

MARC JIANG

**Méthodes heuristiques pour le problème du
T-coloriage avec ensembles**

RAIRO. Recherche opérationnelle, tome 32, n° 1 (1998), p. 75-99

http://www.numdam.org/item?id=RO_1998__32_1_75_0

© AFCET, 1998, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
<http://www.numdam.org/>

MÉTHODES HEURISTIQUES POUR LE PROBLÈME DU T -COLORIAGE AVEC ENSEMBLES (*)

par Marc JIANG ⁽¹⁾

Communiqué par Catherine ROUCAIROL

Résumé. – *Nous nous intéressons aux heuristiques pour le problème de T -coloriage avec ensembles, afin de résoudre un problème d'allocation de fréquences de taille importante dans l'aviation civile. Le but est de minimiser la largeur du spectre de fréquence utilisée. Deux méthodes heuristiques sont étudiées et expérimentées : l'Ordre dynamique et la recherche Tabou. Une hybridation de ces deux méthodes est également expérimentée. Les résultats montrent une convergence rapide de l'Ordre dynamique, alors que la recherche Tabou s'avère être un outil redoutable d'amélioration de solutions.* © Elsevier, Paris

Mots clés : Optimisation combinatoire, coloriage de graphe, allocation des fréquences, méthodes heuristiques, recherche Tabou.

Abstract. – *We are interested in heuristics for the set T -coloring problem, in order to solve a huge frequency allocation problem in civil aviation. The goal is to minimize the frequency bandwidth. Two heuristic methods are studied and experimented: dynamic Order and Tabu search. An hybridisation of these methods is used as well. The results show a swift convergence of the dynamic Order, and the Tabu search is a good tool to improve solutions.* © Elsevier, Paris

Keywords: Combinatorial optimisation, graph coloring, frequency allocation, heuristic methods, tabu search.

1. INTRODUCTION

Le problème de coloriage classique d'un graphe est un problème combinatoire NP -complet. Il peut être généralisé en T -coloriage avec ensembles et adapté à une large variété d'applications. Un exemple est le problème d'allocation de fréquences dans l'aviation civile. Il consiste à allouer des fréquences à des services appartenant à des zones d'allocation, de telle façon que certaines contraintes soient respectées et que le nombre de

(*) Reçu en juillet 1995.

(1) Laboratoire PRiSM, Université de Versailles et Saint-Quentin, 45 avenue des États-Unis, 78035 Versailles, France, E-mail: marc.jiang@prism.uvsq.fr

Ce travail est financé par le Service Technique de Navigation Aérienne de la Direction Générale de l'Aviation Civile.

fréquences utilisées soit minimal. La formalisation du problème d'allocation de fréquences comme un problème de coloriage généralisé a été proposée par K. Hale [2]. Nous décrivons d'abord le problème du T -coloriage avec ensembles, ainsi que la modélisation du problème d'allocation de fréquences dans l'aviation civile. Ensuite, nous proposerons une heuristique spécifique appelée l'Ordre dynamique et une méthode Tabou fondée sur les propositions de Glover [7, 9] dans les paragraphes 4 et 5. Enfin, les expérimentations, les résultats et les conclusions sont abordées dans les paragraphes 6 et 7.

2. PROBLÈME DE T -COLORIAGE AVEC ENSEMBLES

Soit un graphe $G = (X, E)$ où $X = \{x_1, x_2, \dots, x_n\}$. Chaque nœud $x_i \in X$ reçoit un nombre fixé $d_i \in D = \{d_1, d_2, \dots, d_n\}$ de couleurs. Les couleurs sont représentées par des nombres entiers. Parmi l'ensemble de couleurs à allouer, certaines ne peuvent être allouées à un nœud x_i , ces couleurs interdites appartiennent pour chaque nœud x_i à un ensemble $B_i \subset B = \{B_1, B_2, \dots, B_n\}$.

Une arête entre x_i et x_j est associée à la contrainte de séparation, notée par T_{ij} , un ensemble d'entiers non-négatifs caractérisant les différences interdites entre les couleurs allouées aux nœud x_i et x_j . Par exemple, si $T_{ij} = \{2, 5, 7\}$, la différence entre les couleurs allouées aux nœuds x_i et x_j ne doit pas être 2, 5 et 7. La contrainte de séparation appliquée au même nœud x_i est notée par T_{ii} caractérisant les différences interdites entre les couleurs allouées au même nœud x_i . Soit $c(x_i) = \{c_1(x_i), \dots, c_p(x_i), \dots, c_{d_i}(x_i)\}$ un ensemble de couleurs allouées à un nœud x_i . Le problème de T -coloriage avec intervalles est défini ci-dessous.

DÉFINITION 1 : Un T -coloriage avec ensembles d'un graphe G , noté par $N = (X, E, T, D, B)$, consiste à colorier les nœuds de X avec des couleurs prises dans l'intervalle $\{1, 2, \dots, k\}$, de telle sorte que

k soit minimal

$$|c(x_i)| = d_i \quad \forall x_i, c(x_i) = \{c_1(x_i), \dots, c_p(x_i), \dots, c_{d_i}(x_i)\} \quad (2.0)$$

$$1 \leq c_p(x_i) \leq k \quad \text{pour } p = \overline{1, d_i} \quad (2.1)$$

$$|c_p(x_i) - c_q(x_i)| \notin T_{ii} \quad \forall c_p(x_i), c_q(x_i) \in c(x_i) \quad \text{et } p \neq q, \quad (2.2)$$

$$(x_i, x_j) \in E \Rightarrow |c_p(x_i) - c_q(x_j)| \notin T_{ij} \quad \forall c_p(x_i) \in c(x_i), \forall c_q(x_j) \in c(x_j) \quad \text{et } x_i \neq x_j, \quad (2.3)$$

$$c_p(x_i) \notin B_i \quad \forall x_i, \forall c_p(x_i) \in c(x_i). \quad (2.4)$$

Lorsque les contraintes de séparation contiennent des entiers consécutifs, de la forme $T_{ij} = \{0, 1, 2, \dots, t_{ij} - 1\}$, la contrainte de séparation T_{ij} peut être notée par la variable t_{ij} régissant l'écart entre les couleurs. Les contraintes (2.2) et (2.3) deviennent alors

$$|c_p(x_i) - c_q(x_i)| \geq t_{ii} \quad \forall c_p(x_i), c_q(x_i) \in c(x_i) \quad \text{et } p \neq q \quad (2.2')$$

$$(x_i, x_j) \in E \Rightarrow |c_p(x_i) - c_q(x_j)| \geq t_{ij} \quad \forall c_p(x_i) \in c(x_i), \forall c_q(x_j) \in c(x_j) \quad \text{et } x_i \neq x_j \quad (2.3')$$

Prenons un exemple de T -coloriage avec ensembles où les contraintes de séparation sont des formes (2.2') et (2.3'). Soit le graphe de la figure 1, avec $T = \{t_{ij} | t_{ij} = 2, \text{ si } i = j; t_{ij} \in \{1, 2\}, \text{ si } i \neq j\}$, $D = \{3, 1, 4, 2\}$. Les contraintes de couleurs interdites B ne sont pas envisagées. Une solution du problème $N = (X, E, T, D)$ peut être obtenue avec $k = 13$. En respectant les contraintes de séparation et les demandes, les couleurs allouées aux nœuds sont $c(x_1) = \{2, 4, 6\}$, $c(x_2) = \{1\}$, $c(x_3) = \{7, 9, 11, 13\}$, $c(x_4) = \{3, 5\}$.

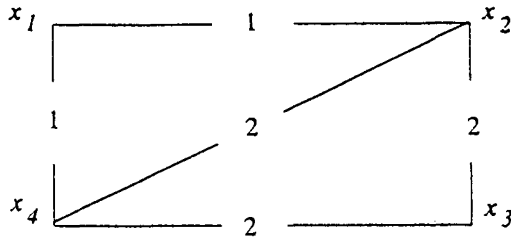


Figure 1. – Graphe de 4 nœuds avec les arêtes associées de contraintes de séparation.

Jusqu'à présent, les méthodes heuristiques connues comme la recherche Tabou et le Recuit simulé ont été utilisées par D. Costa [12] pour résoudre des problèmes de T -coloriage sans ensembles. Une méthode exacte fondée sur l'énumération implicite est comparée par l'auteur à d'autres heuristiques, qui montre qu'elle est capable de traiter des problèmes jusqu'à 36 nœuds, alors que le Recuit simulé ne peut traiter les problèmes que jusqu'à 200 nœuds et la recherche Tabou jusqu'à 1000 nœuds. Ces résultats montrent le handicap des méthodes exactes face aux problèmes de grande taille.

3. PROBLÈME D'ALLOCATION DE FRÉQUENCES DANS L'AVIATION CIVILE

3.1. Problématique

Le problème d'allocation de fréquences consiste ici à allouer des ressources (fréquences) aux émetteurs de façons à optimiser leur utilisation. Le terme « optimisation » a plusieurs sens. Le sens strict est la satisfaction de l'ensemble des émetteurs avec le minimum de ressources. Le sens relâché est l'allocation d'un nombre fixe de ressources à l'ensemble des émetteurs en violant le minimum de contraintes.

L'étude se limite à la bande COM2 (118-137 Mhz) destinée aux communications entre les aéronefs et les contrôleurs aériens. Les services sont classés en 13 types de services selon le groupe de protection, la forme géographique (circulaire ou polygonale) et la Couverture Opérationnelle Spécifiée (COS) (voir tableau I).

TABLEAU I
Types de service de COM2.

Numéro du type	Groupe protection	Symbole	COS (NM/FL×100)	Forme géographique
1	B	AFIS, TWR/L	16/30	Point centre
2	1	TWR	25/40	
3	2	APP/L	25/100	
4	3	APP/I	40/150	
5	A	ATIS	60/200	
6	4	APP/H	50/250	
7	5	VOLMET	260/0	
8	C	ACC/LL	0/150	Polygone
9	4	ACC/L	0/250	
10	D	ACC/I	0/350	
11	5	ACC/U	0/450	
12	-	AS	05/00	Point centre
13	-	OPC	25/00	

La bande COM2 dispose actuellement de 760 fréquences utilisables avec un pas de 25 KHz. Ce nombre de fréquences ne répond plus aux besoins actuels, compte tenu de l'augmentation du trafic aérien. On ne peut plus

allouer de fréquence à un nouveau service sans créer des interférences avec les autres. D'après l'analyse sur la planification actuelle, on a constaté 9400 violations de contraintes d'interférence, qui mettent en cause le fonctionnement correct de certains services.

La cause d'un tel problème est de deux sortes : une mauvaise allocation des fréquences ou une insuffisance du nombre de fréquences. Pour pallier la première cause, on peut réallouer les fréquences aux services de façon à minimiser le nombre de fréquences ou minimiser la violation de contraintes d'interférence. Pour éliminer la deuxième cause, il faudrait tout simplement augmenter le nombre de fréquences.

3.2. Modélisation

Donnons d'abord quelques concepts de base liés au système de l'aviation civile. Une *zone d'allocation* est une région où les fréquences sont opérationnelles. Elle a une forme géographique : *point centre* ou *polygonale*.

– Lorsque la zone est un point centre, une fréquence destinée à un service est opérationnelle dans un espace cylindrique du volume défini par la COS de son type, l'émetteur se situant au point centre.

– Lorsque la zone est polygonale, une fréquence destinée à un service est opérationnelle dans un espace polyédrique (la zone est la projection de ce polyèdre) dont la surface est définie par OACI (Organisation de l'Aviation Civile Internationale), l'altitude est définie par la COS de son type, l'émetteur peut se situer dans cette zone ou en dehors de celle-ci.

Un *service* est défini par son type et la zone d'allocation à laquelle il est affecté. Il ne sera opérationnel que si une fréquence lui est allouée. Une zone peut demander plusieurs services appartenant au même type ou aux plusieurs types. Prenons un exemple de la zone *Paris/Ch. de Gaulle* 4901N 0233E, le tableau II montre les services classés par leur type.

Dans le système de communication COM2, les contraintes à respecter sont les interférences entre les services, les nombres de services à assurer sur chacune de zones et des fréquences interdites à allouer à certains services.

Les contraintes d'interférence (appelées contraintes de séparation) sont interprétées de la distance entre les services. La distance entre les services est mesurée différemment selon que les zones sont point centre ou polygonales. La distance à partir d'un point centre est mesurée à partir du point lui-même ; la distance à partir d'un polygone est mesurée à partir de son extrémité. On

TABLEAU II
Les services de COM2 dans la zone Paris/Ch. de Gaulle.

Groupe protection	Symbole	Nombre de services
B	AFIS, TWR/L	0
1	TWR	5
2	APP/L	2
3	APP/I	3
A	ATIS	1
4	APP/H	4
5	VOLMET	0
C	ACC/LL	0
4	ACC/L	0
D	ACC/I	0
5	ACC/U	0
–	AS	10
–	OPC	20

trouve deux types de contraintes représentées par la séparation entre les fréquences allouées aux services :

– *contrainte de fréquences identiques* : deux services à moins d'une distance critique l'un de l'autre ne peuvent pas utiliser la même fréquence ; la contrainte de séparation vaut 1.

– *contrainte de fréquences adjacentes* : deux services à moins d'une distance critique l'un de l'autre ne peuvent pas utiliser la même fréquence, ni les fréquences à un pas du découpage ; la contrainte de séparation vaut 2.

Les contraintes d'interférence entre les services de types p et q sur les zones d'allocation i et j respectives peuvent être formalisées par (3.1) et illustrées dans la figure 2.

$$t_{ip,jq} = \begin{cases} 0 & \text{si } dI_{pq} \leq d_{ij}, \\ 1 & \text{si } dA_{pq} \leq d_{ij} < dI_{pq}, \\ 2 & \text{si } d_{ij} < dA_{pq}, \end{cases} \quad (3.1)$$

où d_{ij} est la distance entre les zones d'allocation i et j ;

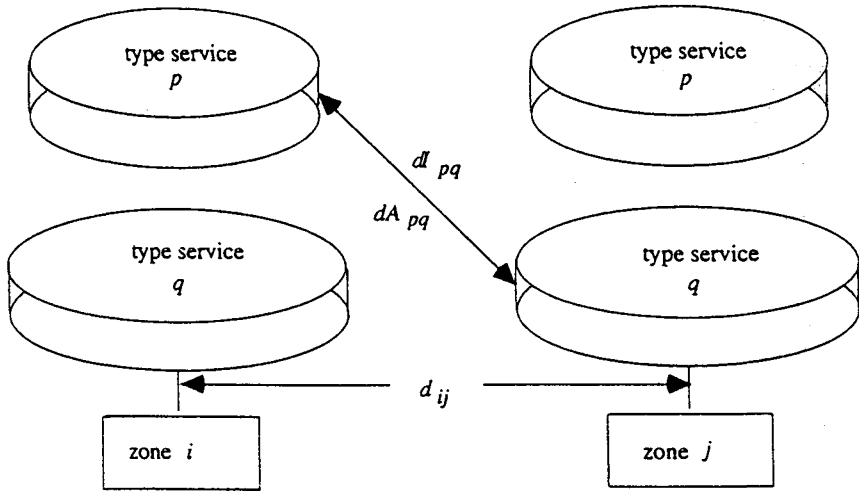


Figure 2. – Critères de contraintes de fréquences identiques et adjacentes.

dI_{pq} est la distance minimale acceptant des fréquences identiques entre les services de types p et q ;

dA_{pq} est la distance minimale acceptant des fréquences adjacentes entre les services de types p et q .

Définissons un graphe $G = (X, E)$, où $x_{ip} \in X$ est un nœud représentant le type de services p appartenant à la zone d'allocation i , d_{ip} est le nombre de services demandés du type p sur la zone i . Nous avons l'ensemble des nœuds $X = \bigcup_{p=1}^s X_p$ où $X_p = \{x_{1p}, x_{2p}, \dots, x_{n_p p}\}$ est un ensemble des nœuds sur la couche p , l'ensemble des arêtes $E = \bigcup_{p=1}^s \bigcup_{q=p}^s E_{pq}$ où E_{pq} est un ensemble des arêtes entre les nœuds sur les couches p et q respectives, et les contraintes de séparation $T = \{t_{ip, jq}\}$ associées avec les arêtes, les contraintes de demandes $D = \{d_{ip}\}$ et les contraintes de fréquences interdites $B = \{B_{ip}\}$. Ce graphe est illustré par la figure 3. C'est un graphe à 13 couches dont chacune est associée à un type de services. Soit $f(x_{ip}) = \{f_1(x_{ip}), \dots, f_r(x_{ip}), \dots, f_{d_{ip}}(x_{ip})\}$ un ensemble des fréquences allouées aux services du type p sur la zone d'allocation i , où $p \in [1, s]$, $i \in [1, n_p]$ et $r \in [1, d_{ip}]$, $f_r(x_{ip}) \in \{1, 2, \dots, k\}$. Nous avons le problème d'allocation de fréquences défini dans la suite.

DÉFINITION 2 : Le problème d'allocation de fréquences $N = (X, E, T, D, B)$ consiste à allouer des fréquences d'un spectre de largeur minimale k

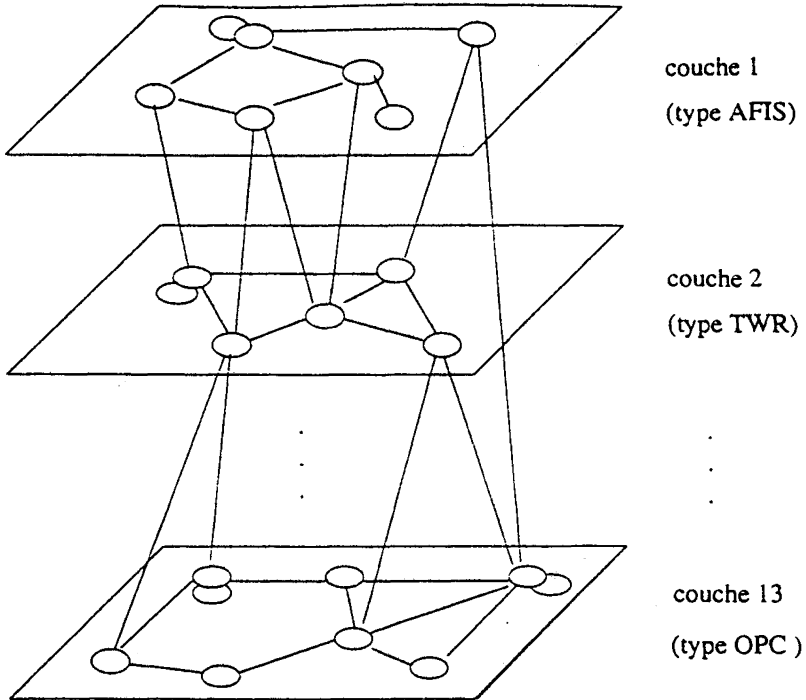


Figure 3. - Graphe à multicouches représentant le système de COM2.

aux services, de telle sorte que

$$|f(x_{ip})| = d_{ip} \quad \forall x_{ip}, f(x_{ip}) = \{f_1(x_{ip}), \dots, f_{d_{ip}}(x_{ip})\} \quad (3.2)$$

$$\begin{aligned} |f_r(x_{ip}) - f_s(x_{ip})| \\ \geq t_{ip,ip} \end{aligned} \quad \begin{aligned} \forall f_r(x_{ip}) \in f(x_{ip}), f_s(x_{ip}) \in f(x_{ip}) \\ \text{et } r \neq s, \end{aligned} \quad (3.3)$$

$$\begin{aligned} (x_{ip}, x_{jq}) \in E \Rightarrow \\ |f_r(x_{ip}) - f_s(x_{jq})| \\ \geq t_{ip,jq} \end{aligned} \quad \begin{aligned} \forall f_r(x_{ip}) \in f(x_{ip}), f_s(x_{jq}) \in f(x_{jq}) \\ \text{et } x_{ip} \neq x_{jq}, \end{aligned} \quad (3.4)$$

$$f_r(x_{ip}) \notin B_{ip} \quad \forall x_{ip}, f_r(x_{ip}) \in f(x_{ip}). \quad (3.5)$$

DÉFINITION 2bis : Le problème d'allocation de fréquences $N = (X, E, T, D, B)$ consiste à allouer des fréquences d'un spectre de largeur fixe k aux services, de telle sorte que les contraintes exprimées par (3.2) et (3.5) soient satisfaites et que le nombre de violations des contraintes (3.3) et (3.4) soit minimal.

Le problème d'allocation de fréquences est ainsi modélisé par un T -coloriage avec ensembles. Pour le cas pratique, ce graphe possède 5468 nœuds répartis sur $s = 13$ couches, 1 873 064 arêtes et 9 131 services (demandes) correspondant au système européen dans le tableau COM2. La densité du graphe est de 0,125.

4. MÉTHODE DE L'ORDRE DYNAMIQUE [1, 5]

Comme un nœud x_i du graphe G doit recevoir un nombre d_i de couleurs, il doit effectuer d_i demandes consécutives pour satisfaire cette contrainte. Ainsi, une demande peut être caractérisée par son appartenance au nœud et son numéro. Toutes les demandes de couleurs sont rangées dans un certain ordre et ensuite satisfaites dans cet ordre avec des couleurs numérotées de 1 à k . Chaque demande est coloriée avec la plus petite couleur possible. Sachant que le nombre de couleurs utilisées dépend de l'ordre dans lequel les demandes de couleurs sont considérées, une certaine permutation de l'ordre des demandes de couleurs doit aboutir à un coloriage optimal.

La méthode de l'Ordre dynamique est une procédure itérative qui tente d'approcher, en partant d'un nombre illimité de couleurs, la borne inférieure calculée préalablement. L'idée principale est d'apprécier la difficulté à satisfaire chaque demande avec une couleur au-dessous de la borne inférieure et d'allouer en priorité les couleurs aux demandes présentant les plus fortes difficultés. Lorsqu'une demande a reçu une couleur plus grande que la borne inférieure, sa valeur de difficulté de satisfaction sera augmentée et ainsi que sa priorité d'allocation.

La borne inférieure joue un rôle d'évaluation de la difficulté et une condition d'arrêt. Diverses façons de calculer cette borne inférieure ont été décrites dans [3]. La méthode la plus précise est de parcourir toutes les cliques et de chercher la clique qui a besoin le plus de couleurs. Pour simplifier ce calcul, on a opté pour la recherche de la clique ayant le nœud avec le plus grand nombre de demandes de couleurs $x = (x_i | \max_i (d_i))$, et calculer le nombre de couleurs nécessaires pour colorier cette clique. Ce nombre sera considéré comme la borne inférieure de coloriage du graphe.

Le nombre total de demandes de couleurs est donné par $M = \sum_{i=1}^n d_i$, où d_i est le nombre de demandes de couleurs au nœud x_i . Par la suite, nous notons, pour $k = 1, \dots, M$ et $i = 1, \dots, n$,

$color(k)$	la couleur allouée à la demande k ; $color(k) = 0$ indique l'absence de couleur de la demande k ;
$dem(k)$	le k -ième élément de la permutation ;
$nœud(k)$	le nœud auquel la demande k appartient ;
$diff(k)$	la difficulté de satisfaction de la demande k ;
$block(i)$	le nombre de couleurs interdites au nœud x_i .

4.1. Initialisation et conditions d'arrêt

Les demandes de couleurs sont initialement rangées suivant l'un des trois ordres : alphanumérique, degré d'espace ou aléatoire.

– L'ordre alphanumérique est le plus naturel, il s'agit de ranger les demandes de couleurs suivant la numérotation de leurs nœuds. Les numéros $1, 2, \dots, d_1$ représentent les demandes appartenant au nœud x_1 , et les numéros $d_1 + 1, d_1 + 2, \dots, d_1 + d_2$ représentent les demandes appartenant au nœud x_2 , etc.

– La notion de degré d'espace vient de celle de degré d'un nœud du graphe qui représente le nombre de voisins d'un nœud : le degré d'espace est la somme des t_{ij} , si x_j est le voisin de x_i . Nous avons le degré d'espace à un nœud x_i

$$\deg(x_i) = \sum_{(x_i, x_j) \in E} t_{ij}.$$

Les demandes sont rangées dans l'ordre décroissant du degré d'espace de leurs nœuds.

Les difficultés de satisfaction associées aux demandes de couleurs sont initialisées, de telle sorte que

$$diff(dem(k)) > diff(dem(k + 1)). \quad (4.1)$$

Deux conditions d'arrêt sont utilisées. Si le nombre de couleurs utilisées dans la phase d'allocation ne dépasse pas la borne inférieure ou si les $nmax$ itérations se sont écoulées sans améliorer la meilleure allocation trouvée, la procédure est arrêtée.

4.2. Procédure de l'Ordre dynamique

Chaque itération de l'Ordre dynamique consiste en trois phases successives décrites dans le tableau III. La première phase range les demandes de couleurs

TABLEAU III

Procédure générale de l'Ordre dynamique.

Initialisation

BI: {borne inférieure}
nmax; {nombre maximal d'itérations entre deux améliorations}
niter = 0; {compteur d'itération}
bniter = 0; {itération où la meilleure allocation est sauvegardée}
best = ∞; {nombre minimal de couleurs utilisées}
 donner un ordre initial des demandes de couleurs $dem(k)$;
 attribuer les valeurs décroissantes à la fonction $diff(k)$ suivant l'ordre de $dem(k)$;

Tant que $best > BI$ et $(niter - bniter) < nmax$ faire

$niter = niter + 1$;

Phase Permutation

Ranger les demandes dans la fonction $dem(k)$, de telle sorte que (4.1) soit satisfaite;

Phase Allocation {appliquer une stratégie d'allocation, donner $ncolor$ }

Phase Evaluation

pour tout k

si $color(k) > BI$ alors $diff(k) = diff(k) + r$; { r est une valeur aléatoire dans un intervalle $(1, 5)$ }

si $ncolor < best$ alors

sauvegarder la meilleure allocation;

$bniter = niter$;

$best = ncolor$;

fin si;

fin faire.

dans l'ordre décroissant de leur difficulté (phase permutation) suivant l'expression (4.1). La deuxième met en œuvre une stratégie d'allocation pour allouer les couleurs aux demandes dans cet ordre (phase d'allocation, voir le tableau IV). Les demandes de couleurs sont satisfaites dans l'ordre décrit par la fonction $dem(k)$ par des couleurs à partir de 1, jusqu'à ce que toutes les demandes soient satisfaites; le plus grand numéro de couleur représente le nombre des couleurs utilisées. Enfin, dans la troisième phase, on évalue la difficulté de satisfaction des demandes qui ont reçu des couleurs plus grandes que la borne inférieure (phase d'évaluation). Souvent, l'évaluation de difficulté est empirique, elle consiste à générer une valeur aléatoire uniformément distribuée dans un intervalle prédéterminé, la difficulté d'une demande sera sa propre valeur incrémentée de la valeur ainsi générée.

TABLEAU IV

*Phase d'allocation.***Initialisation**mise à zéro de la fonction $block(i)$ et $color(k)$ pour tout i et k ; $ncolor = 0$; { nombre de couleurs utilisées } $rep = 0$; { nombre de demandes ayant reçu de couleurs }**Allocation**Tant que $rep < M$ faire $ncolor = ncolor + 1$; { ajouter une couleur }mise à jour de $block(i)$ pour tout i , telles que $block(i) = \max(block(i) - 1, 0)$;si pour tout i $ncolor \in B_i$ et $block(i) = 0$ alors $block(i) = 1$; $k = 0$; { nombre de demandes de couleurs }Tant que $k < M$ faire $k = k + 1$; $j = n\text{æud}(dem(k))$;si $color(dem(k)) = 0$ et $block(j) = 0$ alors $color(dem(k)) = ncolor$; $rep = rep + 1$; $block(i) = \max(block(i), t_{ij})$ pour tout i ;

fin si;

fin faire k ;fin faire rep .

La complexité en temps à l'initialisation est $O(M)$. A chaque itération de la boucle, la complexité de la première phase est $O(M^\ell \log M)$, celle de la deuxième phase est $O(m)$ (m est le nombre d'arêtes du graphe) et celle de la troisième phase est $O(M)$. La complexité au pire cas de chaque itération est $O(m + M \log M)$. Comme $M \ll m \ll n^2$, la complexité de chaque itération de l'Ordre Dynamique est alors $O(m)$.

5. RECHERCHE TABOU

La recherche Tabou est une procédure itérative d'exploration d'espace des solutions. Chaque itération consiste à sélectionner le meilleur mouvement parmi un ensemble de candidats, qui mène d'une solution actuelle vers la meilleure solution de son voisinage, provoquant la plus grande diminution

ou la plus petite augmentation de la fonction coût. Le critère de sélection est fondé sur une liste Tabou L_T et une fonction d'aspiration $A(f(s))$. La liste Tabou interdit les mouvements déjà effectués pendant les $|L_T|$ dernières itérations afin d'éviter le cyclage de la procédure. La fonction d'aspiration permet de lever le statut tabou de mouvements si ceux-ci peuvent améliorer la solution actuelle, elle permet de rendre la liste Tabou moins rigide. Cette technique de base a été étudiée dans beaucoup d'articles [4, 6, 7, 9, 10, 12]. Nous ne la présenterons pas dans les détails, sauf quelques particularités.

Soit un problème de T -coloriage avec ensembles $N = (X, E, T, D, B)$. Nous allons utiliser k couleurs pour l'ensemble des nœuds de X . Nous allons d'abord obtenir une allocation de $k + 1$ couleurs avec d'autres outils. La solution initiale de k couleurs sera dérivée de celle de $k + 1$ couleurs en remplaçant la couleur $k + 1$ sur les demandes reçues de cette couleur par des couleurs entre 1 et k . Les couleurs sur les autres demandes restent inchangées. Le processus de recherche Tabou est alors appliqué à l'allocation en minimisant une fonction coût. Lorsqu'une allocation de k couleurs est trouvée, k sera diminué d'une unité, nous appliquerons le même principe pour trouver la solution initiale de la nouvelle valeur de k et la recherche Tabou sera appliquée à nouveau. Le processus est terminé si une allocation de k couleurs n'est pas obtenue. L'avant-dernière valeur de k sera notre estimation minimale du T -coloriage. Une telle approche est adoptée par D. Costa, A. Hertz et M. Klimowicz dans [4, 12, 13].

Dans un problème de T -coloriage avec k couleurs, un ensemble S de solutions admissibles comprend tous les recouvrements $s \in S$ en k sous-ensembles V_1, V_2, \dots, V_k , où V_i est le sous-ensemble des nœuds V_i recevant la couleur i . Il est possible qu'un nœud x_i appartienne aux ensembles $V_{c_p(x_i)}$, $1 \leq p \leq d_i$, s'il a reçu les couleurs $c_1(x_i), \dots$, et $c_{d_i}(x_i)$.

Pour pouvoir déterminer le coût d'une solution, définissons d'abord un indicateur $\delta(c_p(x_i), c_q(x_j))$ qui mesure la violation de contraintes de séparation entre les couleurs $c_p(x_i), c_q(x_j)$ allouées aux sommets x_i et x_j . Par définition :

$$\delta(c_p(x_i), c_q(x_j)) = \begin{cases} t_{ij} - |c_p(x_i) - c_q(x_j)| & \text{si } |c_p(x_i) - c_q(x_j)| < t_{ij}, \\ 0 & \text{si non.} \end{cases} \quad (5.1)$$

Lorsque $\delta(c_p(x_i), c_q(x_j)) = t_{ij} - |c_p(x_i) - c_q(x_j)|$, les couleurs $c_p(x_i)$ et $c_q(x_j)$ allouées respectivement aux nœuds x_i et x_j violent la contrainte de

séparation entre les nœuds. Lorsque $\delta(c_p(x_i), c_q(x_j)) = 0$, la contrainte est respectée. pour le cas où $x_i = x_j$, l'indicateur ci-dessus mesure la violation de contraintes entre les couleurs allouées au même nœud.

La fonction coût dans (5.2) est construite à partir des contraintes de séparation du problème. Une solution de coût positif indique que certaines contraintes sont encore violées, alors qu'un coût nul indique le respect de toutes les contraintes de séparation. Elle est définie comme suit :

$$\begin{aligned}
 f(s) &= f(V_1, V_2, \dots, V_k) \\
 &= C_1 \sum_{\substack{(x_i, x_j) \in E \\ i \neq j}} \sum_{p=1}^{d_i} \sum_{q=1}^{d_j} \delta(c_p(x_i), c_q(x_j)) \\
 &\quad + C_2 \sum_{i=1}^n \sum_{p=1}^{d_i-1} \sum_{q=p+1}^{d_i} \delta(c_p(x_i), c_q(x_i)). \quad (5.2)
 \end{aligned}$$

où C_1 et C_2 sont des constantes et positives. Le premier terme exprime les contraintes de séparation entre les nœuds et le deuxième sur les mêmes nœuds.

5.1. Voisinage $N(s)$ et évaluation du coût

Le voisinage $N(s)$ de la solution s consiste en tous les recouvrements qui peuvent être obtenus à partir de s en changeant une couleur d'un nœud. Deux recouvrements $s = \{V_1, V_2, \dots, V_k\}$ et $s' = \{V'_1, V'_2, \dots, V'_k\}$ sont *voisins* s'il existe deux sous-ensembles V_α, V_β ($\alpha \neq \beta$) et un nœud $x \in V_\alpha, x \notin V_\beta$ tel que $V'_\alpha = V_\alpha \setminus x, V'_\beta = V_\beta \cup x$, et $V'_p = V_p$ ($p \neq \alpha, \beta$). Le nœud x est effacé de V_α et ajouté dans V_β (ce qui signifie que la couleur α est remplacée par la couleur β sur le nœud x).

Définissons ainsi un *mouvement* (x, α, β) , le remplacement d'une couleur α déjà allouée à un nœud x par une autre couleur β non allouée à un tel nœud. Le voisinage $N(s)$ regroupe l'ensemble des solutions qui pourraient être atteintes à partir de la solution actuelle s par les mouvements (x, α, β) . $\Delta f(ss')$ est le coût du mouvement (x, α, β) , et est exprimé par

$$\Delta f(ss') = \Delta f_{\text{sup}}(ss') + \Delta f_{\text{ins}}(ss'), \quad (5.3)$$

où

$$\begin{aligned} \Delta f_{\text{sup}}(ss') = & -C_1 \sum_{(x, x_i) \in E} \sum_{q=1}^{d_i} \delta(\alpha, c_q(x_i)) - C_2 \\ & \times \sum_{\substack{q=1 \\ \alpha \neq c_q(x)}}^{d_x} \delta(\alpha, c_q(x)), \end{aligned} \quad (5.4)$$

$$\begin{aligned} \Delta f_{\text{ins}}(ss') = & C_1 \sum_{(x, x_i) \in E} \sum_{q=1}^{d_i} \delta(\beta, c_q(x_i)) + C_2 \\ & \times \sum_{\substack{q=1 \\ \alpha \neq c_q(x)}}^{d_x} \delta(\beta, c_q(x)). \end{aligned} \quad (5.5)$$

5.2. Listes de candidats

Une liste de candidats est caractérisée par un ensemble de n voisins mouvements (x, α, β) du voisinage $N(s)$ n'appartenant pas à la liste Tabou T ou satisfaisant le critère d'aspiration A , en plus, les couleurs à insérer n'appartiennent pas à l'ensemble b_x (contraintes de couleurs interdites sur le nœud $x, \beta \notin B_x \subset B$). Sachant qu'un voisinage complet dans un problème de grande taille est impossible à évaluer à chaque itération, en raison du temps consommé, la liste de candidats est restreinte à un sous-ensemble du voisinage $N(s)$. Nous avons développé plusieurs types de liste de candidats.

La liste de candidats la plus simple et la plus facile à mettre en œuvre consiste à générer aléatoirement un ensemble de n voisins candidats. Dans cette liste, les éléments constitutifs d'un candidat sont générés de façon aléatoire avec une probabilité uniforme. La liste peut être décrite par

$$\begin{aligned} L_1 = \{ & (x, \alpha, \beta) \mid i = 1, \dots, n \text{ voisins}; x \in [1, n]; \\ & \alpha \in c(x); \beta \notin c(x) \text{ et } B \notin B_x \}; \end{aligned} \quad (5.6)$$

La deuxième liste de candidats consiste en un ensemble de candidats dans lesquels $\beta_1, \dots, \beta_i, \dots, \beta_m$ sont les couleurs remplaçantes pour chaque

couleur allouée. La liste est décrite par

$$L_2 = \{(x, \alpha, \beta_i) | \forall x \in [1, n]; \forall \alpha \in c(x); \beta_i \notin c(x) \\ \text{et } \beta \notin B_x \text{ avec } i = 1, 2, \dots, m\}. \quad (5.7)$$

Le nombre de candidats dans cette liste est $n \text{ voisin} = m \times \sum_{x=1}^n d_x$.

L'avantage de la liste L_2 est de permettre d'évaluer toutes les couleurs à remplacer à chaque itération, avec le risque d'accroître la consommation en temps. Ainsi, il faut limiter le nombre de couleurs remplaçantes m vis-à-vis de chaque couleur à remplacer (souvent $m = 1$).

La troisième liste de candidats consiste en un ensemble de candidats dont les couleurs à remplacer a violent des contraintes, c'est-à-dire $\Delta f_{sup}(ss') < 0$. La liste est décrite par

$$L_3 = \{(x, \alpha, \beta_i) | \forall x \in [1, n]; \forall \alpha \in c(x) \text{ avec } \Delta f_{sup}(ss') < 0; \\ \forall \beta_i \notin c(x) \text{ et } \beta_i \notin B_x \text{ avec } i = 1, \dots, k - d_x\}. \quad (5.8)$$

La liste L_3 est un sous-ensemble de L_2 . Elle offre la possibilité de trouver rapidement des mouvements améliorants, puisque nous savons qu'un tel mouvement est caractérisé par $\Delta f_{sup}(ss') < 0$.

Une liste L_4 ($L_4 \subset L_3$) de taille encore plus réduite peut être utilisée. Elle est obtenue par la réduction de L_3 de la façon suivante. A chaque itération, les candidats en conflit appartenant à un ou plusieurs nœuds dans L_3 sont sélectionnés; à l'itération suivante, les candidats en conflit appartenant à un ou plusieurs nœuds dans le reste de L_3 sont sélectionnés; ainsi de suite. Cette sélection peut être consécutive ou aléatoire.

La sélection du meilleur mouvement parmi une liste de candidats est fondée sur l'évaluation du coût. Lorsqu'un grand nombre de candidats a des coûts égaux, la sélection devient alors difficile. Dans ce cas, nous construisons une deuxième fonction coût $g(s)$ en introduisant un ensemble de poids : chaque poids est associé à une couleur, $\{w_1, \dots, w_k\}$ et $w_1 < \dots < w_k$, et est tel qu'un nœud contribue pour un poids w_i à la fonction coût $g(s)$ s'il a reçu une couleur i . Ainsi, la valeur de la deuxième fonction coût associée au mouvement $\Delta g(x, \alpha, \beta)$ est donnée par

$$\Delta g(x, \alpha, \beta) = w_\beta - w_\alpha \quad (5.9)$$

La procédure de sélection du meilleur mouvement dans une liste de candidats L est décrite dans le tableau V.

TABLEAU V
Procédure de sélection du meilleur mouvement.

```

best1 = ∞;
best2 = ∞;
Tant que  $L$  n'est pas vide faire
  Extraire un candidat  $(x, \alpha, \beta)$  dans  $L$ ;
  Evaluer  $\Delta f$ ;
  Evaluer  $\Delta g$ ;
  si  $\Delta f < best1$  ou ( $\Delta f = best1$  et  $\Delta g < best2$ ) alors
    best1 =  $\Delta f$ ;
    best2 =  $\Delta g$ ;
    bmove =  $(x, \alpha, \beta)$ ;
  fin si;
fin faire.
```

5.3. Mémoire Tabou L_T et fonction d'aspiration $A(f(s))$

Lorsque (x, α, β) est considéré comme un mouvement, la paire (x, α) est l'attribut du mouvement tabou dans la mémoire Tabou L_T . Cela signifie que la couleur α enlevée du nœud x ne sera plus acceptée pour le même nœud pendant les $|L_T|$ prochaines itérations de la procédure Tabou.

Le paramètre crucial de la mémoire Tabou est la durée du régime Tabou pour un attribut. Ce choix délicat est beaucoup discuté dans la littérature. Une durée trop petite conduit au phénomène de cyclage, une durée trop grande empêche la sélection de mouvements d'élite. Un choix possible et robuste, qui a déjà donné des bons résultats sur d'autres applications, consiste à sélectionner aléatoirement la durée dans un intervalle prédéfini. Nous avons opté pour le choix de durées minimales et maximales proportionnelles à la taille du problème.

La fonction d'aspiration $A(f(s))$ enregistre la plus petite valeur atteinte lorsqu'on sélectionne des mouvements de la solution actuelle vers s . Elle est

initialisée à $A(z) = z - 1$, pour toute valeur entière z . Lors d'un mouvement d'une solution s vers sa voisine s' , la mise à jour de la fonction d'aspiration se fait comme suit :

TABLEAU VI
Procédure de mise à jour de la fonction d'aspiration.

```

si  $f(s') \leq A(f(s))$  alors
     $A(f(s)) = f(s') - 1$ ;
sinon
    si  $f(s) \leq A(f(s'))$  alors
         $A(f(s')) = f(s) - 1$ ;
    fin si;
fin si.
```

Comme nous allons appliquer plusieurs fois la recherche Tabou, nous n'avons utilisé que la version simple (mémoire courte). Les utilisations des stratégies d'intensification et diversification n'ont pas été envisagées, car elles sont coûteuses pour obtenir chaque valeur de k .

6. EXPÉRIMENTATIONS ET RÉSULTATS

Toutes les expérimentations ont été effectuées sur une station de travail SUN SPARCclassic, 24MOctets. Les problèmes vont de la taille 100 à 1500 en nombre de nœuds, pour une densité de 0,5.

6.1. Problèmes aléatoires

Les problèmes de coloriage sont générés de façon aléatoire. Une arête est générée dans le graphe, si une valeur aléatoire de distribution uniforme dans un intervalle $[0, 1]$ est inférieure à la densité donnée. La contrainte de séparation associée est donnée par un entier tiré aléatoirement dans l'ensemble $\{1, 2\}$. Le nombre de demandes de couleurs d_i sur un nœud x_i est généré comme suit :

```

 $\nu = \text{random}(0, 1);$ 
si  $\nu < 0.05$  alors
     $d_i = \text{random}(11, 40);$ 
sinon si  $\nu < 0.15$  alors
     $d_i = \text{random}(2, 10);$ 
sinon
     $d_i = 1;$ 

```

Cette génération irrégulière du nombre de demandes de couleurs est justifiée par les contraintes de problèmes pratiques comme celui de l'allocation de fréquences dans l'aviation civile.

Dans l'algorithme de l'Ordre dynamique, différentes initialisations de l'ordre de rangement des demandes ont conduit la même performance de l'algorithme. Par la suite, l'Ordre dynamique utilise l'ordre initial selon le degré d'espace.

En observant l'évolution du nombre de couleurs utilisées suivant le déroulement de la procédure pour les problèmes de 500 à 1 300 nœuds, nous pouvons noter une descente très rapide du nombre de couleurs utilisées, et puis l'évolution se stabilise. (voir *fig. 4*).

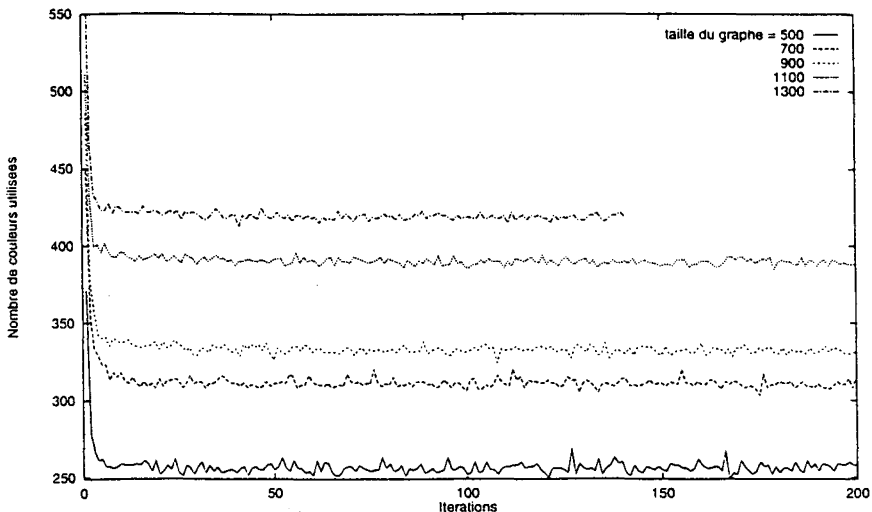


Figure 4. – Évolutions de l'Ordre dynamique pour le coloriage de différentes tailles.

L'évolution de la consommation de temps CPU suivant le changement de la taille du problème et l'influence de condition d'arrêt sur les meilleures solutions obtenues sont données sur le tableau VII et la figure 5. Les résultats obtenus dans le tableau VII sont la moyenne de 5 exécutions de la même procédure pour chaque taille du problème. Nous avons remarqué une légère amélioration des solutions et un très fort accroissement de la consommation de temps CPU dès que n_{max} augmente.

Pour appliquer l'algorithme de recherche Tabou, un paramètre à régler est la durée du régime Tabou $|L_T|$. Pour cela, nous avons appliqué la recherche

TABLEAU VII
Solutions et durées de calcul d'après l'Ordre dynamique.

Problème		Solution en nombre de couleurs utilisées (durée en sec.)			
n	Σd_i	initiale	$n_{max} = 100$	$n_{max} = 50$	$n_{max} = 10$
100	213	103	79 (1,7)	79 (1,4)	79 (1,3)
300	768	248	159 (76,82)	159,8 (42)	162 (15,9)
500	1 525	371	249,7 (344,9)	250,3 (152,7)	252,2 (49)
700	2 050	449	304,8 (629)	304,25 (372,1)	308,2 (90,9)
900	2 282	486	326,6 (773,3)	327,8 (424,1)	330,8 (140,1)
1 100	2 843	511	384,6 (1 254,6)	385,4 (723,1)	389 (239,9)
1 300	3 146	613	413,8 (1 458,3)	414,6 (836,3)	417,4 (305,4)
1 500	3 871	665	496,4 (1 056,7)	498 (1 056,7)	500,8 (335,4)

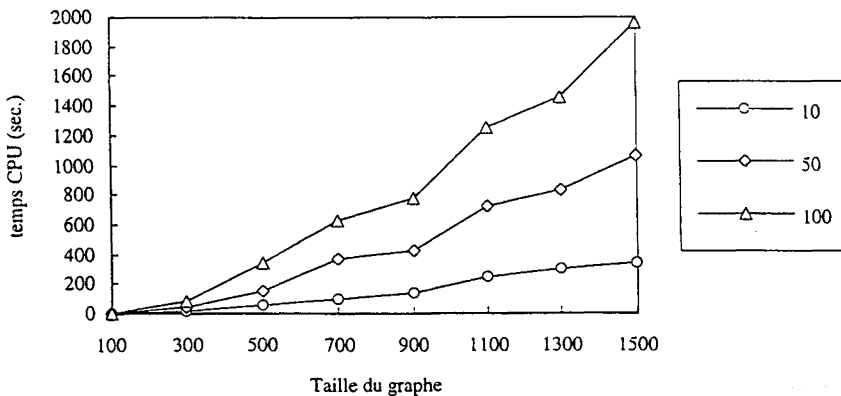


Figure 5. – Évolutions des consommations de temps suivant la taille du problème.

Tabou aux problèmes de taille 100 et 300. La figure 6 montre l'évolution de la meilleure solution obtenue suivant le changement de $|L_T|$. Pour un problème à 100 nœuds, $|L_T|$ va de 2 à 114. On obtient une bonne solution avec 81 couleurs pour $|L_T|$ de 24 à 72. Pour un problème à 300 nœuds, $|L_T|$ va de 14 à 298. La solution obtenue commence à être « bonne (utilise peu de couleurs) » à partir de $|L_T| = 78$; à partir de $|L_T| = 226$, la solution commence à se dégrader; à partir de $|L_T| = 206$, le temps pour obtenir la solution devient de plus en plus long. D'après l'évolution de la figure 6, il nous a semblé intéressant de définir un régime Tabou de durée variable dans un intervalle $[[0, 2n], [0, 7n]]$.

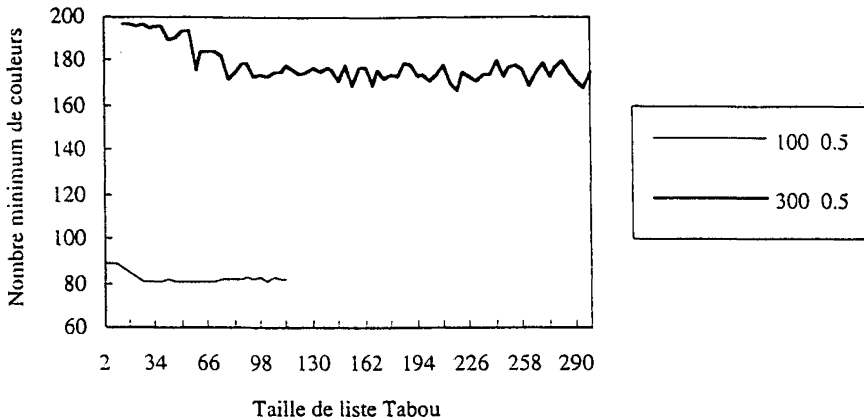


Figure 6. – Évolutions de la meilleure solution suivant la durée du régime Tabou.

Le tableau VIII donne les résultats obtenus par une recherche Tabou pure avec la liste de candidats L_3 qui offre la possibilité de trouver plus rapidement des mouvements améliorant, $n_{max} = 2*n$ et $C_1 = C_2 = 1$. Les solutions initiales sont obtenues par l'algorithme glouton sans itération et par la méthode de l'Ordre dynamique avec $n_{max} = 10$.

Lorsque la recherche Tabou démarre à partir de la solution initiale obtenue par l'algorithme glouton, elle trouve des solutions dont la qualité n'est pas meilleure que celles obtenues par l'Ordre dynamique, avec une consommation de temps CPU moins importante. La recherche Tabou appliquée à l'amélioration de solutions obtenues par l'Ordre dynamique n'échoue jamais. Parmi les coloriage de différentes tailles, tous ont été améliorés par la recherche Tabou, à l'exception de celui de 100 nœuds où la solution par l'Ordre dynamique a déjà atteint la borne inférieure.

TABLEAU VIII

Résultats du coloriage obtenus par la recherche Tabou avec différentes solutions initiales.

n	Solution en nombre de couleurs utilisées (durée en sec.)			
	Glouton	Tabou « pure »	Ordre dynamique $n_{max} = 10$	Tabou « pure »
100	105 (1,28)	81 (12,15)	79 (1,63)	79 (1,63)
300	202 (5,73)	168 (1 624,37)	161 (43,15)	159 (5,65)
500	342 (22,22)	249 (10 617,27)	251 (67,73)	243 (1 957,32)
700	408 (47,72)	301 (38 671,87)	309 (101,77)	298 (1 272,77)
900	431 (73,17)	338 (31 968,33)	331 (138,55)	323 (1 840,73)
1 100	521 (125,58)	–	389 (215,65)	379 (1 394,18)
1 300	558 (172,38)	–	421 (273,27)	410 (2 580,60)
1 500	641 (275,18)	–	500 (389,63)	490 (6 961,62)

6.2. problème d'allocation des fréquences dans l'aviation civile

Appliquons la méthode de l'Ordre dynamique avec comme borne inférieure 760 fréquences (le nombre maximal de fréquences utilisées dans le système COM2), nous arrivons à trouver des solutions en respectant toutes les contraintes d'interférence en temps raisonnable. D'abord, nous avons montré dans la figure 7 une évolution semblable à celles dans la figure 4 pour les problèmes aléatoires. L'évolution converge rapidement, et se stabilise après 50 itérations.

Le tableau IX donne des résultats selon les différents critères d'arrêt. Avec $n_{max} = 100$, nous arrivons à obtenir une meilleure solution respectant toutes les contraintes d'interférence avec 779 fréquences en plus de 4 heures.

TABLEAU IX

Résultats obtenus de la méthode de l'Ordre dynamique.

Critère d'arrêt n_{max}	Meilleure solution	Durée
10	789	35 m 57 s
50	783	1 h 40 m 58 s
100	779	4 h 25 m 41 s

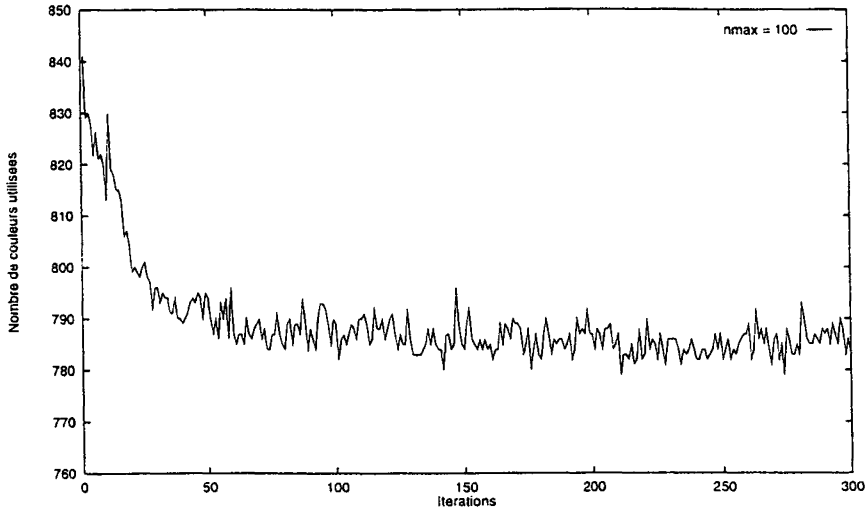


Figure 7. – Évolution de l'Ordre dynamique pour le problème de l'aviation civile ($n_{max} = 100$).

Ce résultat montre que si nous voulions une planification des fréquences dans le système COM2 qui respecte toutes les contraintes d'interférence, il faudrait rajouter au moins 19 fréquences de plus. Ceci montre bien l'insuffisance du nombre de fréquences utilisées par le système COM2.

L'application de la recherche Tabou ne permet pas d'améliorer les solutions obtenues par l'Ordre dynamique. En revanche, elle permet de diminuer le nombre de violations de contraintes d'interférence pour la solution actuellement retenue par l'aviation civile (voir le tableau X). Pour cette solution, la taille du problème exige l'utilisation d'une liste de candidats plus réduite. Nous avons donc utilisé la liste L_4 par sélection consécutive. Trois exécutions de la procédure Tabou avec les différents paramètres (n_{max} , C_1 , C_2) ont été effectuées.

TABLEAU X
Améliorations de la planification dans le système actuel.

Coût initial [†]	Coût final	Durée	Paramètres (n_{max} , C_1 , C_2)
9 400	783	14 h 03 m 02 s	1 000, 1, 1
9 400	501	25 h 38 m 36 s	1 500, 1, 1
9 400	488	27 h 51 m 35 s	1 500, 1, 10

[†] Les coûts ici sont exprimés en nombre de violations de contraintes d'interférence.

Nous savons que 9400 contraintes d'interférence sont violées dans la planification actuelle qui utilise 760 fréquences. La violation de certaines contraintes peut être évitée en échangeant des fréquences pour les services en conflit.

Le tableau X montre que la planification actuelle a pu être améliorée par la recherche Tabou, en réduisant le nombre de violations de contraintes d'interférence de 9400 à 488. Mais il reste encore des contraintes d'interférence violées. Vraisemblablement, le seul moyen de l'éviter est d'augmenter le nombre de fréquences.

7. CONCLUSION

Nous avons évalué le comportement et la performance de l'algorithme de l'Ordre dynamique et de la recherche Tabou pour résoudre des problèmes de T -coloriage avec ensembles. La méthode de l'Ordre dynamique est très performante en temps de calcul, mais la solution obtenue est souvent améliorable par la recherche Tabou. La recherche Tabou a une performance médiocre en temps de calcul, elle est extrêmement lente dans l'obtention d'une solution proche de celle donnée par l'Ordre dynamique. Par contre, elle est redoutable lorsqu'il s'agit d'améliorer une solution obtenue par d'autres moyens. Les meilleures solutions des problèmes de T -coloriage avec ensembles construits aléatoirement, sont obtenues par l'hybridation de la méthode de l'Ordre dynamique et de la recherche Tabou.

Nous avons appliqué les méthodes étudiées au problème d'allocation de fréquences du système COM2 dans l'aviation civile. Notre but était d'étudier le nombre de fréquences nécessaires pour satisfaire les besoins actuels sans violer les contraintes d'interférence. Le meilleur résultat obtenu entraîne un surplus de 19 fréquences par rapport au nombre de fréquences utilisées dans le système actuel. De plus, nous avons montré qu'une amélioration de la planification retenue dans système COM2 est possible en réduisant le nombre des violations de contraintes d'interférence, sans ajouter des fréquences supplémentaires.

RÉFÉRENCES

1. F. Box, A Heuristic Technique for Assigning Frequencies to Mobile Radio Nets, IEEE Transaction on vehicular technology, mai 1978, vol. VT-27, n° 2, p. 57-64.
2. W. K. HALE, Frequency Assignment: Theory and Application, Proceedings of the IEEE, December 1980, 68, n° 12, p. 1498-1573.



3. A. GAMST, Some Lower Bounds for a Class of Frequency Assignment Problems, IEEE Transaction on vehicular technology, février 1986, vol. VT-35, n° 1, p. 8-14.
4. A. HERTZ et D. de WERRA, Using Tabu Search Techniques for Graph Coloring, Computing, 1987, 39, p. 345-351.
5. A. GAMST, A Resource Allocation Technique for FDMA Systems, Alfa Frequenza, 1988, vol. LVII-N.2, p. 89-96.
6. C. FRIDEN, A. HERTZ et D. de WERRA, STABULUS: A Technique of Finding Stable Sets in large Graphs with Tabu Search, 1989, Computing, 42, p. 35-44.
7. F. GLOVER, Tabu search, part I, ORSA Journal on computing, 1989, 1, n° 3, p. 190-206.
8. M. KUBALE, Interval vertex-coloring of a graph with forbidden colors, Discrete Mathematic, 1989, 74, p. 125-136.
9. F. GLOVER, Tabu search, part II, ORSA Journal on computing, 1990, 2, n° 1, p. 4-31.
10. E. TAILLARD, Robust Taboo Search for the Quadratic Assignment Problem, Parallel Computing, 1991, 17, p. 443-455.
11. M. KUBALE, Some results concerning the complexity of restricted colorings of graphs, Discrete Applied Mathematics, 1992, 36, p. 35-46.
12. D. COSTA, On the use of some known methods for T -colorings of graphs, Annals of Operations Research, 1993, 41, p. 343-358.
13. D. KLIMOWICZ et M. KUBALE, Graph coloring by Tabu search and simulated annealing, Archives of Control Sciences, 1993, 2 (XXXVIII), n° 1-2, p. 41-54.
14. D. de WERRA et Y. GAY, Chromatic scheduling and Frequency assignment, Discrete Applied Mathematics, 1994, 49, p. 165-174.