusion

En conclusion, nous avons donné un kit de contraintes.

Celui-ci n'est pas détaché d'une résolution réelle car on obtient une résolution complète et exacte de problèmes en recherche opérationnelle par une modélisation sur une recherche de nombre chromatique en se basant sur les transformations vues dans les papiers précédents qui change le nombre chromatique en nombre de stabilité (et même vice-versa) puis en indice chromatique et aussi dans sa résolution exacte et polynomiale de l'indice chromatique.

Cela va permettre de créer une boîte à outil pour instancier et décrire plus précisément chaque problème exactement.

Le futur serait de réfléchir sur un relâchement de contraintes quand le nombre de contraintes ne donne aucune solution. Ce point est traité dans un algorithme de résolution de l'indice chromatique en cours de rédaction (avec sa partie théorique). Cet algorithme dans sa résolution va chercher les sous-graphes χ' -critiques, c'est-à-dire les sous-graphes tel que l'enlèvement de chaque arête de ce sous-graphe fait baisser son indice chromatique. En marquant les arêtes générées par les contraintes d'ordonnancement, cela permet de trouver les parties critiques et montrer toutes les arêtes contraignantes. En fonction de leur priorité, on peut trouver la contrainte à relâcher totalement ou en partiel.

5. Quelques références intéressantes à consulter

- [1] **C. Berge**, *Graphes*, Bordas, Paris, 1983.
- [2] **C. Berge**, *Hypergraphes, Combinatoire des ensembles finis*, Bordas, Paris, 1987.
- [3] **X. Cousin**, *Application de la Programmation en Logique avec Contraintes au problème d'emploi du temps*, rapport de Doctorat de l'Université de Rennes, 1993 : <https://hal.archives-ouvertes.fr/tel-02270451>
- [4] **X. Cousin**, *Emploi du temps : problème mathématique ou problème pour la programmation en logique avec contraintes ?*, Note Technique CNET, NT/LAA/SLC/313, juillet 1989.
- [5] **X. Cousin**, *Emploi du temps : problème mathématique ou problème pour la programmation en logique avec contraintes ?*, Publication Interne N°494 IRISA, septembre 1989 : <https://hal.archives-ouvertes.fr/hal-02270455>
- [6] **X. Cousin**, *Still other limits of chromatic number*. 2019 : <https://hal.archives-ouvertes.fr/hal-02314827>
- [7] **X. Cousin**, *Graph Cartesian sum: pivot between chromatic number and stability number*. 2019 : <https://hal.archives-ouvertes.fr/hal-02334849>
- [8] **X. Cousin**, *To transform the chromatic number search to the chromatic index one in cubic graph*. 2019 : <https://hal.archives-ouvertes.fr/hal-02334852>
- [9] **M. R. Garey, D. S. Johnson, L. Stockmeyer**, *Some simplified polynomial complete problems*, Proc. 6th Annual ACM Symposium on Theory of computing, pp 47-63, 1974.
- [10] **M. R. Garey, D. S. Johnson**, *Computers and Intractability, A guide to the Theory of NP-Completeness*, W. H. Freeman and Company, San Franscisco, 1979.
- [11] **M. Gondran, M. Minoux**, *Graphes et algorithmes*, Eyrolles, Paris, 1985.
- [12] **Ian Holyer**, *The NP-Completeness of edge-colouring*, SIAM J. Comput, Vol 10, N° 4, November 1981 pp 718-720.
- [13] **R. C. Read, R.E. Trajan**, *Bounds backtrack algorithms for listing cycles paths spanning trees*, NETWORKS, N° 5, 1975, pp 237-252
- [14] **Stephan Grünewald, Eckhard Steffen**, *Cyclically 5-edge connected non-bicritical critical snarks*, DIMACS Technical report 97-76, November 1997.
- [15] **Gunnar Brinkmann, Eckhard Steffen**, *Chromatic Index Critical Graphs of Orders 11 and 12*, DIMACS Technical report 97-59, September 1997.
- [16] **Stephan Grünewald, Eckhard Steffen**, *Chromatic-index critical graphs of even order*, DIMACS Technical report 97-39, September 1997.
- [17] **Jean-Claude Fournier**, *Colorations des arêtes d'un graphe*, Cahiers du centre d'études de recherche opérationnelle, Vol 15 N° 3, 1973 pp 311-314
- [18] **Jean-Claude Fournier**, *Sur les stables d'un graphe*, Cahiers du centre d'études de recherche opérationnelle, Vol 17, 1975 pp 193-195.
- [19] **Jean-Claude Fournier**, *Méthode et théorème général de coloration des arêtes d'un multigraphe*, Journal de mathématiques pures et appliquées, Vol 9, N° 56, 1977 pp 437-453.

- [20] **M.K. Goldberg**, *Construction of class 2 graphs with maximum vertex degree 3*, N° B 31, 1981, pp 282-291
- [21] **A.G. Chetwynd, R.J. Wilson**, *The rise and fall of the critical graph conjecture*, Vol 7, N° 2, 1983, pp 153-157
- [22] **Julius Peter Christian Petersen**, « Die Theorie der regulären Graphs », [*Acta Mathematica*](#), n° 15, 1891, p. 193–220