



Titre: Transformations de graphes et nombre de stabilité
Title:

Auteur: Karine Dufresne
Author:

Date: 2005

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Dufresne, K. (2005). Transformations de graphes et nombre de stabilité [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/7361/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/7361/>
PolyPublie URL:

Directeurs de recherche: Alain Hertz, Philippe Galinier, & Odile Marcotte
Advisors:

Programme: Non spécifié
Program:

NOTE TO USERS

This reproduction is the best copy available.

UMI[®]

UNIVERSITÉ DE MONTRÉAL

TRANSFORMATIONS DE GRAPHS ET NOMBRE DE STABILITÉ

KARINE DUFRESNE

DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(MATHÉMATIQUES APPLIQUÉES)

MAI 2005



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-494-01307-9

Our file Notre référence

ISBN: 0-494-01307-9

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

TRANSFORMATIONS DE GRAPHS ET NOMBRE DE STABILITÉ

présenté par: DUFRESNE Karine

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. AUDET Charles, Ph.D., président

M. HERTZ Alain, Doct.ès Sc., membre et directeur de recherche

M. GALINIER Philippe, Doctorat, membre et codirecteur de recherche

Mme MARCOTTE Odile, Ph.D., membre et codirectrice de recherche

M. GENDRON Bernard, Ph.D., membre

REMERCIEMENTS

Je désire tout d'abord remercier mon directeur de recherche, Alain Hertz, et mes codirecteurs, Odile Marcotte et Philippe Galinier, de m'avoir proposé ce sujet qui m'a beaucoup intéressée. La qualité de leur encadrement, leur disponibilité ainsi que l'aide financière qu'ils m'ont accordés furent grandement appréciés. Je tiens également à les remercier pour la patience dont ils ont fait preuve durant la rédaction de ce mémoire.

Je tiens à exprimer ma gratitude envers mon père, Robert Dufresne, et ma mère, Lorraine Desnoyers, qui m'ont encouragée et soutenue tout au long de mes études.

Un merci particulier à mon copain, Benoit Langlois, qui a su être présent et me rassurer à tout moment au cours de ces deux dernières années. De plus, je ne peux passer sous silence la reconnaissance que j'ai envers mes amies Caroline Desjardins et Geneviève Cyr pour avoir fait leurs devoirs de mathématiques à la même table que moi. Finalement, merci à mon amie Marie-France Allard, à mon frère Yannick et aux gens précédemment mentionnés qui font de mon parcours universitaire un agréable souvenir.

RÉSUMÉ

Le problème du stable maximum consiste à trouver, dans un graphe, un sous-ensemble de sommets deux à deux non-adjacents de cardinalité maximum. Cette cardinalité est appelée nombre de stabilité. Ce mémoire porte sur l'étude de la résolution de ce problème NP-dur en utilisant des transformations de graphes. Le but poursuivi par cette approche est de réduire le coût de résolution du problème dans certains graphes.

La notion de transformation de graphes est définie et quelques exemples de transformations sont présentées. Les transformations sont qualifiées d'exactes ou d'inexactes selon leur incidence sur le nombre de stabilité indépendamment du graphe auquel elles sont appliquées. Lorsque la variation du nombre de stabilité est constante peu importe le graphe initial et le graphe résultant de l'application d'une transformation, celle-ci est qualifiée d'exacte. Dans le cas contraire, la transformation est qualifiée d'inexacte. Pour chacun de ces deux types de transformations, une stratégie les employant dans la résolution du problème du stable maximum est présentée.

Dans ce travail, nous avons étudié et expérimenté la stratégie employant des transformations exactes. Une méthode empirique a été développée. Elle permet d'évaluer l'effet des transformations sur le coût de résolution du problème. Étant donné un algorithme de résolution du problème du stable maximum, cette méthode consiste à construire un ensemble d'estimateurs du coût de résolution de l'algorithme en fonction de la structure des graphes. Ensuite, étant donné un graphe et une transformation, l'incidence de celle-ci sur la structure du graphe est observée. Enfin, il s'agit de calculer le gain escompté à l'aide des estimateurs du coût de résolution et d'évaluer le gain effectif. La comparaison de ces deux gains est un indice sur la ca-

pacité de la méthode à prévoir l'impact d'une transformation. Cette méthode a été expérimentée en utilisant trois transformations exactes différentes. Les résultats provenant de cette expérimentation sont présentés et discutés.

Finalement, on qualifie un graphe de séparable lorsqu'il est possible d'obtenir une partition de son ensemble de sommets en deux sous-ensembles tels que le nombre de stabilité du graphe est égal à la somme des nombres de stabilité des graphes induits par ces deux sous-ensembles. Quelques classes de graphes séparables et non séparables sont présentées.

ABSTRACT

The maximum stable set problem consist of finding a subset of pairwise nonadjacent vertices. The maximum cardinality of such a set in a graph is called the stability number of the graph. This master's thesis is about solving this NP-hard problem using graph transformations. With this approach, we seek to reduce the computation cost of the problem in some graphs.

We define graph transformations and present some examples. We also define two types of graph transformations: exact and inexact. A graph transformation is exact if its incidence on the stability number is constant independently of the graph on which it is applied. Otherwise, the graph transformation is inexact. We propose two strategies for solving the maximum stable set problem. One uses exact graph transformations and the other uses inexact graph transformations.

We report and discuss the experimental results of the use of three exact graph transformations before solving the maximum stable set problem. We present the method developed to evaluate the incidence of graph transformations on the computational cost of the problem. First of all, given an algorithm solving the maximum stable set problem, this method consists of determining a set of computation cost estimators in function of the graph structure. Given a graph and a transformation, we observe the incidence of the transformation on the structure of the graph. Then, we estimate the computational cost of solving the problem in the resulting graph with the appropriate estimator and evaluate it with the real computational cost of the algorithm. By comparing the estimated computational cost and the real one, we evaluate whether this method can predict the incidence of graph transformations on the computational cost of the problem.

Finally, we introduce the notion of separable graphs. A graph is said to be separable if we can partition its vertex set into two subsets such that the sum of the stability numbers of the graphs induced by these subsets equals the stability number of the graph. We also show that some classes of graphs are separable and others are not.

TABLE DES MATIÈRES

REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vii
TABLE DES MATIÈRES	ix
LISTE DES TABLEAUX	xiii
LISTE DES FIGURES	xvi
LISTE DES ANNEXES	xix
INTRODUCTION	1
CHAPITRE 1 NOTIONS PRÉLIMINAIRES	5
1.1 Définitions et concepts	5
1.2 Le problème du stable maximum	8
1.2.1 Description du problème	9
1.2.2 Algorithmes de résolution du problème	10
1.2.2.1 Heuristique naïve	10
1.2.2.2 Heuristique gloutonne	10
1.2.2.3 Recherche tabou	11
1.2.2.4 DFMAX	12
CHAPITRE 2 TRANSFORMATIONS DE GRAPHS	14
2.1 Définitions et concepts	14
2.2 Incidence des transformations	19

2.3	Stratégies d'utilisation des transformations	21
2.3.1	Transformations exactes	22
2.3.2	Transformations inexactes	24
2.4	Exemples de transformations inexactes	27
2.4.1	Suppression d'une arête	29
2.4.2	Insertion d'une arête	30
2.4.3	Suppression d'un sommet	30
2.4.4	Insertion d'un sommet	31
2.4.5	Identification de deux sommets	32
2.4.6	Partition du voisinage d'un sommet	34
2.5	Exemples de transformations exactes	37
2.5.1	Réduction simple	38
2.5.2	Réduction simpliciale	40
2.5.3	Suppression d'un sommet dominant	42
2.5.4	Réduction d'un magnet	44
2.5.5	Suppression d'un sommet	46
2.5.6	Suppression d'une arête	48
2.5.7	Insertion d'une arête	49
2.5.8	Réduction d'un BAT	52
2.5.9	Struction inversée	54
CHAPITRE 3 INCIDENCE DE TRANSFORMATIONS DE GRAPHS .		58
3.1	Définitions et concepts	59
3.1.1	Démarche expérimentale	59
3.1.2	Algorithme de résolution	64
3.1.3	Échantillon de graphes	66
3.1.4	Module informatique	67
3.2	Évaluation expérimentale des estimateurs du coût de résolution . .	69

3.3	Séquences de transformations	76
3.3.1	Séquence de structions inversées	76
3.3.2	Séquence de suppressions de sommets dominants	77
3.3.3	Séquence d'insertions d'arêtes	79
3.3.4	Suite de séquences de transformations	80
3.3.5	Incidence sur la structure des graphes	80
3.3.6	Graphes de référence du Second DIMACS Challenge	94
3.4	Évaluation des gains escomptés et réels	97
3.5	Discussion	114
3.5.1	Séquence de transformations	114
3.5.2	Méthode proposée	118
CHAPITRE 4 GRAPHS SÉPARABLES		121
4.1	Définitions et concepts	121
4.2	Quelques classes de graphes séparables	122
4.2.1	Arbres	122
4.2.2	Cycles pairs	123
4.2.3	Cycle additionné à une chaîne	124
4.2.4	Graphes bipartis	125
4.2.5	Graphes parfaits	126
4.2.6	Graphes contenant un point d'articulation	127
4.2.7	Graphes contenant une clique d'articulation	130
4.2.8	Graphes série-parallèles	133
4.3	Quelques classes de graphes non-séparables	138
4.3.1	Cycles impairs	138
4.3.2	Roues impaires	139
4.3.3	Anti-trous	140
CONCLUSION		142

RÉFÉRENCES	146
ANNEXES	149

LISTE DES TABLEAUX

Tableau 2.1	Transformations inexactes appliquées à un graphe $G = (V, E)$	37
Tableau 2.2	Transformations exactes appliquées à un graphe quelconque $G = (V, E)$	56
Tableau 3.1	Exemple de résultats expérimentaux	62
Tableau 3.2	Structure des graphes de référence du Second DIMACS Chal- lenge	94
Tableau 3.3	Incidence de SSD sur la résolution du problème du stable maximum	103
Tableau 3.4	Incidence de IATS sur la résolution du problème du stable maximum	104
Tableau 3.5	Incidence de SI2 sur la résolution du problème du stable maximum	105
Tableau I.1	Estimateurs du coût de résolution du problème du stable maximum par DFMAX mesurés en nombre de retours arrières en fonction de la structure	151
Tableau I.2	Estimateurs du coût de résolution du problème du stable maximum par DFMAX mesurés en secondes en fonction de la structure	152
Tableau I.3	Incidence de SSD sur des graphes initialement de 50 sommets	153
Tableau I.4	Incidence de SSD sur des graphes initialement de 60 sommets	154
Tableau I.5	Incidence de SSD sur des graphes initialement de 70 sommets	155
Tableau I.6	Incidence de SSD sur des graphes initialement de 80 sommets	156
Tableau I.7	Incidence de SSD sur des graphes initialement de 90 sommets	157
Tableau I.8	Incidence de SSD sur des graphes initialement de 100 sommets	158
Tableau I.9	Incidence de IATS sur des graphes initialement de 50 sommets	159
Tableau I.10	Incidence de IATS sur des graphes initialement de 60 sommets	160

Tableau I.11	Incidence de IATS sur des graphes initialement de 70 sommets	161
Tableau I.12	Incidence de IATS sur des graphes initialement de 80 sommets	162
Tableau I.13	Incidence de IATS sur des graphes initialement de 90 sommets	163
Tableau I.14	Incidence de IATS sur des graphes initialement de 100 sommets	164
Tableau I.15	Incidence de SI1 sur des graphes initialement de 50 sommets	165
Tableau I.16	Incidence de SI1 sur des graphes initialement de 60 sommets	166
Tableau I.17	Incidence de SI1 sur des graphes initialement de 70 sommets	167
Tableau I.18	Incidence de SI1 sur des graphes initialement de 80 sommets	168
Tableau I.19	Incidence de SI1 sur des graphes initialement de 90 sommets	169
Tableau I.20	Incidence de SI1 sur des graphes initialement de 100 sommets	170
Tableau I.21	Incidence de SI1 suivie de SSD sur des graphes initialement de 50 sommets	171
Tableau I.22	Incidence de SI1 suivie de SSD sur des graphes initialement de 60 sommets	172
Tableau I.23	Incidence de SI1 suivie de SSD sur des graphes initialement de 70 sommets	173
Tableau I.24	Incidence de SI1 suivie de SSD sur des graphes initialement de 80 sommets	174
Tableau I.25	Incidence de SI1 suivie de SSD sur des graphes initialement de 90 sommets	175
Tableau I.26	Incidence de SI1 suivie de SSD sur des graphes initialement de 100 sommets	176
Tableau I.27	Incidence de SIALEA1 sur des graphes initialement de 100 sommets	177
Tableau I.28	Incidence de SIALEA1 suivie de SI1 sur des graphes initiale- ment de 100 sommets	178
Tableau I.29	Incidence de SIALEA2 sur des graphes initialement de 100 sommets	179

Tableau I.30	Incidence de SIALEA2 suivie de SI2 sur des graphes initialement de 100 sommets	180
--------------	--	-----

LISTE DES FIGURES

Figure 2.1	Exemple de suppression d'un sommet	17
Figure 2.2	Transformation générique d'identification de deux sommets adjacents.	19
Figure 2.3	Suppression et insertion d'une arête	29
Figure 2.4	Identification de deux sommets	34
Figure 2.5	Partition du voisinage d'un sommet	36
Figure 2.6	Réduction simple	39
Figure 2.7	Réduction simpliciale	41
Figure 2.8	Suppression d'un sommet dominant	44
Figure 2.9	Réduction d'un magnet	46
Figure 2.10	Suppression d'un sommet	47
Figure 2.11	Suppression d'une arête	49
Figure 2.12	Insertion d'une arête	51
Figure 2.13	Struction inversée	57
Figure 3.1	Coût de résolution du problème du stable maximum	61
Figure 3.2	Estimateurs du coût de résolution du problème du stable maximum dans des graphes de 50 sommets	72
Figure 3.3	Estimateurs du coût de résolution du problème du stable maximum dans des graphes de 100 sommets	73
Figure 3.4	Estimateurs du coût de résolution du problème du stable maximum	74
Figure 3.5	Nombre de stabilité en fonction de la densité des graphes de 100 sommets	75
Figure 3.6	Coûts de résolution du problème du stable maximum dans des graphes de 100 sommets	75

Figure 3.7	Structure résultant de l'application de SSD à des graphes de 50 sommets	85
Figure 3.8	Structure résultant de l'application de SSD à des graphes de 100 sommets	86
Figure 3.9	Structure résultant de l'application de IATS à des graphes de 50 sommets	87
Figure 3.10	Structure résultant de l'application de IATS à des graphes de 100 sommets	88
Figure 3.11	Structure résultant de l'application de SI2 à des graphes de 50 sommets	89
Figure 3.12	Structure résultant de l'application de SI1 à des graphes de 50 sommets	90
Figure 3.13	Structure résultant de l'application de SI1 à des graphes de 100 sommets	91
Figure 3.14	Structure résultant de l'application de SI1 suivie de SSD à des graphes de 50 sommets	92
Figure 3.15	Structure résultant de l'application de SI2 suivie de SSD à des graphes de 50 sommets	93
Figure 3.16	Coût de résolution du problème du stable maximum dans les graphes résultant de l'application de IATS à des graphes initialement de 50 sommets	106
Figure 3.17	Coût de résolution du problème du stable maximum dans les graphes résultant de l'application de IATS à des graphes initialement de 100 sommets	107
Figure 3.18	Incidence de l'application de SSD à des graphes initialement de 50 sommets sur le coût de résolution du problème du stable maximum	108

Figure 3.19	Incidence de l'application de SSD à des graphes initialement de 100 sommets sur le coût de résolution du problème du stable maximum	109
Figure 3.20	Incidence de l'application de IATS à des graphes initialement de 50 sommets sur le coût de résolution du problème du stable maximum	110
Figure 3.21	Incidence de l'application de IATS à des graphes initialement de 100 sommets sur le coût de résolution du problème du stable maximum	111
Figure 3.22	Incidence de l'application de SSD et de IATS à des graphes initialement de 50 sommets sur le coût de résolution du problème du stable maximum	112
Figure 3.23	Incidence de l'application de SSD et de IATS à des graphes initialement de 100 sommets sur le coût de résolution du problème du stable maximum	113

LISTE DES ANNEXES

ANNEXE I	TABLEAUX DE RÉSULTATS	149
----------	---------------------------------	-----

INTRODUCTION

Le problème du stable maximum est un problème d'optimisation qui consiste à trouver dans un graphe un sous-ensemble de sommets de cardinalité maximum tel que tous les sommets de ce sous-ensemble sont deux à deux non-adjacents. Le problème consistant à trouver la cardinalité d'un tel ensemble est également désigné par ce nom. Ce problème est souvent traité, pour des raisons qui seront évoquées plus loin, conjointement avec le problème de la clique maximum. De plus, en affectant des poids sur les sommets, le problème devient celui du stable de poids maximum.

Les articles cités dans cette introduction le sont à titre d'exemples.

Plusieurs approches peuvent être adoptées pour traiter ce problème NP-dur. Parmi celles-ci, on distingue des méthodes employant des stratégies exactes, d'autres employant des stratégies heuristiques et certaines se penchant sur des classes de graphes en particulier. Plus précisément, certains algorithmes sont développés pour résoudre le problème du stable maximum de manière exacte, ce qui signifie qu'un tel algorithme trouve, en autant qu'on lui en laisse le temps, une solution optimale peu importe le graphe dans lequel il doit résoudre le problème du stable maximum. Cependant, ce type d'algorithmes est tel que le coût de résolution est en temps exponentiel. Ainsi, l'objectif des recherches au niveau de ce type d'algorithmes est de réduire le coût de résolution du problème du stable maximum. Plusieurs stratégies ont été adaptées à la résolution de ce problème. Par exemple, on trouve des algorithmes d'énumération implicite ayant des stratégies d'évaluation ou d'élagage différentes (Östergård, 2002; Carraghan et Pardalos, 1990) et des algorithmes de programmation par contraintes (Régim, 2003).

Il y a également des heuristiques qui trouvent une solution proche de l'optimum, mais sans garantie d'optimalité. Cette approche sert à établir une borne inférieure sur la cardinalité de l'ensemble stable maximum ou à trouver une solution satisfaisante lorsque la connaissance de la valeur optimale n'est pas indispensable. La qualité de la solution dépend de l'heuristique développée. Plusieurs stratégies heuristiques ont été adaptées pour trouver une solution proche de celle du problème du stable maximum. On trouve, entre autres, la recherche tabou (Friden et al., 1989), les réseaux de neurones (Zissimopoulos, 1995) et la reproduction recuite (annealed replication (Bomze et al., 2002)).

D'autres approches se concentrent sur certaines classes de graphes. Plusieurs utilisent cette approche afin de conclure que le problème du stable maximum se résout en temps polynomial dans certaines classes de graphes.

Dans le contexte de la présente recherche, une approche moins directe que celles précédemment mentionnées a été étudiée. Il s'agit en fait de transformer les graphes avant de résoudre le problème du stable maximum à l'aide d'un algorithme existant. Quelques recherches ont été parallèlement effectuées sur des transformations et leurs effets sur le nombre de stabilité. Récemment, un article (Alekseev, 2004) est paru dans lequel un formalisme de la notion de transformation préservant le nombre de stabilité est décrit. Les articles antérieurs au sujet des transformations de graphes et de l'ensemble stable maximum traitent, en général, d'une seule transformation et parfois d'une classe de graphes en particulier.

Ce travail poursuit trois objectifs connexes. Le premier objectif consiste à vérifier la validité de l'approche de résolution du problème du stable maximum utilisant les transformations de graphes. En second lieu, il s'agit d'analyser et d'effectuer des expérimentations sur l'incidence de certaines transformations de graphes au niveau de la structure des graphes et de la résolution du problème du stable max-

imum. Il y a finalement l'étude d'une propriété telle qu'un graphe la possédant peut être séparé en deux composantes connexes et qui permettrait éventuellement l'élaboration d'une transformation.

Une méthode empirique a été utilisée dans la poursuite des deux premiers objectifs. Ainsi, une méthode a été élaborée puis expérimentée en utilisant différentes transformations. L'incidence de certaines transformations de graphes sur la structure des graphes et leur nombre de stabilité a également été analysée de façon théorique. L'étude de la propriété des graphes d'être séparable s'est limitée à la démonstration que certaines classes de graphes possèdent cette propriété et que d'autres ne la possèdent pas.

Ce mémoire comporte quatre chapitres. Le premier permet de réviser quelques notions de la théorie des graphes qui seront utiles à la compréhension tout au long du présent document. Il sert également à décrire de façon plus détaillée le problème du stable maximum et présente quelques algorithmes pouvant être utilisés pour le résoudre.

En second lieu, la notion de transformations de graphes est abordée théoriquement. L'incidence de ces dernières sur le nombre de stabilité est expliquée puis deux stratégies de résolution du problème du stable maximum utilisant les transformations de graphes sont proposées. La présentation d'exemples de transformations de graphes clôt le chapitre.

Les résultats empiriques sont présentés et discutés au troisième chapitre. Ils concernent l'utilisation de trois différentes transformations appliquées aux graphes avant la résolution du problème du stable maximum. En premier lieu, on observe le coût de résolution de l'algorithme en fonction de la structure des graphes. Puis, l'incidence de l'utilisation de transformations est observée sur la structure

des graphes et sur le coût de résolution du problème du stable maximum. Avant la présentation des résultats, certaines notions sont introduites afin de décrire la démarche expérimentale.

Finalement, la propriété de certains graphes d'être séparables est énoncée et expliquée. Puis, nous présentons quelques classes de graphes séparables et d'autres classes de graphes non-séparables.

CHAPITRE 1

NOTIONS PRÉLIMINAIRES

Afin de bien comprendre le problème du stable maximum, il est important d'effectuer une révision succincte de certaines notions de la théorie des graphes. En premier lieu, quelques définitions et notations sont présentées. Ensuite, le problème est énoncé et expliqué. Finalement, quelques algorithmes de résolution de ce problème sont présentés.

1.1 Définitions et concepts

Soit V un ensemble. L'ensemble des parties de V de cardinalité k est noté $\mathcal{P}_k(V)$. Le graphe simple non orienté¹ $G = (V, E)$ est formé d'un ensemble de sommets V et d'un ensemble d'arêtes $E \subseteq \mathcal{P}_2(V)$.

Soit $G = (V, E)$ un graphe. Si $x \in V$ et $y \in V$ sont tels que $\{x, y\} \in E$, alors on dit que x et y sont adjacents. Dans ce cas, x est un voisin de y , y un voisin de x et x et y sont tous deux incidents à l'arête $\{x, y\}$. $N_G(x) = \{u \in V \mid \{u, x\} \in E\}$ est l'ensemble des voisins du sommet x et $d(x) = |N_G(x)|$ est le degré de x . Un sommet est dit isolé si $N_G(x)$ est vide. Lorsqu'il n'y a aucune ambiguïté au niveau du graphe, $N(x)$ peut être utilisé pour représenter l'ensemble des voisins du sommet x dans ce graphe.

Définition 1 *Le **graphe complémentaire** de $G = (V, E)$ est le graphe $\overline{G} =$*

¹Pour la suite de ce document, l'emploi du mot graphe sous-entend graphe simple non orienté.

(V, \overline{E}) ayant le même ensemble de sommets que G et l'ensemble d'arêtes complémentaire, $\overline{E} = \mathcal{P}_2(V) - E$.

Le graphe $G = (V, \mathcal{P}_2(V))$ est appelé graphe complet.

Définition 2 Soit un graphe $G = (V, E)$. Un **sous-graphe induit** de G est un graphe dont l'ensemble de sommets est $W \subseteq V$ et l'ensemble d'arêtes est $\mathcal{P}_2(W) \cap E$. Un **sous-graphe partiel** de G est un graphe dont l'ensemble de sommets est $W \subseteq V$ et l'ensemble d'arêtes est $A \subseteq \mathcal{P}_2(W) \cap E$.

Les définitions suivantes caractérisent certains des sous-ensembles de sommets ou d'arêtes des graphes et spécifient un paramètre de graphes pour chacune de ces caractéristiques.

Définition 3 Soit $G = (V, E)$ un graphe et soit $W \subseteq V$ un sous-ensemble de sommets du graphe. W est un ensemble **stable** si aucune arête de E n'a ses deux extrémités dans W . La cardinalité maximum d'un ensemble stable dans G , appelée **nombre de stabilité**, est notée $\alpha(G)$.

Dans un graphe $G = (V, E)$, un ensemble stable $S \subseteq V$ est dit maximal si pour tout ensemble stable $X \subseteq V$ différent de S , $S \not\subseteq X$.

Définition 4 Soit $G = (V, E)$ un graphe et soit $W \subseteq V$ un sous-ensemble de sommets du graphe. W est une **clique** si pour chaque paire de sommets $\{u, v\} \subseteq W$, on a $\{u, v\} \in E$. La cardinalité de la plus grande clique dans G est notée $\omega(G)$.

Une clique est donc un sous-graphe induit complet. De plus, notons que $S \subseteq V$ est un ensemble stable dans un graphe $G = (V, E)$ si et seulement si S est une clique dans le graphe complémentaire $\overline{G} = (V, \overline{E})$.

Définition 5 Soit $G = (V, E)$ un graphe et soit $W \subseteq V$ un sous-ensemble de sommets du graphe. W est un ensemble **transversal** si pour chaque arête $\{u, v\} \in E$, on a $u \in W$ ou $v \in W$. La cardinalité minimum d'un ensemble transversal dans G est notée $\tau(G)$.

Rappelons qu'un ensemble $S \subseteq V$ d'un graphe $G = (V, E)$ est stable si et seulement si chaque arête de E a au plus une extrémité dans S . On en déduit que $V - S$ est un transversal puisque cet ensemble contient au moins une extrémité de chaque arête de G . On a donc $\alpha(G) + \tau(G) = |V|$.

Définition 6 Soit $G = (V, E)$ un graphe et soit $A \subseteq E$ un sous-ensemble d'arêtes de G . A est un **couplage** si chaque sommet $u \in V$ est incident à au plus une arête de A . La cardinalité maximum d'un couplage dans G est notée $\kappa(G)$.

Définition 7 Soit $G = (V, E)$ un graphe. Une **coloration des sommets** de G est une affectation de couleurs aux sommets de G ayant la propriété que si $\{u, v\} \in E$, alors la couleur du sommet u est différente de celle du sommet v . Le nombre minimum de couleurs nécessaires pour colorer les sommets de G , appelé **nombre chromatique**, est noté $\chi(G)$.

Lorsqu'on a une coloration des sommets du graphe G , alors on obtient une couverture des sommets avec des cliques dans le graphe complémentaire \overline{G} . Le plus petit nombre de cliques nécessaires au recouvrement des sommets est noté $\theta(G)$, et donc $\chi(G) = \theta(\overline{G})$.

Les dernières définitions décrivent des agencements de sommets et d'arêtes présents dans les graphes.

Définition 8 Une **chaîne** est une suite $\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_{k-1}, v_k\}$ d'arêtes. Un **cycle** est une chaîne telle que $v_1 = v_k$.

Une chaîne est élémentaire si elle passe au plus une fois par chaque sommet et est simple si elle passe au plus une fois par chaque arête. Les mêmes qualificatifs s'emploient pour le cycle. Sauf indication contraire, les termes chaîne et cycle désigneront respectivement une chaîne élémentaire et un cycle élémentaire.

Définition 9 Soit P (respectivement C) un sous-graphe partiel qui est une chaîne (respectivement un cycle). Une **corde** dans P (respectivement C) est une arête reliant deux sommets non consécutifs de cette chaîne (respectivement de ce cycle) dans le sous-graphe induit par les sommets de P (respectivement C). Une chaîne sans corde de k sommets est notée P_k et un cycle sans corde de k sommets est noté C_k .

1.2 Le problème du stable maximum

Il existe de nombreuses recherches traitant du problème du stable maximum. Cette section est un résumé se limitant aux notions utiles à la compréhension des chapitres qui suivent. Le problème est décrit et la présentation des algorithmes de recherche d'un ensemble stable que nous avons utilisés à des fins de résolution clôt le chapitre.

Pour un meilleur aperçu des développements récents relatifs au problème du stable maximum, la thèse de Sergiy Butenko (Butenko, 2003) présente un survol des recherches portant sur ce problème tant au niveau de la résolution algorithmique du problème qu'au niveau des applications de celui-ci. Un article moins récent (Bomze et al., 1999) écrit par Immanuel Bomze, Marco Budinich, Panos M. Pardalos et

Marcello Pelillo peut aussi être consulté. Il relate les développements par rapport au problème de la clique maximum qui est étroitement relié à celui du stable maximum.

1.2.1 Description du problème

Le problème du stable maximum est le problème d'optimisation consistant à trouver un ensemble stable $S \subseteq V$ de cardinalité maximum dans un graphe $G = (V, E)$. Il peut aussi être présenté en tant que problème de décision qui considère un graphe $G = (V, E)$ et un entier k et consiste à déterminer s'il existe un ensemble stable $S \subseteq V$ de cardinalité supérieure ou égale à k . Le problème du stable maximum pour les graphes généraux est NP-dur (Garey et Jonhson, 1979). On ne connaît donc pas d'algorithme polynomial pour résoudre toutes les instances de ce problème.

Le problème de la clique maximum consiste à trouver une clique $K \subseteq V$ de cardinalité maximum dans un graphe $G = (V, E)$. Le problème de décision lui étant associé considère un graphe $G = (V, E)$ et un entier k et consiste à déterminer si G contient une clique de cardinalité supérieure ou égale à k . Le problème de la clique maximum équivaut au problème du stable maximum, car une clique dans un graphe est un stable dans le graphe complémentaire. Ces deux problèmes sont donc souvent traités conjointement.

Notons qu'il existe une version du problème du stable maximum considérant des poids sur les sommets. Ce dernier, appelé problème du stable de poids maximum, consiste en la recherche d'un ensemble stable dont la somme des poids des sommets est maximale. Le problème du stable maximum est donc le cas particulier du problème du stable de poids maximum où tous les sommets du graphe ont le même poids.

1.2.2 Algorithmes de résolution du problème

Même s'il est connu que le problème du stable maximum est NP-dur, il existe certaines classes de graphes pour lesquels le problème du stable maximum se résout à l'aide d'un algorithme s'exécutant en temps polynomial (Brandstädt, 2004). Mais puisqu'il y a encore beaucoup de graphes pour lesquels aucun algorithme polynomial n'a été découvert, plusieurs recherches sont effectuées au niveau de la découverte d'algorithmes exacts autant que d'heuristiques. Cette section présente les heuristiques et l'algorithme exact que nous avons utilisés pour la résolution du problème du stable maximum.

1.2.2.1 Heuristique naïve

Étant donné un graphe $G = (V, E)$ et une numérotation des sommets de l'ensemble V , cette heuristique consiste à prendre le sommet $x \in V$ ayant le plus petit numéro, à inclure x dans l'ensemble stable puis à supprimer x et $N(x)$ de G . Ces opérations sont effectuées jusqu'à ce qu'il n'y ait plus aucun sommet dans G .

1.2.2.2 Heuristique gloutonne

L'heuristique gloutonne de recherche d'un ensemble stable maximal est un algorithme constructif. Il diffère de l'heuristique naïve au niveau de la stratégie utilisée pour choisir les sommets à inclure dans l'ensemble stable en construction. Un sommet de plus petit degré est tout d'abord déterminé. Ce sommet qui est celui ayant le moins de voisins est le plus prometteur pour faire partie de l'ensemble stable maximum. Une fois ce sommet trouvé, il est mis dans l'ensemble stable, puis lui et l'ensemble de ses voisins sont retirés du graphe. Ces étapes sont répétées jusqu'à

ce qu'il n'y ait plus de sommets dans le graphe.

1.2.2.3 Recherche tabou

La recherche tabou est une méthode heuristique introduite par Fred Glover à la fin des années 80 pour la résolution de problèmes d'optimisation combinatoire et non-linéaire. La recherche tabou utilisée pour trouver un stable maximal dans un graphe se base sur l'heuristique STABULUS développée par Charles Friden, Alain Hertz et Dominique de Werra (Frident et al., 1989).

Étant donné un graphe $G = (V, E)$ et un entier k , une configuration de cardinalité k est un sous-ensemble de sommets $C \subseteq V$ tel que $C \in \mathcal{P}_k(V)$. La recherche tabou utilisée explore des configurations de même cardinalité dans le but de trouver celle ayant le moins d'arêtes. La première configuration, C_0 , à être explorée est choisie aléatoirement. Considérant une certaine configuration explorée C_i , la suivante, C_{i+1} , est déterminée en retirant un sommet $y \in C_i$ de C_i et en incluant un sommet $x \in V - C_i$ dans C_i .

La recherche est dotée d'une mémoire conservant les configurations explorées précédemment. Cette mémoire est implantée sous forme de deux listes tabous: une conservant les sommets qui ont été retirés d'une configuration et l'autre conservant les sommets qui ont été inclus dans une configuration. Un sommet demeure dans une liste tabou pour un nombre déterminé d'itérations. Pendant qu'il y est, le sommet doit demeurer à l'extérieur des configurations explorées s'il est dans la liste tabou conservant les sommets qui ont été retirés d'une configuration ou à l'intérieur s'il est dans l'autre liste tabou. Bien entendu, la liste tabou conservant les sommets qui ont été retirés d'une configuration ne peut contenir tous les sommets n'appartenant pas à la configuration et de la même façon, la liste tabou conservant

les sommets qui ont été inclus dans une configuration ne peut contenir tous les sommets appartenant à la configuration. Le nombre d'itérations pendant lesquelles chaque sommet demeure dans une liste tabou doit être déterminé en conséquence.

La procédure de recherche tabou considère un graphe $G = (V, E)$, une configuration de départ $C \subseteq V$ de cardinalité donnée et un entier représentant le nombre maximum d'itérations pouvant être effectuées. Elle débute avec la recherche de deux sommets $x \in V - C$ et $y \in C$, si possible adjacents, tels que le sommet x possède le plus petit nombre de voisins dans C et le sommet y possède le plus grand nombre de voisins dans C . La configuration devient $C - \{y\} \cup \{x\}$. Les sommets x et y sont placés dans une liste tabou pour un nombre donné d'itérations. La recherche d'un sommet à retirer de la configuration et d'un sommet à inclure dans celle-ci est réitérée jusqu'à ce que C forme un ensemble stable dans G ou que le nombre d'itérations atteigne le nombre maximum fixé par l'utilisateur. Si C est un ensemble stable, la solution est mémorisée, une configuration ayant une cardinalité d'une unité supérieure à celle de l'ensemble stable est construite, puis la recherche de sommets à retirer de la configuration et à inclure dans celle-ci est recommencée. Si le nombre d'itérations a atteint le nombre maximum fixé par l'utilisateur, la procédure se termine.

1.2.2.4 DFMAX

L'algorithme DFMAX est disponible en ligne sur le site du "second DIMACS challenge" (Trick, 1993). C'est un algorithme exact de recherche d'une clique maximum très similaire à l'algorithme de Randy Carraghan et Panos M. Pardalos (Carraghan et Pardalos, 1990). Il a été écrit en langage C par David Applegate et David Johnson en 1983. Par la suite, il fut modifié en 1988 pour permettre de trouver un stable maximum. En 1993, il fut adapté pour la lecture des fichiers contenant les données

relatives à un graphe encodés selon le standard du “second DIMACS challenge”.

Cet algorithme implémente une méthode de séparation et d'évaluation progressive (branch-and-bound) simple. L'évolution de son exécution est indiquée à l'écran en imprimant la taille de la plus grande clique trouvée et le temps d'exécution chaque fois qu'il est établi qu'un sommet n'appartient pas à une clique maximum. Les résultats finaux affichés à l'écran sont le temps de résolution, la taille de la plus grande clique et les sommets inclus dans cette clique. Nous avons intégré un compteur de retours arrières au programme pour permettre l'analyse des résultats au chapitre 3. Le nombre de retours arrières effectués est indiqué à l'écran une fois que la clique maximum est trouvée.

Soit $G = (V, E)$, un graphe pour lequel une clique maximum est recherchée. L'algorithme DFMAX considère un sous-ensemble de sommets $U \subseteq V$ et une clique $K \subseteq V$ qui ont comme valeurs initiales respectives V et \emptyset . Il insère dans la clique un sommet susceptible de former une clique de grande taille dans G et retire de U ce sommet ainsi que tous les sommets qui ne sont pas dans son voisinage. Lorsque U est vide et que la clique K est de cardinalité supérieure à la dernière trouvée, K est sauvegardée. La stratégie d'élagage d'un noeud est d'effectuer un retour arrière lorsque $|U| + |K|$ est plus petit que la cardinalité de la plus grande clique trouvée à présent. Il est établi qu'un sommet n'appartient pas à la clique maximum lorsqu'il est de degré inférieur à la plus grande clique trouvée. DFMAX explore les sommets en ordre décroissant de degrés.

CHAPITRE 2

TRANSFORMATIONS DE GRAPHERS

Ce chapitre a pour but de définir plus précisément la notion de transformations de graphes et leurs impacts sur le nombre de stabilité. La première section introduit les notions d'opérations élémentaires, de transformations spécifiques et génériques. La seconde section décrit la façon dont l'effet d'une transformation sur le nombre de stabilité d'un graphe est quantifié. À la section suivante, deux stratégies d'utilisation des transformations pour la résolution du problème du stable sont élaborées. Les deux dernières sections sont consacrées à la présentation de différentes transformations de graphes.

2.1 Définitions et concepts

Une transformation de graphes modifie la structure du graphe auquel elle est appliquée. Elle consiste en une séquence d'opérations élémentaires qui sont de quatre types différents:

- l'ajout d'un sommet isolé,
- l'élimination d'un sommet isolé,
- l'ajout d'une arête,
- l'élimination d'une arête.

Étant donnée une numérotation des sommets du graphe, la renumérotation d'un sommet, consistant à donner le nouveau numéro j à un sommet possédant le numéro i , peut être considérée comme un cinquième type d'opération élémentaire. Cependant, cette dernière n'a aucun effet sur la structure des graphes.

Définition 10 Appliquée à un graphe $G = (V, E)$ tel que $x \notin V$, l'opération élémentaire d'**ajout du sommet isolé** x , notée aS , génère un graphe $G' = (V', E') = aS(G, x)$ tel que $V' = V \cup \{x\}$ et $E' = E$.

Définition 11 Appliquée à un graphe $G = (V, E)$ tel que $x \in V$ avec $N(x) = \emptyset$, l'opération élémentaire d'**élimination du sommet isolé** x , notée eS , génère un graphe $G' = (V', E') = eS(G, x)$ tel que $V' = V - \{x\}$ et $E' = E$. eS est l'opération élémentaire réciproque de aS .

Notons que l'opération élémentaire d'élimination d'un sommet doit uniquement être effectuée lorsque le sommet est isolé, car éliminer un sommet dont le voisinage est non vide engendre une structure qui n'est pas un graphe. Cette structure possède des arêtes dont une extrémité n'est incidente à aucun sommet.

Définition 12 Appliquée à un graphe $G = (V, E)$ tel que $x \in V$, $y \in V$, $x \neq y$ et $\{x, y\} \notin E$, l'opération élémentaire d'**ajout de l'arête** $\{x, y\}$, notée aA , génère un graphe $G' = (V', E') = aA(G, \{x, y\})$ tel que $V' = V$ et $E' = E \cup \{\{x, y\}\}$.

Définition 13 Appliquée à un graphe $G = (V, E)$ tel que $x \in V$, $y \in V$ et $\{x, y\} \in E$, l'opération élémentaire d'**élimination de l'arête** $\{x, y\}$, notée eA , génère un graphe $G' = (V', E') = eA(G, \{x, y\})$ tel que $V' = V$ et $E' = E - \{\{x, y\}\}$. eA est l'opération élémentaire réciproque de aA .

Les opérations élémentaires peuvent uniquement effectuer des modifications de base sur les graphes. Il est toutefois possible de les insérer dans une séquence de façon à obtenir des modifications plus complexes de la structure des graphes.

Définition 14 Une **transformation spécifique** d'un graphe est une séquence d'opérations élémentaires entièrement définie par le graphe auquel elle s'applique et un ensemble de paramètres de la transformation, sommets ou arêtes, impliqués dans les opérations élémentaires¹ de la séquence.

L'application d'une transformation spécifique à un graphe G , le graphe initial, implique, en posant $G_0 = G$, que la i^e opération de la séquence définissant la transformation spécifique soit appliquée au graphe G_{i-1} ce qui génère le graphe G_i . Le dernier graphe à être généré, appelé graphe transformé, est noté $G' = T(G, \bullet)$ où \bullet représente des paramètres de transformation impliqués dans les opérations de la séquence. Ces paramètres sont les sommets ou les arêtes ne pouvant être déduits de G ou de la transformation spécifique.

Exemple 1 Soit $G = (V, E)$ un graphe. La transformation de suppression d'un sommet $x \in V$, contrairement à l'opération d'élimination d'un sommet isolé, ne requiert pas que ce dernier soit isolé. La séquence définissant cette transformation doit donc contenir les opérations d'élimination des arêtes incidentes au sommet x . Pour déterminer les arêtes à supprimer, il suffit de connaître le graphe G et le sommet à supprimer x . Ainsi, le graphe résultant de la suppression du sommet x du graphe G est noté $T_{ss}(G, x)$.

Le développement des transformations spécifiques en opérations permet d'obtenir automatiquement la transformation réciproque. La réciproque d'une transforma-

¹Afin d'alléger le texte, pour la suite, le terme opération sera utilisé pour désigner opération élémentaire.

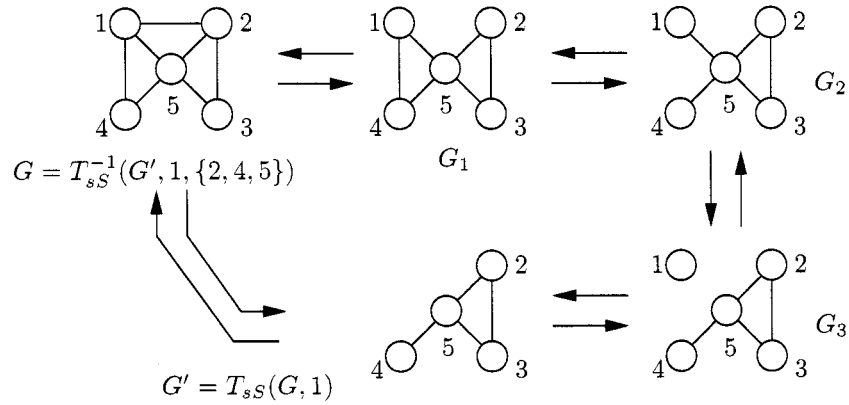


Figure 2.1 Transformation spécifique de suppression d'un sommet

tion spécifique est une séquence d'opérations construite à partir de la séquence définissant la transformation spécifique d'origine. L'ordre des opérations de cette dernière est inversé et chaque opération est remplacée par sa réciproque. Soit $G = (V, E)$ un graphe et $G' = T(G, \bullet)$ un graphe résultant de l'application de la transformation T à G . On a $T^{-1}(G', \bullet) = G$ où l'ensemble des paramètres de la transformation réciproque peut être différent de celui de la transformation initiale.

Exemple 2 Soit $G = (V, E)$, le graphe de la figure 2.1. La séquence des opérations présentée dans cette figure décrit la transformation spécifique de suppression du sommet 1 du graphe G . Tout d'abord, l'opération d'élimination de l'arête $\{1, 2\}$ est appliquée à G . Au graphe obtenu, $G_1 = eA(G, \{1, 2\})$, est appliquée l'opération éliminant l'arête $\{1, 4\}$. Ensuite, l'opération d'élimination de l'arête $\{1, 5\}$ est appliquée au graphe $G_2 = eA(G_1, \{1, 4\})$. Finalement, le sommet 1 est éliminé du graphe $G_3 = eA(G_2, \{1, 5\})$ d'où résulte le graphe transformé $G' = eS(G_3, 1) = T_{sS}(G, 1)$.

La réciproque de la transformation de suppression du sommet 1 du graphe G débute avec l'opération d'ajout du sommet 1, isolé, dans G' . Ensuite, l'opération d'ajout de

l'arête $\{1, 5\}$ au graphe $G_3 = aS(G', 1)$ est effectuée, suivie de l'opération d'ajout de l'arête $\{1, 4\}$ au graphe $G_2 = aA(G_3, \{1, 5\})$, puis de l'opération d'ajout de l'arête $\{1, 2\}$ au graphe $G_1 = aA(G_2, \{1, 4\})$, d'où résulte le graphe $G = aA(G_1, \{1, 2\}) = T_{sS}^{-1}(G', 1, \{1, 4, 5\})$. La réciproque correspond donc à l'insertion du sommet 1 avec l'ensemble de voisins $\{2, 4, 5\}$.

Les transformations spécifiques modifient la structure d'un graphe particulier. Toutefois, dans un contexte plus général, il est possible que le graphe initial soit quelconque ou que, pour un graphe donné, les paramètres ne soient pas spécifiés. Dans ce cas, la séquence des opérations ne peut être définie précisément.

Définition 15 Par **transformation générique** de graphes, notée T , nous désignerons l'ensemble des transformations spécifiques qui modifient un graphe donné en suivant des règles précises.

Par exemple, la suppression d'un sommet sera considérée comme une transformation générique. Elle consiste à appliquer la transformation spécifique de l'exemple 2 à un graphe $G = (V, E)$ en choisissant un sommet quelconque $x \in V$.

Lorsque G est précisé, on peut associer à toute transformation générique T , un ensemble, noté $T(G)$, de graphes pouvant être obtenus en appliquant à G une transformation spécifique appartenant à T .

Exemple 3 L'identification de deux sommets adjacents, notée T''_{idS} , est la transformation générique qui consiste à supprimer ces deux sommets, à en ajouter un nouveau et à relier ce dernier à chaque sommet qui était voisin d'au moins un des deux sommets supprimés. Cette transformation est décrite plus en détail à la section 2.4.5.

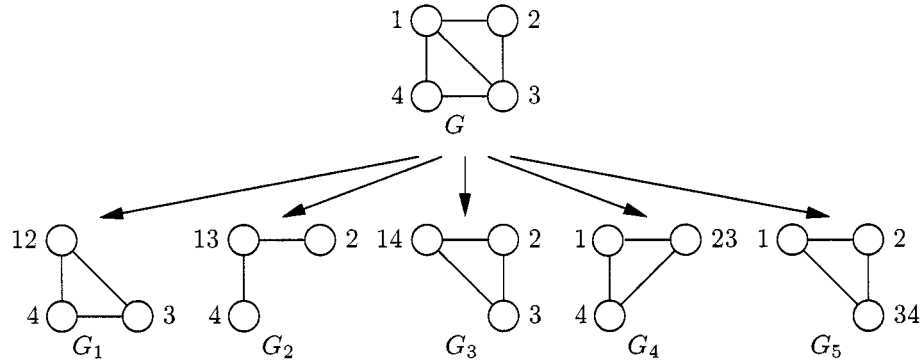


Figure 2.2 Transformation générique d'identification de deux sommets adjacents dans le graphe G

La figure 2.2 représente un graphe G ainsi que les cinq graphes appartenant à $T''_{idS}(G)$. Le graphe $G_1 = T''_{idS}(G, 1, 2)$ résulte de la transformation spécifique d'identification des sommets 1 et 2. De même, on obtient $G_2 = T''_{idS}(G, 1, 3)$, $G_3 = T''_{idS}(G, 1, 4)$, $G_4 = T''_{idS}(G, 2, 3)$ et $G_5 = T''_{idS}(G, 3, 4)$.

Soit $G = (V, E)$, un graphe. L'application itérative d'une transformation générique T à G consiste à générer une suite G_0, G_1, \dots, G_k telle que $G_0 = G$ est le graphe initial et $G_i \in T(G_{i-1})$ est le graphe résultant de la i^e itération.

2.2 Incidence des transformations

L'application d'une transformation spécifique à un graphe G introduit dans ce graphe ou retire de ce graphe des sommets ou arêtes, produisant ainsi un graphe transformé G' . La valeur du nombre de stabilité, celle de la cardinalité de la plus grande clique, celle du nombre chromatique et celle d'autres paramètres de G' peut être différente de la valeur du paramètre correspondant dans G . Dans ce qui suit, il sera question du nombre de stabilité. Il est toutefois possible de suivre un

raisonnement similaire pour la cardinalité de la plus grande clique, pour le nombre chromatique et pour les autres paramètres.

Étant donné une transformation générique T et un graphe $G = (V, E)$, considérons deux graphes différents G' et G'' appartenant à $T(G)$. On a donc deux constantes entières, k et ℓ , telles que

$$k = \alpha(G') - \alpha(G) \text{ et } \ell = \alpha(G'') - \alpha(G).$$

Il est possible que la valeur de k soit différente de celle de ℓ .

Exemple 4 *Considérons le graphe initial G de la figure 2.2 et les graphes transformés G_1 et G_2 . On a*

$$k = \alpha(G_1) - \alpha(G) = -1 \text{ et } \ell = \alpha(G_2) - \alpha(G) = 0.$$

Lors de l'étude de l'incidence des transformations sur le nombre de stabilité, l'intérêt d'une transformation spécifique est limité, car son incidence est constatée à partir d'un graphe initial donné et de l'unique graphe transformé généré. Afin de connaître les conséquences générales d'une transformation sur le nombre de stabilité, l'emploi des transformations génériques est approprié.

L'incidence d'une transformation générique T sur le nombre de stabilité est décrite par la variation de celui-ci lors de l'application de T à un graphe quelconque. Nous nous intéresserons donc à borner la différence entre $\alpha(G')$ et $\alpha(G)$, où G est un graphe quelconque et $G' \in T(G)$. En particulier, nous considérons des transformations génériques T pour lesquelles il existe au moins une des deux constantes $k \in \mathbb{Z}$ et $k' \in \mathbb{Z}$ telles que

$$\alpha(G) + k \leq \alpha(G') \leq \alpha(G) + k' \tag{2.1}$$

pour tout G et pour tout $G' \in T(G)$.

Définition 16 Une transformation T pour laquelle il existe une constante $k \in \mathbb{Z}$ telle que

$$\alpha(G') = \alpha(G) + k \quad (2.2)$$

pour tout G et pour tout $G' \in T(G)$ est dite **exacte**. Lorsqu'une telle constante n'existe pas, la transformation est dite **inexacte**.

Exemple 5 Considérons la transformation générique T de suppression d'un sommet. Supprimer un sommet dans un graphe implique l'élimination de toutes les arêtes auxquelles ce sommet est incident suivie de l'élimination de ce sommet. Cette transformation ne cause jamais d'augmentation du nombre de stabilité et peut le faire diminuer au maximum d'une unité. Ainsi, pour un graphe G et pour un graphe transformé $G' \in T(G)$, on a

$$\alpha(G) - 1 \leq \alpha(G') \leq \alpha(G).$$

La suppression d'un sommet est une transformation inexacte.

2.3 Stratégies d'utilisation des transformations

Une stratégie d'utilisation des transformations élabore une méthode utilisant un graphe transformé afin de déterminer le nombre de stabilité du graphe initial. Par graphe transformé, nous entendons un graphe résultant de l'application d'une séquence de transformations à un graphe initial. Deux stratégies d'utilisation sont présentées, une employant les transformations exactes et l'autre employant les transformations inexactes.

2.3.1 Transformations exactes

Soit T une transformation générique exacte. Il existe donc une constante $k \in \mathbb{Z}$ telle que pour tout graphe G et tout graphe $G' \in T(G)$, $\alpha(G') - \alpha(G)$ est égal à k . Dans ce cas, le nombre de stabilité de G' peut être déduit du nombre de stabilité de G et vice versa. Cette déduction est le point de départ de la stratégie d'utilisation des transformations exactes que nous présentons.

Tout d'abord, il faut déterminer le temps t alloué à l'exécution de l'algorithme de résolution du problème du stable maximum. La stratégie utilisant une transformation générique exacte T pour tenter de déterminer le nombre de stabilité du graphe G s'effectue en trois étapes.

La première étape de la stratégie consiste à produire le graphe transformé G' en appliquant itérativement T à G . Notons η_T le nombre d'applications nécessaires à l'obtention de G' .

La seconde étape vise à déterminer le nombre de stabilité de G' à l'aide de l'algorithme de résolution du problème du stable maximum. Si la solution n'a pas été trouvée en un temps inférieur à t , la stratégie a échoué.

À la dernière étape, le nombre de stabilité de G est déterminé par

$$\alpha(G) = \alpha(G') + (\eta_T \cdot k)$$

où k est la constante entière de l'équation 2.2 associée à T .

Il est simple de prouver par induction que la stratégie présentée est valide. Considérons le graphe G , le graphe G' résultant de l'application itérative d'une transformation générique exacte T à G et k la constante entière de l'équation 2.2 associée

à la transformation T . Notons G_i le graphe résultant de la i^e itération et posons $G_0 = G$.

$$1^{ère} \text{ itération: } \alpha(G_1) = \alpha(G_0) + k.$$

$$\text{Donc } \alpha(G_1) = \alpha(G_0) + (1 \cdot k).$$

$$i^e \text{ itération: } \alpha(G_i) = \alpha(G_{i-1}) + k.$$

$$\text{On suppose que } \alpha(G_i) = \alpha(G_{i-1}) + (i \cdot k).$$

$$i + 1^e \text{ itération: } \alpha(G_{i+1}) = \alpha(G_i) + k.$$

$$\text{Donc } \alpha(G_{i+1}) = (\alpha(G_{i-1}) + (i \cdot k)) + k = \alpha(G_i) + ((i + 1) \cdot k).$$

L'utilisation d'une telle stratégie requiert le choix d'un algorithme \mathcal{A} de résolution du problème du stable maximum et d'une transformation générique exacte T . Nous avons mesuré la taille d'un graphe en fonction de son nombre de sommets et de sa densité. La densité d'un graphe $G = (V, E)$ représente le rapport entre le nombre d'arêtes de G et le nombre d'arêtes du graphe complet de $|V|$ sommets. Pour l'algorithme \mathcal{A} choisi, nous mesurons la taille de l'entrée en fonction de la taille du graphe.

La pertinence de cette stratégie d'utilisation des transformations exactes réside dans la diminution du temps requis pour déterminer le nombre de stabilité d'un graphe. Afin de tenter de déterminer si la stratégie est avantageuse pour résoudre le problème du stable maximum dans un graphe, il doit être possible d'estimer le temps nécessaire à son exécution.

Par conséquent, il doit être possible d'estimer le temps d'exécution de \mathcal{A} pour résoudre le problème du stable maximum dans un graphe en fonction de la taille de ce graphe. Pour la transformation T , il doit être possible d'estimer le temps nécessaire à la production d'un graphe transformé ainsi que le temps d'exécution de \mathcal{A} pour résoudre le problème du stable maximum dans le graphe transformé.

Plus précisément, l'estimation du temps pour produire un graphe transformé dépend

de la manière dont T est appliquée itérativement. Étant donné un graphe G , on doit pouvoir estimer le temps moyen requis pour appliquer une transformation spécifique appartenant à T et estimer le nombre d'itérations nécessaires pour obtenir le graphe transformé G' par l'application itérative de T à G en fonction de la taille de G .

De plus, l'estimation du temps d'exécution de \mathcal{A} pour résoudre le problème du stable maximum dans un graphe transformé dépend de la taille de ce graphe. L'incidence de T sur la taille d'un graphe doit être connue. Étant donné un graphe G , le nombre estimé d'itérations nécessaires pour obtenir le graphe transformé G' par l'application itérative de T à G permet d'estimer la taille de G' .

Pour conclure, cette stratégie semble avantageuse lorsque l'estimation du temps d'exécution de \mathcal{A} pour résoudre le problème du stable maximum dans le graphe résultant de l'application itérative de T à un graphe initial est inférieure à l'estimation du temps d'exécution de \mathcal{A} pour résoudre le problème dans le graphe initial. Il faut également que la différence entre ces deux estimations de temps soit supérieure à l'estimation du temps de transformation du graphe initial par l'application itérative de T .

2.3.2 Transformations inexactes

Soit T une transformation générique inexacte. S'il existe une constante $k \in \mathbb{Z}$ telle que pour tout graphe G et pour tout graphe $G' \in T(G)$, on a

$$\alpha(G) + k \leq \alpha(G'), \quad (2.3)$$

alors il est possible d'utiliser T afin de tenter de déterminer le nombre de stabilité de G .

Il faut préalablement déterminer le temps maximum t_1 alloué à l'exécution de l'heuristique de recherche d'un ensemble stable et un temps maximum t_2 alloué à l'exécution de l'algorithme exact de résolution du problème du stable maximum. La stratégie utilisant une transformation générique inexacte T pour tenter de déterminer le nombre de stabilité du graphe G s'effectue en quatre étapes.

La stratégie débute par la recherche d'un ensemble stable dans le graphe G à l'aide de l'heuristique. Notons A la cardinalité de l'ensemble stable trouvé. Si aucun ensemble stable n'a été trouvé en un temps inférieur à t_1 , la stratégie a échoué.

Ensuite, posons $G_0 = G$ et $A_0 = A$. La deuxième étape consiste à générer une suite $G_0, G_1, \dots, G_{\eta_T}$ telle que $G_i \in T(G_{i-1})$. G_i est le i^e graphe de la suite si l'heuristique trouve un ensemble stable dans G_i de cardinalité $A_{i-1} + k$ où A_{i-1} est la cardinalité de l'ensemble stable trouvé par l'heuristique dans G_{i-1} et k est la constante entière de l'équation 2.3 associée à T . Le graphe transformé, noté G' , résultant de cette étape est le dernier graphe de la suite.

Le troisième étape vise à déterminer B , le nombre de stabilité de G' , à l'aide de l'algorithme de résolution du problème du stable maximum. Si la solution n'a pas été trouvée en un temps inférieur à t_2 , la stratégie a échoué.

Lorsque le nombre de stabilité de G' est déterminé, la dernière étape consiste à valider que le nombre de stabilité du graphe initial est A . Ceci s'effectue en vérifiant la valeur de vérité de

$$B - A = \eta_T \cdot k. \quad (2.4)$$

Cette stratégie, bien qu'elle emploie une heuristique de recherche d'un ensemble stable et une transformation générique inexacte, peut résoudre de manière exacte

le problème du stable maximum. Le nombre de stabilité du graphe initial est trouvé lorsque la stratégie n'échoue pas avant la dernière étape et qu'à cette étape, l'égalité 2.4 est vérifiée. C'est-à-dire que la différence entre le nombre de stabilité B du graphe transformé résultant de la deuxième étape et la cardinalité A , dans le graphe initial, de l'ensemble stable trouvé par une heuristique à la première étape est d'exactly k multipliée par le nombre de graphes transformés de la suite générée à la deuxième étape. Dans ce cas, l'ensemble stable trouvé par l'heuristique à la première étape est un ensemble stable maximum.

En effet, considérons le graphe G , le dernier graphe G' de la suite de η_T graphes générée en utilisant des transformations spécifiques appartenant à la transformation générique inexacte T et k la constante entière de l'équation 2.3 associée à T . De la première étape, on a $A \leq \alpha(G)$, où A est la cardinalité de l'ensemble stable dans G trouvé par l'heuristique. La deuxième étape détermine une borne inférieure sur le nombre de stabilité de G' : $\alpha(G) + (\eta_T \cdot k) \leq \alpha(G')$. À la troisième étape, on calcule $B = \alpha(G')$. Finalement, à la quatrième étape, si $B - A = \eta_T \cdot k$, on peut conclure que $\alpha(G) = A$, car

$$\begin{aligned} & (\alpha(G) \geq A) \wedge (\alpha(G') \geq \alpha(G) + \eta_T \cdot k) \wedge (\alpha(G') = B) \\ \Rightarrow & (\alpha(G) \geq A) \wedge (B - \eta_T \cdot k \geq \alpha(G)) \\ \Rightarrow & A \leq \alpha(G) \leq B - (\eta_T \cdot k). \end{aligned}$$

L'utilisation de cette stratégie requiert le choix d'une heuristique \mathcal{H} de recherche d'un ensemble stable maximal, d'une transformation générique inexacte T et d'un algorithme \mathcal{A} de résolution du problème du stable maximum. On considère que la taille de l'entrée de \mathcal{A} et celle de l'entrée de \mathcal{H} sont la taille du graphe en entrée. La taille d'un graphe peut être décrite par la cardinalité de son ensemble de sommets

et celle de son ensemble d'arêtes.

Dans certains cas, il peut être possible de réduire le temps de calcul du nombre de stabilité d'un graphe en utilisant la stratégie précédemment décrite. Supposons qu'il soit possible d'estimer le temps d'exécution de \mathcal{A} et celui de \mathcal{H} en fonction de la taille du graphe en entrée. Étant donné un graphe G et en considérant l'impact de T sur l'ensemble de sommets et l'ensemble d'arêtes de G , il est possible d'estimer le temps d'exécution de \mathcal{A} pour résoudre le problème du stable maximum dans le graphe transformé G' résultant de l'étape 2. Il est probable que le temps de calcul du nombre de stabilité soit moindre en utilisant la stratégie si l'estimation du temps d'exécution de \mathcal{A} pour résoudre le problème du stable dans G' est inférieure à l'estimation du temps d'exécution de \mathcal{A} pour résoudre le problème dans G . Cette stratégie est avantageuse pour résoudre le problème du stable maximum dans un graphe lorsque le temps total nécessaire à l'exécution des quatre étapes est inférieur au temps de l'algorithme exact de résolution du problème du stable maximum dans G .

2.4 Exemples de transformations inexactes

Une transformation générique T est inexacte si la différence entre $\alpha(G)$ et $\alpha(G')$ dépend du graphe G choisi et du graphe $G' \in T(G)$. Six transformations inexactes sont présentées dans cette section. Pour chacune de ces transformations, la variation du nombre de stabilité est mentionnée et expliquée lorsqu'il n'est pas trivial de la déduire. À titre indicatif, la variation du nombre chromatique est également mentionnée.

Procédure de recherche du nombre de stabilité de G

Entrée : Un graphe G

Sortie : Un booléen indiquant si $\alpha(G) = A$

Étape 1:

appliquer une heuristique afin de déterminer A , la cardinalité d'un ensemble stable dans G .

si l'heuristique n'a pas trouvé d'ensemble stable en un temps t_1 **alors**

└ la procédure a échoué.

Étape 2:

$\eta_T \leftarrow 0$

$G' \leftarrow G$

$A' \leftarrow A$

boucle condition d'arrêt déterminée par l'utilisateur

┌ choisir $G'' \in T(G')$

┌ appliquer une heuristique afin de déterminer A'' , la cardinalité d'un ensemble stable dans G'' .

┌ **si** l'heuristique n'a pas trouvé d'ensemble stable en un temps t_1 **alors**

└ la procédure a échoué.

┌ **sinon si** $A'' = A' + k$ **alors**

└ $\eta_T \leftarrow \eta_T + 1$

└ $G' \leftarrow G''$

└ $A' \leftarrow A' + k$

Étape 3:

appliquer un algorithme afin de déterminer B , le nombre de stabilité de G' .

si l'algorithme n'a pas trouvé d'ensemble stable en un temps t_2 **alors**

└ la procédure a échoué.

Étape 4:

si $B - (\eta_T \cdot k) = A$ **alors**

└ $\alpha(G) = A$

sinon

└ $\alpha(G) \geq A$

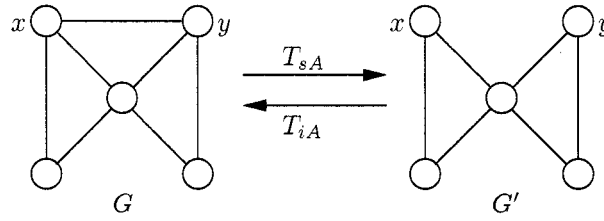


Figure 2.3 Transformation spécifique de suppression et d'insertion d'une arête

2.4.1 Suppression d'une arête

La suppression d'une arête, notée T_{sA} , est parmi les transformations les plus simples. Elle consiste à retirer une arête d'un graphe.

Pour un graphe $G = (V, E)$ et pour une paire de sommets $\{x, y\} \subseteq V$ adjacents dans G , le graphe résultant de la transformation spécifique de suppression de l'arête $\{x, y\}$ dans G est noté $T_{sA}(G, \{x, y\})$.

Entrées : Un graphe $G = (V, E)$ et une arête $\{x, y\} \in E$ Sortie : Un graphe $G' = T_{sA}(G, \{x, y\})$ début $G \leftarrow eA(G, \{x, y\})$ fin

Algorithme 2.2 : Séquence des opérations de la transformation spécifique de suppression de l'arête $\{x, y\}$ du graphe G

Proposition 17 Soit G un graphe et soit $G' \in T_{sA}(G)$. On a

$$\alpha(G) \leq \alpha(G') \leq \alpha(G) + 1,$$

$$\gamma(G) - 1 \leq \gamma(G') \leq \gamma(G).$$

2.4.2 Insertion d'une arête

L'insertion d'une arête, notée T_{iA} , est la transformation réciproque de la suppression d'une arête. Elle consiste à introduire une arête dans un graphe.

Pour un graphe $G = (V, E)$ et pour une paire de sommets $\{x, y\} \subseteq V$ non adjacents dans G , le graphe résultant de la transformation spécifique d'insertion de l'arête $\{x, y\}$ dans G est noté $T_{iA}(G, \{x, y\})$.

Entrées : Un graphe $G = (V, E)$ et une paire de sommets $\{x, y\} \subseteq V$
Sortie : Un graphe $G' = T_{iA}(G, \{x, y\})$
début
 | $G \leftarrow aA(G, \{x, y\})$
fin

Algorithme 2.3 : Séquence des opérations de la transformation spécifique d'insertion de l'arête $\{x, y\}$ dans le graphe G

Proposition 18 Soit G un graphe et soit $G' \in T_{iA}(G)$. On a

$$\alpha(G) - 1 \leq \alpha(G') \leq \alpha(G),$$

$$\gamma(G) \leq \gamma(G') \leq \gamma(G) + 1.$$

2.4.3 Suppression d'un sommet

La suppression d'un sommet, notée T_{sS} , consiste à retirer d'un graphe toutes les arêtes auxquelles un sommet est incident et à retirer ce sommet.

Pour un graphe $G = (V, E)$ et pour un sommet $x \in V$, le graphe résultant de la transformation spécifique de suppression du sommet x de G est noté $T_{sS}(G, x)$.

```

Entrées : Un graphe  $G = (V, E)$  et un sommet  $x \in V$ 
Sortie : Un graphe  $G' = T_{sS}(G, x)$ 
début
  |   pour chaque sommet  $v \in N(x)$  faire
  |   |    $G \leftarrow eA(G, \{x, v\})$ 
  |    $G \leftarrow eS(G, x)$ 
fin

```

Algorithme 2.4 : Séquence des opérations de la transformation spécifique de suppression du sommet x dans le graphe G

Proposition 19 Soit G un graphe et soit $G' \in T_{sS}(G)$. On a

$$\alpha(G) - 1 \leq \alpha(G') \leq \alpha(G),$$

$$\gamma(G) - 1 \leq \gamma(G') \leq \gamma(G).$$

2.4.4 Insertion d'un sommet

L'insertion d'un sommet, notée T_{iS} , est la transformation réciproque de la suppression d'un sommet. Elle consiste à introduire dans un graphe un sommet qui doit être adjacent à un sous-ensemble de sommets du graphe.

Pour un graphe $G = (V, E)$ et pour un sommet $x \notin V$ qui doit être adjacent à l'ensemble de sommets $N_{G'}(x) \subseteq V$, le graphe résultant de la transformation spécifique d'insertion du sommet x avec l'ensemble de voisins $N_{G'}(x)$ dans G est noté $T_{iS}(G, x, N_{G'}(x))$.

```

Entrées : Un graphe  $G = (V, E)$ , un sommet  $x \notin V$  et un ensemble de voisins  $N_{G'}(x) \subseteq V$ 
Sortie : Un graphe  $G' = T_{iS}(G, x, N_{G'}(x))$ 
début
  |    $G \leftarrow eS(G, x)$ 
  |   pour chaque sommet  $v \in N_{G'}(x)$  faire
  |   |    $G \leftarrow aA(G, \{x, v\})$ 
fin

```

Algorithme 2.5 : Séquence des opérations de la transformation spécifique d'insertion du sommet x devant être adjacent à l'ensemble de voisins $N_{G'}(x)$ dans le graphe G

Proposition 20 Soit G un graphe et soit $G' \in T_{iS}(G)$. On a

$$\begin{aligned}\alpha(G) &\leq \alpha(G') \leq \alpha(G) + 1, \\ \gamma(G) &\leq \gamma(G') \leq \gamma(G) + 1.\end{aligned}$$

Remarque 1 Notons T'_{iS} le cas particulier de la transformation générique T_{iS} où l'ensemble de sommets devant être adjacent au sommet inséré est vide. On a alors

$$\alpha(G') = \alpha(G) + 1$$

pour tout graphe G et pour tout $G' \in T'_{iS}(G)$. T'_{iS} est donc une transformation exacte.

De même, notons T''_{iS} le cas particulier de la transformation générique T_{iS} où le sommet inséré doit être adjacent à tous les sommets du graphe. On a alors

$$\alpha(G') = \alpha(G)$$

pour tout graphe G et pour tout $G' \in T''_{iS}(G)$. T''_{iS} est également une transformation exacte.

2.4.5 Identification de deux sommets

L'identification de deux sommets, notée T_{idS} , consiste à insérer un nouveau sommet, à lier ce nouveau sommet à chaque voisin d'au moins un des deux sommets à identifier et à supprimer les deux sommets à identifier.

Pour un graphe $G = (V, E)$ et pour deux sommets $x \in V$ et $y \in V$ adjacents ou non, le graphe résultant de la transformation spécifique d'identification des sommets x

et y dans G est noté $T_{idS}(G, x, y)$.

```

Entrées : Un graphe  $G = (V, E)$  et deux sommets  $x \in V$  et  $y \in V$ 
Sortie : Un graphe  $G' = T_{idS}(G, x, y)$ 
début
   $G \leftarrow aS(G, z)$ 
  pour chaque sommet  $v \in N(x)$  faire
     $G \leftarrow eA(G, \{x, v\})$ 
    si  $v \notin N(y)$  alors
       $G \leftarrow aA(G, \{z, v\})$ 
   $G \leftarrow eS(G, x)$ 
  pour chaque sommet  $v \in N(y)$  faire
     $G \leftarrow eA(G, \{y, v\})$ 
     $G \leftarrow aA(G, \{z, v\})$ 
   $G \leftarrow eS(G, y)$ 
fin

```

Algorithme 2.6 : Séquence des opérations de la transformation spécifique d'identification des sommets x et y dans le graphe G

Proposition 21 Soit G un graphe et soit $G' \in T_{idS}(G)$. On a

$$\alpha(G) - 1 \leq \alpha(G') \leq \alpha(G), \quad (2.5)$$

$$\gamma(G) - 1 \leq \gamma(G') \leq \gamma(G) + 1.$$

DÉMONSTRATION de l'équation 2.5

Soit $G = (V, E)$ un graphe et $x \in V, y \in V$ deux sommets à identifier. Soit $G' = (V', E') = T_{idS}(G, x, y)$. Notons z le sommet de G' tel que $V' - \{z\} = V - \{x, y\}$.

D'une part, soit S' un ensemble stable dans G' . Si $z \notin S'$, alors S' est aussi un ensemble stable dans G . Si $z \in S'$, alors $S' - \{z\} \cup \{x\}$ est aussi un stable dans G . On en déduit que $\alpha(G') \leq \alpha(G)$.

D'autre part, soit S un ensemble stable dans G . Si $x \notin S$ et $y \notin S$, alors S est aussi un ensemble stable dans G' . Si $x \in S$ et $y \notin S$, alors $S - \{x\}$ est un stable dans G' et de la même façon si $y \in S$ et $x \notin S$, alors $S - \{y\}$ est un stable dans G' . Finalement, si $x \in S$ et $y \in S$, alors $S - \{x, y\} \cup \{z\}$ est un stable dans G' . On en déduit que $\alpha(G) - 1 \leq \alpha(G')$. ■

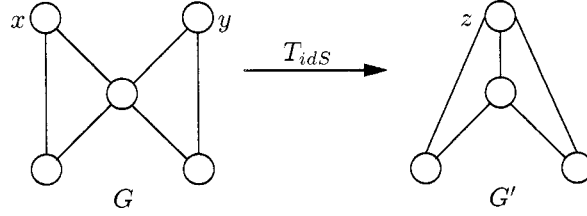


Figure 2.4 Transformation spécifique d'identification de deux sommets

Remarque 2 Notons T'_{idS} , le cas particulier de la transformation générique T_{idS} où les deux sommets à identifier doivent être non adjacents. Dans ce cas, la borne inférieure sur le nombre chromatique peut être améliorée puisque

$$\gamma(G) \leq \gamma(G')$$

pour tout graphe G et pour tout $G' \in T'_{idS}(G)$.

Plus précisément, soit $G = (V, E)$ un graphe et $x \in V$, $y \in V$ deux sommets tels que $\{x, y\} \notin E$. Soit $G' = (V', E') = T'_{idS}(G, x, y)$. Notons z le sommet de G' tel que $V' - \{z\} = V - \{x, y\}$. En considérant une coloration de G' , il est possible de déduire une coloration de G en affectant à x et y dans G la même couleur que celle affectée à z dans G' .

2.4.6 Partition du voisinage d'un sommet

La partition du voisinage d'un sommet, notée T_{pV} , consiste à éclater un sommet en autant de sommets que son voisinage non vide possède de composantes connexes puis à rendre chaque nouveau sommet adjacent à chaque sommet d'une seule de ces composantes connexes tel que chaque sommet d'une composante connexe est adjacent au même nouveau sommet.

Pour un graphe $G = (V, E)$ et pour un sommet $x \in V$ de voisinage non vide, le graphe résultant de la transformation spécifique de partition du voisinage du sommet x dans G est noté $T_{pV}(G, x)$. Pour obtenir ce graphe, il s'agit donc de considérer $N_1(x), N_2(x), \dots, N_k(x)$, les ensembles de sommets des k composantes connexes de $G[N(x)]$, puis pour chacun de ces ensembles, d'insérer un sommets x_i , $i = 1, 2, \dots, k$, tel que x_i sera adjacent à tous les sommets de $N_i(x)$, et finalement, de supprimer x .

Entrées : Un graphe $G = (V, E)$ et un sommet $x \in V$ de voisinage non vide
Sortie : Un graphe $G' = T_{pV}(G, x)$
début
 Soient $N_1(x), N_2(x), \dots, N_k(x)$ les ensembles de sommets des k composantes connexes de $G[N(x)]$
 pour chaque ensemble de sommets $N_i(x)$ **faire**
 $G \leftarrow aS(G, x_i)$
 pour chaque sommet $v \in N_i(x)$ **faire**
 $G \leftarrow aA(G, \{x_i, v\})$
 $G \leftarrow eA(G, \{x, v\})$
 $G \leftarrow eS(G, x)$
fin

Algorithme 2.7 : Séquence des opérations de la transformation spécifique de partition du voisinage non vide du sommet x dans G

Proposition 22 Soit G un graphe et soit $G' \in T_{pV}(G)$. On a

$$\alpha(G) \leq \alpha(G'),$$

$$\gamma(G) - 1 \leq \gamma(G') \leq \gamma(G).$$

DÉMONSTRATION

Voir (Gerber et Hertz, 2001).

Remarque 3 Notons T'_{pV} , le cas particulier de la transformation générique T_{pV} où le sommet à considérer est tel que le sous-graphe induit de son voisinage possède exactement deux composantes connexes. On observe que T'_{pV} est un cas particulier de la réciproque de T'_{ids} .

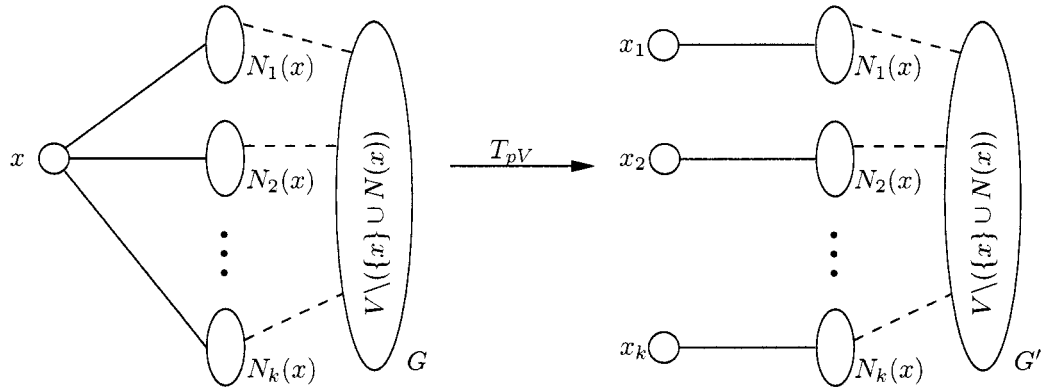


Figure 2.5 Transformation spécifique de partition du voisinage d'un sommet

Plus précisément, soit $G = (V, E)$ un graphe, soit $x \in V$ tel que $G[N(x)]$ possède deux composantes connexes et soient x_1 et x_2 les deux sommets ajoutés à G pour obtenir $G' = T'_{pV}(G, x)$. On a $G = T'_{idS}(G', x_1, x_2)$. On a vu que

$$\alpha(G') - 1 \leq \alpha(G),$$

d'où on déduit que

$$\alpha(G') \leq \alpha(G) + 1,$$

et on a donc une borne supérieure pour $\alpha(G')$ ce qui n'est pas le cas pour T_{pV} .

Le tableau 2.1 présente un résumé des transformations présentées dans cette section. On note que k et k' sont les constantes respectivement de la borne inférieure et de la borne supérieure de l'inéquation 2.1 (page 20).

Tableau 2.1 Transformations inexactes appliquées à un graphe $G = (V, E)$

T	Paramètres	k	k'	Nombre de sommets de $T(G, \bullet)$	Nombre d'arêtes de $T(G, \bullet)$
Suppression d'une arête	$\{x, y\} \in E$	0	1	$ V $	$ E - 1$
Insertion d'une arête	$\{x, y\} \notin E$	-1	0	$ V $	$ E + 1$
Suppression d'un sommet	$x \in V$	-1	0	$ V - 1$	$ E - N(x) $
Insertion d'un sommet	$x \notin V,$ $N(x) \subseteq V$	0	1	$ V + 1$	$ E + N(x) $
Identification de deux sommets	$x \in V,$ $y \in V$	-1	0	$ V - 1$	$ E - N(x) \cap N(y) $ si $\{x, y\} \notin E$ $ E - N(x) \cap N(y) - 1$ si $\{x, y\} \in E$
Partition du voisinage	$x \in V$	0	—	$ V + k$	$ E $

2.5 Exemples de transformations exactes

Une transformation T est exacte s'il existe une constante $k \in \mathbb{Z}$ telle que $\alpha(G') = \alpha(G) + k$ pour tout graphe G et pour tout graphe $G' \in T(G)$. Il y a deux types de transformations exactes: celles ne modifiant pas le nombre de stabilité ($k = 0$) et celles qui l'augmentent ou le diminuent ($k \neq 0$).

Dans cette section, quelques transformations exactes sont présentées. Plusieurs des transformations présentées ont été puisées dans un article non publié de Vadim V. Lozin (Lozin, 2003).

Une nouvelle notation pour décrire le voisinage des sommets, inspirée de celle de Vadim V. Lozin, est utilisée dans cette section. Soit U et W deux ensembles de sommets tels que $U \cap W = \emptyset$. On note $c_{U,W}$ l'ensemble des sommets n'appartenant

pas à $U \cup W$, voisins de tous les sommets de U et d'aucun sommet de W . Par exemple, soit $G = (V, E)$ un graphe et $x \in V$, $y \in V$, $z \in V$ trois sommets de G . On a que $c_{\{x\}, \{y, z\}} = N_G(x) - (N_G(x) \cap (N_G(y) \cup N_G(z)))$ est l'ensemble de sommets adjacents à x et non adjacents à y et à z .

Certaines figures représentant les transformations de cette section diffèrent légèrement de celles de la section précédente. Sauf indications contraires, il n'y a pas de contraintes au niveau de la présence d'arêtes entre les différents voisinages $c_{U, W}$. L'absence d'arêtes entre deux voisinages $c_{U, W}$ n'indique pas nécessairement qu'il ne peut y avoir d'arêtes. Il y a absence d'arêtes entre deux voisinages lorsqu'ils sont séparés par des pointillés.

2.5.1 Réduction simple

La réduction simple, notée T_{rS} , consiste à supprimer l'unique sommet voisin d'un sommet de degré un.

Pour un graphe $G = (V, E)$ et un sommet $x \in V$ de degré un, le graphe résultant de la transformation spécifique de suppression du voisin du sommet x de degré un dans G est noté $T_{rS}(G, x)$.

Comme les pointillés l'indiquent sur la figure 2.6, il y a au moins deux composantes connexes dans le graphe résultants d'un réduction simple.

Entrées : Un graphe $G = (V, E)$, un sommet $x \in V$ de degré un et le sommet $y \in N(x)$
Sortie : Un graphe $G' = T_{rS}(G, x)$
début
 pour chaque *sommet* $v \in N(y)$ **faire**
 $G \leftarrow eA(G, \{y, v\})$
 $G \leftarrow eS(G, y)$
fin

Algorithme 2.8 : Séquence des opérations de la transformation spécifique de réduction simple dans le graphe G

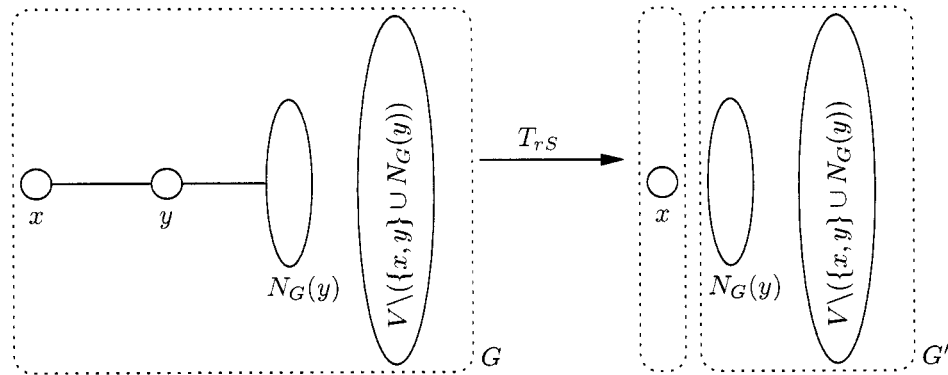


Figure 2.6 Transformation spécifique de réduction simple

Proposition 23 Soit G un graphe et soit $G' \in T_{rs}(G)$. On a

$$\alpha(G) = \alpha(G').$$

DÉMONSTRATION

Soit $G = (V, E)$ un graphe, soit $x \in V$ un sommet de degré un et soit $y \in V$ l'unique voisin de x . Pour chaque ensemble stable maximal S dans G , si $y \in S$, alors l'ensemble $S - \{y\} \cup \{x\}$ est également stable dans G . On déduit que $\alpha(T_{rs}(G, x)) \geq \alpha(G)$.

Comme $T_{rs}(G, x) = T_{rs}(G, y)$ et par la proposition 19, on a $\alpha(T_{rs}(G, x)) \leq \alpha(G)$. ■

Définition 24 Un **arbre** est un graphe connexe sans cycle.

Cette transformation est utile pour la classe des forêts, car elle permet de résoudre le problème du stable maximum. Une forêt est un graphe dont chaque composante connexe est un arbre.

Soit G un arbre. Construisons une séquence $\{G_i\}$ de graphes telle que $G_0 = G$ et $G_i \in T_{rS}(G_{i-1})$. Soit ℓ un indice tel que $T_{rS}(G_\ell) = \emptyset$. Le graphe G_ℓ est constitué de sommets isolés. Son nombre de stabilité est donc égal à son nombre de sommets. Comme $\alpha(G_i) = \alpha(G_{i-1})$ pour tout $i = 1, 2, \dots, \ell$, on déduit que $\alpha(G)$ est égal au nombre de sommets de G_ℓ .

2.5.2 Réduction simpliciale

Définition 25 Soit $G = (V, E)$ un graphe. Un sommet $x \in V$ dont le voisinage est une clique non vide est appelé **sommet simplicial**.

La réduction simpliciale, notée T_{rSi} , consiste à supprimer un sommet voisin d'un sommet simplicial.

Pour un graphe $G = (V, E)$, un sommet simplicial $x \in V$ et un sommet $y \in V$ voisin de x , le graphe résultant de la transformation spécifique de suppression du sommet y de G est noté $T_{rSi}(G, x, y)$.

Entrées : Un graphe $G = (V, E)$, un sommet simplicial $x \in V$ et un sommet $y \in N(x)$
Sortie : Un graphe $G' = T_{rSi}(G, x, y)$
début
 pour chaque sommet $v \in N(y)$ **faire**
 $G \leftarrow eA(G, \{y, v\})$
 $G \leftarrow eS(G, y)$
fin

Algorithme 2.9 : Séquence des opérations de la transformation spécifique de réduction simpliciale dans le graphe G

Remarque 4 Pour un graphe contenant un sommet de degré un, le voisinage de ce sommet ne contient qu'un sommet. Un sommet étant une clique, la réduction simpliciale est une généralisation de la réduction simple. Ainsi, $T_{rS}(G) \subset T_{rSi}(G)$ pour tout graphe G .

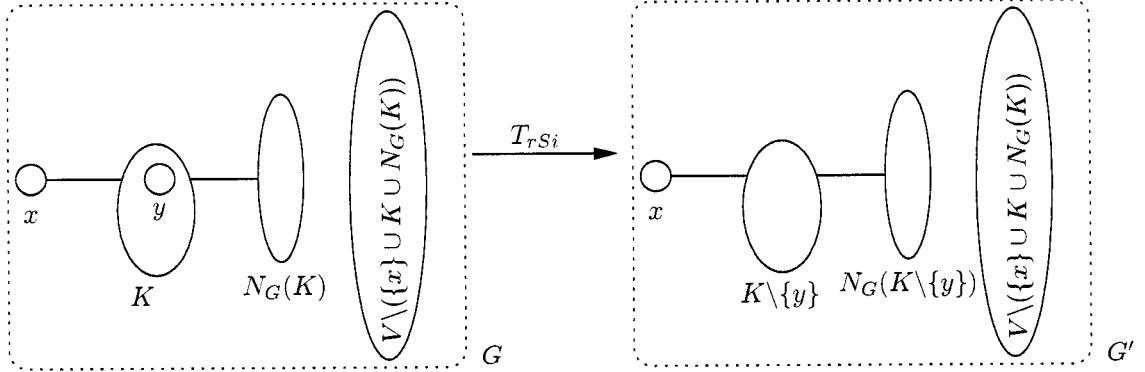


Figure 2.7 Transformation spécifique de réduction simpliciale

Proposition 26 Soit G un graphe et soit $G' \in T_{rSi}(G)$. On a

$$\alpha(G) = \alpha(G').$$

DÉMONSTRATION

Soit $G = (V, E)$ un graphe, soit $x \in V$ un sommet simplicial et soit $y \in N(x)$. Pour chaque ensemble stable $S \subset V$, si $y \in S$, alors l'ensemble $S - \{y\} \cup \{x\}$ est également stable dans G . On déduit que $\alpha(T_{rSi}(G, x, y)) \geq \alpha(G)$.

Comme $T_{rSi}(G, x, y) = T_{sS}(G, y)$ et par la proposition 19, on a

$$\alpha(T_{rSi}(G, x, y)) \leq \alpha(G).$$

■

Remarque 5 Une transformation exacte, nommée réduction de clique et notée T'_{rSi} , consistant à supprimer tous les voisins d'un sommet simplicial peut être déduite de la réduction simpliciale.

Soit $G = (V, E)$ un graphe et soit $x_K \in V$ un sommet simplicial tel que $N_G(x_K) = \{y_1, y_2, \dots, y_\ell\}$. Posons $G_0 = G$ et considérons $G_i = T'_{rSi}(G_{i-1}, x_K, y_i)$. Le graphe $G' = G_\ell$ sera tel que $d(x_K) = 0$ et $\alpha(G') = \alpha(G)$ puisque T_{rSi} est une transformation exacte. Ainsi, la transformation consistant à supprimer le voisinage d'un sommet simplicial est aussi une transformation exacte.

Définition 27 Un graphe est **triangulé** si tous ses cycles de plus de trois sommets contiennent au moins une corde.

Un résultat concernant les graphes triangulés permet d'élaborer une méthode de résolution du problème du stable dans cette classe de graphes à l'aide de la réduction simpliciale.

Théorème 28 (Dirac (Dirac, 1961)) Tout graphe triangulé G autre qu'une clique contient au moins deux sommets simpliciaux non adjacents.

Si G est un graphe triangulé, $G' \in T_{rSi}(G)$ est aussi triangulé, car tout sous-graphe induit d'un graphe triangulé est également triangulé.

Soit G un graphe triangulé. Construisons une séquence $\{G_i\}$ de graphes telle que $G_0 = G$ et $G_i \in T_{rSi}(G_{i-1})$. Soit ℓ un indice tel que $T_{rSi}(G_\ell) = \emptyset$. Le graphe G_ℓ est constitué de sommets isolés. Son nombre de stabilité est donc égal à son nombre de sommets. Comme $\alpha(G_i) = \alpha(G_{i-1})$ pour tout $i = 1, 2, \dots, \ell$, on déduit que $\alpha(G)$ est égal au nombre de sommets dans G_ℓ .

2.5.3 Suppression d'un sommet dominant

Définition 29 Soit $G = (V, E)$ un graphe. On dit que x est un **sommet dominant** s'il existe un sommet $y \in N(x)$ tel que $N(y) - \{x\} \subseteq N(x)$. On dit également

que x **domine** y .

La suppression d'un sommet dominant, notée T_{sSD} , consiste à supprimer un sommet dominant.

Pour un graphe $G = (V, E)$ et un sommet dominant $x \in V$, le graphe résultant de la transformation spécifique de suppression du sommet dominant x de G est noté $T_{sSD}(G, x)$.

Entrées : Un graphe $G = (V, E)$ et un sommet $x \in V$ dominant Sortie : Un graphe $G' = T_{sSD}(G, x)$ début pour chaque <i>sommet</i> $v \in N(x)$ faire $G \leftarrow eA(G, \{x, v\})$ $G \leftarrow eS(G, x)$ fin
--

Algorithme 2.10 : Séquence des opérations de la transformation spécifique de suppression du sommet dominant x du graphe G

Remarque 6 *Pour un graphe contenant deux sommets que nous noterons x et y tels que x domine y et $N(y)$ est une clique. Dans ce cas, y est un sommet simplicial. Ainsi, le graphe résultant de la suppression du sommet dominant x peut aussi être obtenu de la réduction simpliciale considérant y et consistant à supprimer le sommet x . Ainsi, $T_{rSi}(G) \subset T_{sSD}(G)$ pour tout graphe G .*

Proposition 30 Soit G un graphe et soit $G' \in T_{sSD}(G)$. On a

$$\alpha(G) = \alpha(G').$$

DÉMONSTRATION

Soit $G = (V, E)$ un graphe et $x \in V$ un sommet dominant. Soit $y \in V$ tel que x domine y . Pour chaque ensemble stable S dans G , si $x \in S$, alors l'ensemble $S - \{x\} \cup \{y\}$ est également stable dans G . On déduit que $\alpha(T_{sSD}(G, x)) \geq \alpha(G)$.

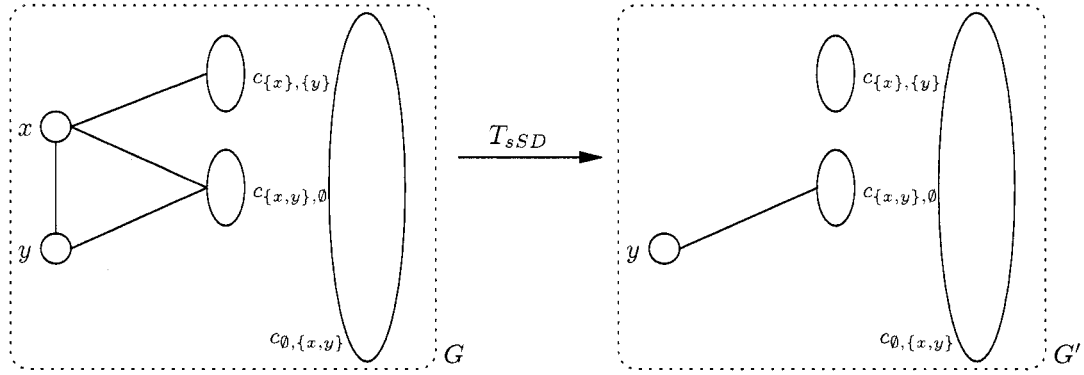


Figure 2.8 Transformation spécifique de suppression d'un sommet dominant

Comme $T_{sSD}(G, x) = T_{sS}(G, x)$ et par la proposition 19, on a

$$\alpha(T_{sSD}(G, x)) \leq \alpha(G).$$

■

2.5.4 Réduction d'un magnet

Définition 31 Soit un graphe $G = (V, E)$. Une arête $\{x, y\} \in E$ est un **magnet** si $\{u, v\} \in E$ pour tout $u \in c_{\{x\},\{y\}}$ et pour tout $v \in c_{\{y\},\{x\}}$.

La réduction d'un magnet, notée T_{rM} , consiste à supprimer un sommet incident à une arête formant un magnet et les arêtes entre le sommet non supprimé du magnet et ses voisins propres.

Pour un graphe $G = (V, E)$ et deux sommets $x \in V$ et $y \in V$ tels que $\{x, y\} \in E$ est un magnet, le graphe résultant de la transformation spécifique de suppression du sommet y de G et de suppression des arêtes entre le sommet x et le voisinage $c_{\{x\},\{y\}}$ est noté $T_{rM}(G, x, y)$.

Entrées : Un graphe $G = (V, E)$ et deux sommets $x \in V$ et $y \in V$ tels que $\{x, y\} \in E$ est un magnet Sortie : Un graphe $G' = T_{rM}(G, x, y)$ début pour chaque sommet $v \in N(y)$ faire $G \leftarrow eA(G, \{y, v\})$ $G \leftarrow eS(G, y)$ pour chaque sommet $v \in c_{\{x\}, \{y\}}$ faire $G \leftarrow eA(G, \{x, v\})$ fin

Algorithme 2.11 : Séquence des opérations de la transformation spécifique de la réduction du magnet $\{x, y\}$ dans le graphe G

Remarque 7 Soit $G = (V, E)$ un graphe et soient deux sommets $x \in V$ et $y \in V$ tels que $\{x, y\} \in E$ est un magnet et que $c_{\{x\}, \{y\}} = \emptyset$. On a $T_{rM}(G, x, y) = T_{sSD}(G, y)$. Le sommet y domine le sommet x dans G , car $N(x) = c_{\{x, y\}, \emptyset} \cup \{y\}$ et $N(y) = c_{\{y\}, \{x\}} \cup c_{\{x, y\}, \emptyset} \cup \{x\}$. Ainsi, $T_{sSD}(G) \subset T_{rM}(G)$ pour tout graphe G .

Proposition 32 Soit G un graphe et soit $G' \in T_{rM}(G)$. On a

$$\alpha(G) = \alpha(G').$$

DÉMONSTRATION

Soit un graphe $G = (V, E)$ et soient deux sommets $x \in V$ et $y \in V$ tels que $\{x, y\} \in E$ est un magnet. Soit S un ensemble stable dans G . Si $y \in S$, alors $(c_{\{y\}, \{x\}} \cup c_{\{x, y\}, \emptyset} \cup \{x\}) \cap S = \emptyset$. L'ensemble $S' = S - \{y\} \cup \{x\}$ est stable dans $T_{rM}(G, x, y)$. On déduit que $\alpha(T_{rM}(G, x, y)) \geq \alpha(G)$.

Soit S' un ensemble stable dans $G' = T_{rM}(G, x, y)$. Si $x \notin S'$ ou $c_{\{x\}, \{y\}} \cap S' = \emptyset$, alors S' est un ensemble stable dans G . Sinon, $c_{\{y\}, \{x\}} \cap S' = \emptyset$, car chaque sommet de $c_{\{x\}, \{y\}}$ est adjacent à chaque sommet de $c_{\{y\}, \{x\}}$ dans G' . L'ensemble $S' - \{x\} \cup \{y\}$ est stable dans G . On déduit que $\alpha(G) \geq \alpha(T_{rM}(G, x, y))$. ■

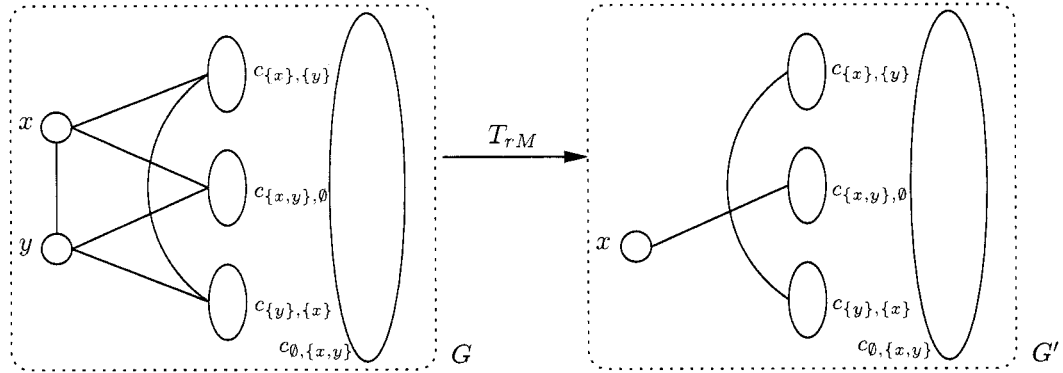


Figure 2.9 Transformation spécifique de réduction d'un magnet

2.5.5 Suppression d'un sommet

La suppression d'un sommet a été vue à la sous-section 2.4.3 comme étant une transformation inexacte. Cependant, en ajoutant des contraintes sur les voisinages des sommets, il est possible d'obtenir une transformation exacte.

Soit $G = (V, E)$ un graphe. On dit qu'un sommet $x \in V$ est spécial dans G s'il existe deux sommets $y \in V$ et $z \in V$ induisant un P_3 dans G avec les arêtes $\{x, y\} \in E$ et $\{x, z\} \in E$ et si le voisinage de ces trois sommets est tel que $c_{\{y\},\{x,z\}} \cup c_{\{z\},\{x,y\}} \cup c_{\{y,z\},\{x\}}$ induit une clique dans G .

La suppression d'un sommet spécial, notée T_{sSS} , consiste à supprimer un sommet spécial.

Pour un graphe $G = (V, E)$ et un sommet spécial $x \in V$, le graphe résultant de la transformation spécifique de suppression du sommet spécial x de G est noté $T_{sSS}(G, x)$.

Entrées : Un graphe $G = (V, E)$ et un sommet spécial $x \in V$
Sortie : Un graphe $G' = T_{sSS}(G, x)$
début
 pour chaque *sommet* $v \in N(x)$ **faire**
 $G \leftarrow eA(G, \{x, v\})$
 $G \leftarrow eS(G, x)$
fin

Algorithme 2.12 : Séquence des opérations de la transformation spécifique de suppression du sommet spécial x du graphe G

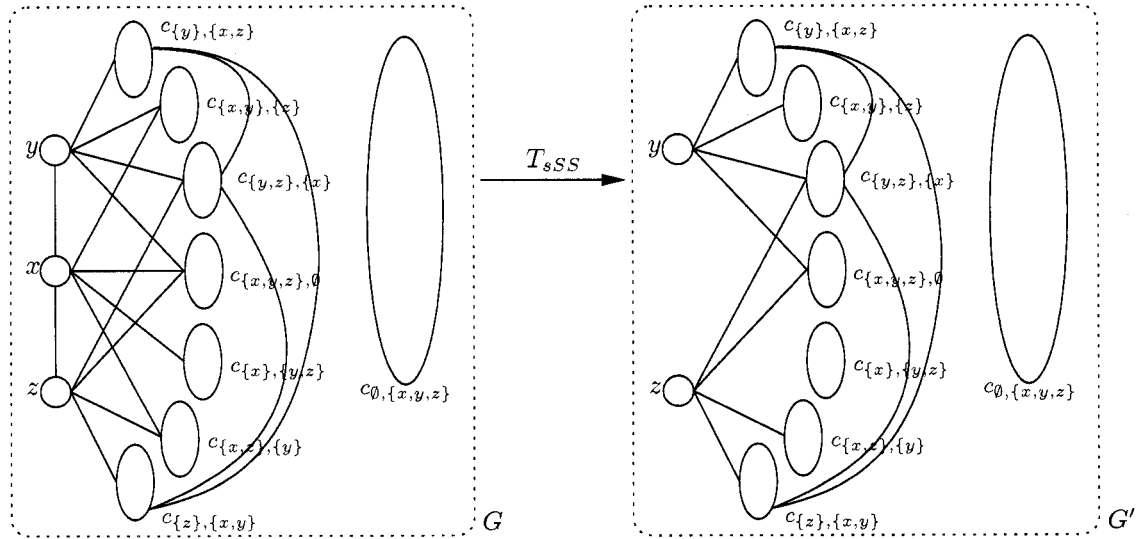


Figure 2.10 Transformation spécifique de suppression d'un sommet spécial

Proposition 33 Soit G un graphe et soit $G' \in T_{sSS}(G)$. On a

$$\alpha(G) = \alpha(G').$$

DÉMONSTRATION

Voir (Billionnet, 1981).

2.5.6 Suppression d'une arête

À la sous-section 2.4.1, il a été vu que la suppression d'une arête est une transformation inexacte. Néanmoins, il est possible d'effectuer cette transformation de sorte qu'elle ne modifie pas le nombre de stabilité en ajoutant des contraintes sur le voisinage des sommets incidents à l'arête à supprimer.

Soit $G = (V, E)$, un graphe. On dit qu'une arête $\{x, y\} \in E$ est spéciale s'il existe un sommet $z \in V$ induisant un P_3 dans G avec les arêtes $\{x, y\} \in E$ et $\{x, z\} \in E$ et si le voisinage de ces trois sommets est tel que $c_{\{z\}, \{x, y\}} = \emptyset$.

La suppression d'une arête spéciale, notée T_{sAS} , consiste à supprimer une arête spéciale.

Pour un graphe $G = (V, E)$ et une arête spéciale $\{x, y\}$, le graphe résultant de la transformation spécifique de suppression de l'arête spéciale $\{x, y\}$ de G est noté $T_{sAS}(G, \{x, y\})$.

Entrées : Un graphe $G = (V, E)$ et une arête spéciale $\{x, y\} \in E$
Sortie : Un graphe $G' = T_{sAS}(G, \{x, y\})$
début
 | $G \leftarrow eA(G, \{x, y\})$
fin

Algorithme 2.13 : Séquence des opérations de la transformation de suppression de l'arête spéciale $\{x, y\}$ du graphe G

Proposition 34 Soit $G = (V, E)$ un graphe et soit $G' \in T_{sAS}(G)$. On a

$$\alpha(G) = \alpha(G').$$

DÉMONSTRATION

Voir (Butz et al., 1982).

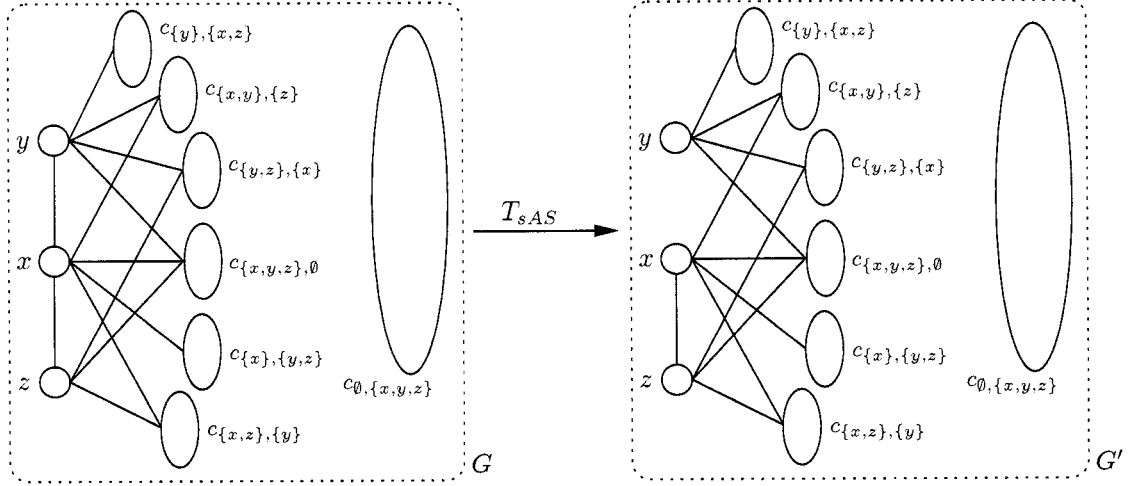


Figure 2.11 Transformation spécifique de suppression d'une arête spéciale

2.5.7 Insertion d'une arête

Il a été vu à la sous-section 2.4.2 que l'insertion d'une arête est une transformation inexacte. Cependant, l'insertion d'une arête peut se faire de manière exacte, en ajoutant des contraintes sur le voisinage des sommets incidents à l'arête à insérer.

Soit $G = (V, E)$ un graphe. On dit qu'un triplet de sommets ordonné (x, y, z) est spécial s'il est tel que $\{x, z\} \in E$, $\{x, y\} \notin E$ et $\{y, z\} \notin E$ et si le voisinage des sommets du triplet est tel que $\{u, v\} \in E$ pour tout $u \in c_{\{z\},\{x,y\}}$ et pour tout $v \in c_{\{x\},\{y,z\}} \cup c_{\{x,y\},\{z\}} \cup c_{\{y,z\},\{x\}}$.

L'insertion d'une arête lorsqu'il y a un triplet de sommets ordonné spécial, notée T_{iAtS} , consiste à insérer une arête entre le premier et le deuxième sommet du triplet, à insérer une arête entre le premier sommet du triplet et chaque sommet du voisinage propre du troisième sommet et à supprimer chaque arête entre le

troisième sommet du triplet et les sommets de son voisinage propre.

Pour un graphe G et un triplet de sommets ordonné spécial (x, y, z) de G , le graphe résultant de la transformation spécifique d'insertion de l'arête $\{x, y\}$ dans le graphe G est noté $T_{iAtS}(G, (x, y, z))$.

Entrées : Un graphe $G = (V, E)$ et un triplet de sommets ordonné spécial (x, y, z) Sortie : Un graphe $G' = T_{iAtS}(G, (x, y, z))$ début $G \leftarrow aA(G, \{x, y\})$ pour chaque <i>sommet</i> $v \in c_{\{z\}, \{x, y\}}$ faire $G \leftarrow eA(G, \{z, v\})$ $G \leftarrow aA(G, \{x, v\})$ fin
--

Algorithme 2.14 : Séquence des opérations de la transformation spécifique d'insertion de l'arête $\{x, y\}$ dans le graphe G lorsqu'il y a un triplet de sommets ordonné spécial (x, y, z)

Proposition 35 Soit G un graphe et soit $G' \in T_{iAtS}(G)$. On a

$$\alpha(G) = \alpha(G').$$

DÉMONSTRATION

Soit $G = (V, E)$ un graphe, soit (x, y, z) un triplet de sommets ordonné spécial et soit $G' = (V', E') = T_{iAtS}(G, (x, y, z))$.

D'une part, soit $S \subseteq V$ un stable dans G .

Si $x \notin S$ ou si $y \notin S$ et $c_{\{z\}, \{x, y\}} \cap S = \emptyset$, alors S est un ensemble stable dans G' .

Si $x \in S$, $y \in S$ et $c_{\{z\}, \{x, y\}} \cap S = \emptyset$, alors $(N_G(x) \cup N_G(y)) \cap S = \emptyset$. Comme $N_{G'}(z) \subseteq N_G(x) \cup N_G(y)$, on a que l'ensemble $S - \{x\} \cup \{z\}$ est stable dans G' .

Si $x \in S$ et $c_{\{z\}, \{x, y\}} \cap S \neq \emptyset$, alors il existe au moins un $w \in c_{\{z\}, \{x, y\}} \cap S'$. Puisque chaque sommet de $c_{\{z\}, \{x, y\}}$ est adjacent à chaque sommet de $c_{\{y, z\}, \{x\}}$, on a $c_{\{y, z\}, \{x\}} \cap S' = \emptyset$. Ainsi, l'ensemble $S - \{x\} \cup \{z\}$ est stable dans G' .

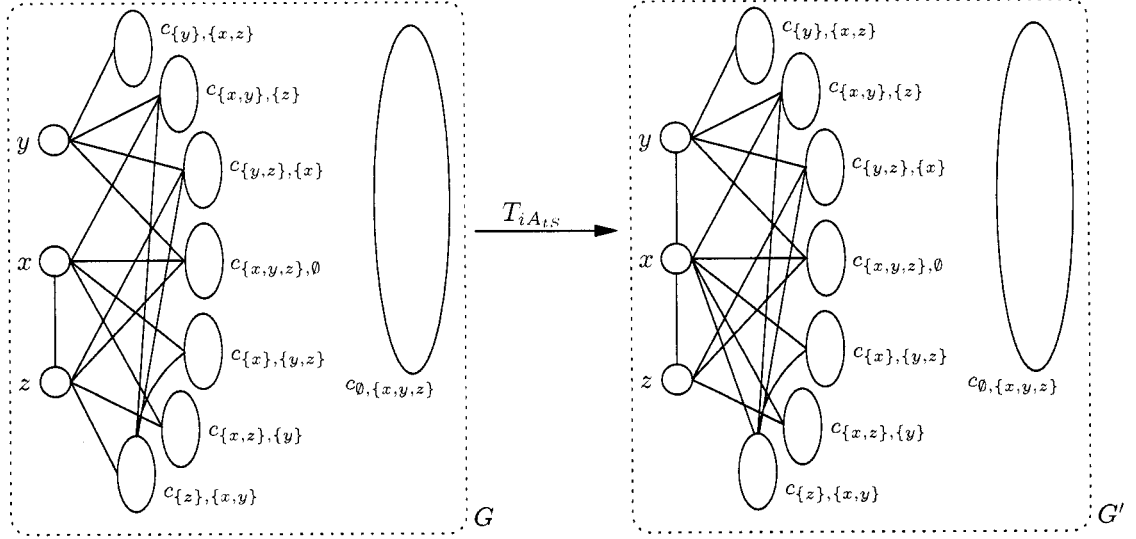


Figure 2.12 Transformation spécifique d'insertion d'une arête lorsqu'il y a un triplet de sommets ordonné spécial

On en déduit que $\alpha(G') \geq \alpha(G)$.

D'autre part, soit $S' \subseteq V'$ un stable dans G' .

Si $z \notin S'$ ou $N_G(z) \cap S' = \emptyset$, alors S' est un stable dans G .

Si $z \in S'$ et $N_G(z) \cap S' \neq \emptyset$, alors il existe au moins un $w \in c_{\{z\},\{x,y\}} \cap S'$. Puisque chaque sommet de $c_{\{z\},\{x,y\}}$ est adjacent à chaque sommet de $c_{\{x\},\{y,z\}}$, on a $c_{\{x\},\{y,z\}} \cap S' = \emptyset$ et puisque chaque sommet de $c_{\{z\},\{x,y\}}$ est adjacent à chaque sommet de $c_{\{x,y\},\{z\}}$, on a $c_{\{x,y\},\{z\}} \cap S' = \emptyset$. Ainsi, l'ensemble $S' - \{z\} \cup \{x\}$ est stable dans G .

On en déduit que $\alpha(G) \geq \alpha(G')$. ■

Remarque 8 Soit (x, y, z) un triplet de sommets ordonné spécial dans un graphe $G = (V, E)$. Si $c_{\{z\},\{x,y\}} = \emptyset$, on dit que (x, y, z) est un bon triplet de sommets.

L'insertion d'une arête lorsqu'il y a un bon triplet de sommets, notée $T'_{iA_{tS}}$, consiste à ajouter une arête entre le premier et le deuxième sommet du bon triplet.

Pour un graphe G et un bon triplet de sommets (x, y, z) de G , le graphe résultant de l'insertion de l'arête $\{x, y\}$ dans G est noté $T'_{iA_{tS}}(G, (x, y, z))$.

On remarque que $T'_{iA_{tS}}(G) \subseteq T_{iA_{tS}}(G)$ pour tout graphe G et que $T'_{iA_{tS}}$ est la transformation réciproque de T_{sAS} .

2.5.8 Réduction d'un BAT

Définition 36 Soit $G = (V, E)$, un graphe. Un triplet de sommets ordonné (x, y, z) dans G est un **BAT** si

- x, y et z induisent un P_3 dans G dont les arêtes sont $\{x, y\}$ et $\{x, z\}$,
- chaque sommet de $c_{\{x\},\{y,z\}}$ est adjacent à chaque sommet d'au moins un des deux ensembles $c_{\{y\},\{x,y\}} \cup c_{\{x,y\},\{z\}} \cup c_{\{y,z\},\{x\}}$ ou $c_{\{z\},\{x,y\}} \cup c_{\{x,z\},\{y\}} \cup c_{\{y,z\},\{x\}}$,
- chaque sommet de $c_{\{x,z\},\{y\}}$ est adjacent à chaque sommet dans $c_{\{y\},\{x,z\}} \cup c_{\{x,y\},\{z\}} \cup c_{\{y,z\},\{x\}}$,
- chaque sommet de $c_{\{x,y\},\{z\}}$ est adjacent à chaque sommet dans $c_{\{z\},\{x,y\}} \cup c_{\{x,z\},\{y\}} \cup c_{\{y,z\},\{x\}}$.

La réduction de BAT, notée T_{rBAT} , consiste à supprimer les sommets d'un BAT, à insérer deux sommets non adjacents dont un est adjacent à chaque sommet voisin d'au moins un sommet du BAT supprimé et l'autre est adjacent à chaque voisin commun aux trois sommets du BAT supprimé.

Pour un graphe $G = (V, E)$ et trois sommets $x \in V$, $y \in V$ et $z \in V$ tels que (x, y, z) est un BAT, le graphe résultant de la transformation spécifique de réduction de ce BAT est noté $T_{rBAT}(G, (x, y, z))$.

```

Entrées : Un graphe  $G = (V, E)$  et un BAT  $(x, y, z)$  dans  $G$ 
Sortie : Un graphe  $G' = T_{rBAT}(G, (x, y, z))$ 
début
  pour chaque sommet  $v \in N(x)$  faire
     $G \leftarrow eA(G, \{x, v\})$ 
   $G \leftarrow eS(G, x)$ 
  pour chaque sommet  $v \in N(y)$  faire
     $G \leftarrow eA(G, \{y, v\})$ 
   $G \leftarrow eS(G, y)$ 
  pour chaque sommet  $v \in N(z)$  faire
     $G \leftarrow eA(G, \{z, v\})$ 
   $G \leftarrow eS(G, z)$ 
   $G \leftarrow aS(G, x_1)$ 
   $G \leftarrow aS(G, x_2)$ 
  pour chaque sommet  $v \in N(x) \cup N(y) \cup N(z)$  faire
     $G \leftarrow aA(G, \{x_2, v\})$ 
  pour chaque sommet  $v \in c_{\{x, y, z\}, \emptyset}$  faire
     $G \leftarrow aA(G, \{x_1, v\})$ 
fin

```

Algorithme 2.15 : Séquence des opérations de la transformation spécifique de la réduction du BAT (x, y, z) dans le graphe G

Proposition 37 Soit G un graphe et soit $G' \in T_{rBAT}(G)$. On a

$$\alpha(G) = \alpha(G').$$

DÉMONSTRATION

Voir (Hertz, 1997).

Remarque 9 La réduction d'un BAT tel que son premier sommet est de degré deux, notée T'_{rBAT} , consiste à supprimer toutes les arêtes incidentes à au premier sommet d'un BAT et à identifier les deux derniers sommets de ce BAT. T'_{rBAT} est également une transformation exacte.

Soit $G = (V, E)$ un graphe et soit $x \in V$ un sommet tel que $d(x) = 2$. On a que (x, y, z) est un BAT dans G où $N_G(x) = \{y, z\}$. La transformation spécifique

qui consiste à effectuer la réduction d'un tel BAT consiste à identifier y et z et à supprimer toutes les arêtes incidentes à x .

2.5.9 Struction inversée

La struction inversée, notée T_{SI} , consiste à choisir deux sommets adjacents puis à ajouter deux nouveaux sommets adjacents. Ensuite, un des deux nouveaux sommets est lié aux deux sommets choisis et l'autre à l'ensemble des voisins communs des deux sommets choisis. Finalement, les arêtes entre les deux sommets choisis et l'ensemble de leurs voisins communs sont supprimées.

Pour un graphe G et une arête $\{x, y\} \in E$, le graphe résultant de la transformation spécifique de struction inversée visant l'arête $\{x, y\}$ dans G est noté $T_{SI}(G, \{x, y\})$.

```

Entrées : Un graphe  $G = (V, E)$  et une arête  $\{x, y\}$ 
Sortie : Un graphe  $G' = T_{SI}(G, \{x, y\})$ 
début
   $G \leftarrow aS(G, v_1)$ 
   $G \leftarrow aS(G, v_2)$ 
  pour chaque sommet  $v \in \{v_1, x, y\}$  faire
     $G \leftarrow aA(G, \{v_2, v\})$ 
  pour chaque sommet  $v \in c_{\{x, y\}, \emptyset}$  faire
     $G \leftarrow eA(G, \{x, v\})$ 
     $G \leftarrow eA(G, \{y, v\})$ 
     $G \leftarrow aA(G, \{v_1, v\})$ 
fin

```

Algorithme 2.16 : Séquence des opérations de la transformation spécifique de struction inversée visant l'arête $\{x, y\}$ dans le graphe G

Proposition 38 Soit $G = (V, E)$ un graphe et soit $G' \in T_{SI}(G)$. On a

$$\alpha(G) = \alpha(G') - 1.$$

La preuve de la proposition 38 découle des observations qui suivent.

Il existe une transformation appelée struction (Ebenegger, 1984) que l'on note T_S .

Pour un graphe $G = (V, E)$ et un sommet $x \in V$, le graphe résultant de la transformation spécifique de struction est noté $T_S(G, x)$. Considérons une numérotation des sommets telle que $v_0 = x$, $N(x) = \{v_1, v_2, \dots, v_p\}$ et $V - N(x) = \{v_{p+1}, \dots, v_{|V|-1}\}$. $T_S(G, x)$ est obtenu de

- la suppression des sommets $v_0, v_1, v_2, \dots, v_p$,
- de l'ajout du sommet $v_{i,j}$ si $\{v_i, v_j\} \notin E$ pour $1 \leq i < j \leq p$,
- de l'ajout de l'arête $\{v_{i,j}, v_{k,l}\}$ si $i \neq k$ ou $\{v_j, v_l\} \in E$ pour $1 \leq i < j \leq p$ et $1 \leq k < l \leq p$, et
- de l'ajout de l'arête $\{v_{i,j}, v_k\}$ si $\{v_i, v_k\} \in E$ ou $\{v_j, v_k\} \in E$ pour $1 \leq i < j \leq p$ et $p+1 \leq k \leq |V|-1$.

Soit G un graphe et $G' \in T_S(G)$. Il a été prouvé dans (Ebenegger, 1984) que la struction (STability number RedUCTION) est telle que

$$\alpha(G') = \alpha(G) - 1.$$

La struction inversée est la transformation réciproque du cas particulier de la struction où le sommet visé est tel que le sous-graphe induit de son voisinage comprend exactement trois sommets et une seule arête. Par conséquent, si $G' \in T_{SI}(G)$, alors $G \in T_S(G')$.

Plus précisément, soit $G = (V, E)$ un graphe et $x \in V$ et $y \in V$ deux sommets. Soit v_2 le sommet ajouté à G et voisin de x et y dans $G' = T_{SI}(G, \{x, y\})$. On a $G = T_S(G', v_2)$.

T	Paramètres	k	Nombre de sommets de $T(G, \bullet)$	Nombre d'arêtes de $T(G, \bullet)$
Réduction simple	$x \in V$ tel que $N(x) = \{y\}$	0	$ V - 1$	$ E - N(y) $
Réduction simpliciale	$x \in V$ simplicial, $y \in N(x)$	0	$ V - 1$	$ E - N(y) $
Suppression d'un sommets dominant	$x \in V$ dominant	0	$ V - 1$	$ E - N(x) $
Réduction d'un magnet	$x \in V, y \in V$ tels que $\{x, y\}$ est un magnet	0	$ V - 1$	$ E - c_{\{x\}, \{y\}} - N(y) $
Suppression d'un sommets	$x \in V$ spécial	0	$ V - 1$	$ E - N(x) $
Suppression d'une arête	$\{x, y\} \in E$ spéciale	0	$ V $	$ E - 1$
Insertion d'une arête	(x, y, z) un triplet de sommets ordonné spécial dans G	0	$ V $	$ E + 1$
Réduction d'un BAT	(x, y, z) un BAT dans G	0	$ V $	$ E - c_{\{x, y\}, \{z\}} -$ $ c_{\{x, z\}, \{y\}} -$ $ c_{\{y, z\}, \{x\}} -$ $ c_{\{x, y, z\}, \emptyset} - 2$
Struction inversée	$\{x, y\} \in E$	1	$ V + 2$	$ E - c_{\{x, y\}, \emptyset} + 3$

Tableau 2.2 Transformations exactes appliquées à un graphe quelconque $G = (V, E)$

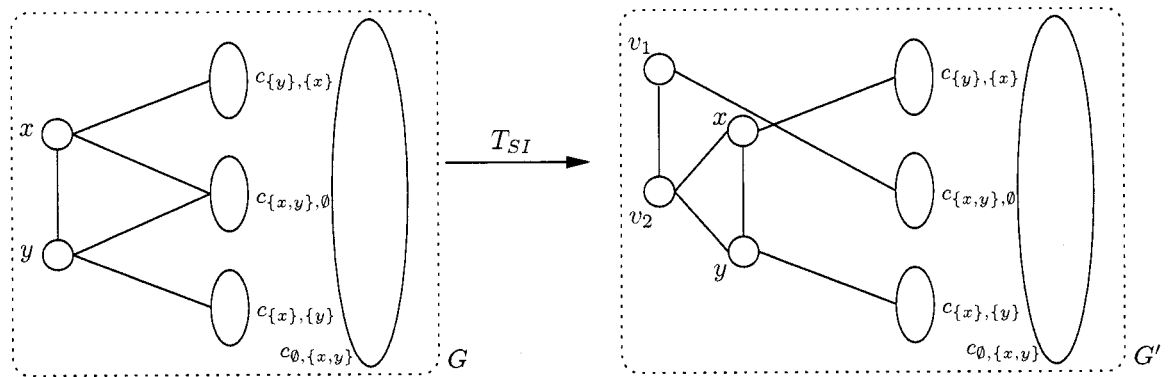


Figure 2.13 Struction inversée

CHAPITRE 3

INCIDENCE DE TRANSFORMATIONS DE GRAPHS

Dans le chapitre précédent, nous avons défini la notion de transformation et en particulier celle de transformation exacte. Tel que mentionné à la section 2.3, le but poursuivi par l'utilisation de transformations avant la résolution du problème du stable maximum est de réduire le coût de résolution de ce problème.

Ce chapitre présente les résultats expérimentaux provenant de l'application d'une méthode permettant de déterminer s'il est avantageux d'employer la stratégie d'utilisation des transformations exactes présentées à la section 2.3.1.

En premier lieu, la méthode élaborée est décrite et l'algorithme de résolution du problème du stable maximum ainsi que l'échantillon de graphes utilisé sont présentés. Ensuite, la présentation des résultats provenant de la méthode élaborée se fait en trois étapes. Tout d'abord, les résultats relatifs au coût de résolution du problème du stable maximum sont présentés. Suivent les résultats par rapport à l'incidence sur la structure des graphes de trois différentes transformations utilisées dans des séquences qui sont décrites avant la présentation des résultats. La troisième série de résultats permet d'évaluer l'incidence réelle des transformations sur le coût de résolution du problème du stable maximum et de déterminer s'il était possible de prédire leur incidence. De plus, quelques résultats sur des graphes de référence sont présentés. Les résultats sont discutés dans la dernière section.

3.1 Définitions et concepts

Les résultats se rapportant à la première transformation expérimentée semblaient montrer que l'utilisation de cette dernière augmentait le coût de résolution. Il nous semblait donc approprié de tenter de déterminer pour quels graphes et en utilisant quelle transformation l'objectif de réduction du coût de résolution a des chances d'être atteint. À cet effet, une méthode a été développée.

Cette section débute par l'explication de la démarche expérimentale élaborée. Afin de bien comprendre cette dernière, un exemple de son application est présenté. Puis, le choix de DFMAX comme algorithme de résolution du problème du stable maximum est expliqué. L'échantillon de graphes et son utilisation dans les expérimentations sont présentés. Finalement, le module informatique développé pour l'expérimentation est brièvement décrit.

3.1.1 Démarche expérimentale

Soit \mathcal{A} un algorithme de résolution du problème du stable maximum et $G = (V, E)$ un graphe de $n(G)$ sommets et de densité $d(G)$. Rappelons que la densité d'un graphe est le rapport entre le nombre d'arêtes de ce graphe et le nombre d'arêtes du graphe complet contenant un nombre de sommets identique à ce graphe. Nous appellerons structure du graphe G le couple $(n(G), d(G))$.

Le coût de résolution du problème du stable maximum dans G à l'aide de \mathcal{A} est noté $C_{\mathcal{A}}(G)$. On note $c_{\mathcal{A}}(n, d)$ le coût moyen pour la résolution de ce problème à l'aide de l'algorithme \mathcal{A} dans les graphes de n sommets et de densité d .

Notons que le coût d'un algorithme de résolution du problème du stable maximum peut être mesuré par le temps d'exécution de l'algorithme, par le nombre de retours

arrières dans le cas, par exemple, d'un algorithme implémentant une méthode de séparation et d'évaluation progressive, ou par le nombre d'itérations.

Considérons G un graphe de $n(G)$ sommets et de densité $d(G)$, \mathcal{T} une séquence de transformations et G' le graphe de $n(G')$ sommets et de densité $d(G')$ résultant de l'application de \mathcal{T} à G .

Notons que \mathcal{T} est une séquence de transformations ce qui implique qu'une suite de transformations est appliquée à G pour l'obtention de G' . La notion de séquence de transformations et les différentes séquences utilisées sont présentées à la section 3.3. Une séquence de transformations sera notée par des lettres majuscules représentant le nom de la transformation qu'elle utilise.

Le coût de résolution du problème du stable maximum dans le graphe initial est estimé par $c_1 = c_{\mathcal{A}}(n(G), d(G))$ et celui du graphe résultant de l'application de \mathcal{T} par $c_2 = c_{\mathcal{A}}(n(G'), d(G'))$. Le gain escompté sera évalué par le rapport $r_{\mathcal{A}} = \frac{c_2}{c_1}$. Si $r_{\mathcal{A}}$ est supérieur ou égal à 1, on a une perte escomptée.

Finalement, le coût réel de résolution du problème du stable maximum est mesuré en appliquant l'algorithme \mathcal{A} à G et à G' d'où on obtient respectivement $c'_1 = C_{\mathcal{A}}(G)$ et $c'_2 = C_{\mathcal{A}}(G')$. Le gain obtenu est évalué par le rapport $r'_{\mathcal{A}} = \frac{c'_2}{c'_1}$. Si $r'_{\mathcal{A}}$ est supérieur ou égal à 1, on a une perte réelle.

Notons que la génération aléatoire d'un graphe consiste en l'utilisation d'un générateur aléatoire $\mathcal{G}_{n,p}$ tel que n fixe le nombre de sommets du graphe à générer et p représente la probabilité de chaque arête d'appartenir à ce même graphe. On considère qu'un graphe ainsi produit a n sommets et une densité de p .

Illustrons les concepts précédemment décrits à l'aide de l'exemple suivant.

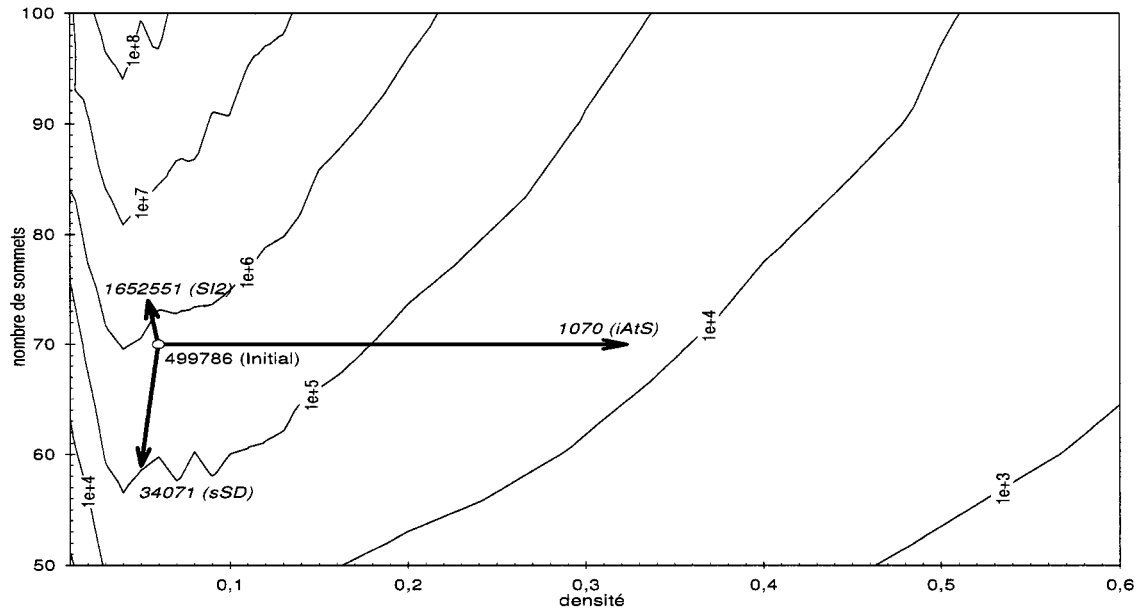


Figure 3.1 Courbes de niveau du coût de résolution du problème du stable maximum en fonction du nombre de sommets et de la densité des graphes initiaux de 50 à 100 sommets

Exemple 6 Considérons un graphe G généré de façon aléatoire tel que $n(G) = 70$ et $d(G) = 0,06$. Considérons également G'_1 le graphe résultant de l'application de la séquence de transformations SSD, décrite à la sous-section 3.3.2, à G , G'_2 le graphe résultant de l'application de la séquence de transformations iATS, décrite à la sous-section 3.3.3, à G et G'_3 le graphe résultant de l'application de la séquence de transformations SI2, décrite à la sous-section 3.3.1, à G .

Le graphique de la figure 3.1 représente en abscisse la densité et en ordonnée le nombre de sommets. Les courbes de niveau représentent les valeurs des estimateurs du coût de résolution $c_A(n, d)$ en fonction du nombre de sommets et de la densité. Le coût de résolution a été mesuré par le nombre de retours arrières de DFMAX (page 12).

Tableau 3.1 Exemple de résultats expérimentaux

Structure de G (n, d)		Estimateur $c_A(n, d)$	Coût réel $C_A(G)$		
(70, 0, 06)		518 004	499 786		

G' résultant de l'application de	Structure de G' ($n(G'), d(G')$)	Estimateur $c_A(n(G'), d(G'))$	Coût réel $C_A(G')$	Gain estimé r_A	Gain réel r'_A
SSD	(59, 0, 05)	123 256	34 071	0, 24	0, 07
IATS	(70, 0, 32)	16 950	1070	0, 03	0, 00
SI2	(74, 0, 05)	2 901 810	1 652 551	5, 60	3, 31

Le point correspondant à la coordonnée (70, 0, 06) indique la valeur estimée du coût de résolution du problème du stable maximum dans G . La valeur réelle du coût de résolution est de 499 786.

La coordonnée correspondant à l'extrémité des flèches indique la valeur estimée du coût de résolution en fonction de la structure des graphes G'_1 , G'_2 et G'_3 . Plus précisément, la coordonnée (59, 0, 05) indique la valeur estimée du coût de résolution dans G'_1 , celle à la coordonnée (70, 0, 32) indique la valeur estimée du coût de résolution dans G'_2 et celle à la coordonnée (74, 0, 05) indique la valeur estimée du coût de résolution dans G'_3 . La valeur réelle du coût de résolution pour ces trois graphes est respectivement de 34 071, 1070 et 1 652 551.

Le tableau 3.1 présente un résumé des valeurs obtenues.

Notons que pour SSD et IATS, le gain réel r'_A est supérieur au gain escompté r_A tandis que pour SI2, il y a une perte réelle inférieure à celle escomptée. De plus, on remarque que lorsqu'un gain est escompté, il y a réellement un gain et lorsqu'une perte est escomptée, il y a réellement une perte.

L'exemple 6 montre un cas particulier de la méthode élaborée pour atteindre les objectifs précédemment mentionnés.

La première étape de la méthode n'est pas illustrée dans l'exemple 6. Elle consiste en la construction d'un ensemble d'estimateurs du coût de résolution du problème du stable maximum $c_{\mathcal{A}}(n, d)$ à l'aide de l'algorithme \mathcal{A} . Expérimentalement, $c_{\mathcal{A}}(n, d)$ est déterminé par la moyenne du coût de résolution du problème dans des graphes de n sommets et de densité d . Considérons k graphes G_1, G_2, \dots, G_k de n sommets et de densité d . On a

$$c_{\mathcal{A}}(n, d) \approx \frac{\sum_{i=1}^k C_{\mathcal{A}}(G_i)}{k}.$$

Les étapes suivantes visent à déterminer l'incidence d'une séquence de transformations \mathcal{T} sur la structure d'un graphe G et sur la résolution du problème du stable maximum à l'aide de l'algorithme \mathcal{A} de même qu'à vérifier si l'utilisation de l'ensemble d'estimateurs du coût de résolution permet d'évaluer ces impacts. Le graphe résultant de l'application de \mathcal{T} à G est noté G' .

Tout d'abord, il faut appliquer les séquences de transformations à chacun des graphes de l'échantillon. Rappelons que la structure d'un graphe G' résultant de l'application d'une séquence de transformations est notée $(n(G'), d(G'))$.

Ensuite, le coût de résolution pour G et G' est estimé puis, le coût réel de résolution pour G et G' est calculé. Ces étapes incluent également le calcul du gain escompté et du gain réel.

Lorsqu'aucun estimateur du coût de résolution n'a été déterminé expérimentalement pour une structure $(n(G), d(G))$, une approximation linéaire est effectuée.

Considérons N et D respectivement l'ensemble des nombre de sommets et l'ensemble des densités pour lesquels un estimateur a été calculé. Les estimateurs $c_{\mathcal{A}}(n_1, d_1)$,

$c_{\mathcal{A}}(n_2, d_2)$, $c_{\mathcal{A}}(n_1, d_2)$ et $c_{\mathcal{A}}(n_2, d_1)$ utilisés sont tels que

$$\begin{aligned}(n_1, d_1) &= (\max\{n \in N \mid n \leq n(G)\}, \max\{d \in D \mid d \leq d(G)\}) \text{ et} \\ (n_2, d_2) &= (\min\{n \in N \mid n \geq n(G)\}, \min\{d \in D \mid d \geq d(G)\}).\end{aligned}$$

L'approximation linéaire consiste, si n_1 est différent de n_2 , tout d'abord à calculer

$$\begin{aligned}c_{\mathcal{A}}(n_1, d(G)) &= c_{\mathcal{A}}(n_1, d_1) + (d(G) - d_1) \left(\frac{c_{\mathcal{A}}(n_1, d_2) - c_{\mathcal{A}}(n_1, d_1)}{d_2 - d_1} \right) \text{ et} \\ c_{\mathcal{A}}(n_2, d(G)) &= c_{\mathcal{A}}(n_2, d_1) + (d(G) - d_1) \left(\frac{c_{\mathcal{A}}(n_2, d_2) - c_{\mathcal{A}}(n_2, d_1)}{d_2 - d_1} \right)\end{aligned}$$

puis à calculer l'estimateur recherché :

$$c_{\mathcal{A}}(n(G), d(G)) = c_{\mathcal{A}}(n_1, d(G)) + (n(G) - n_1) \left(\frac{c_{\mathcal{A}}(n_2, d(G)) - c_{\mathcal{A}}(n_1, d(G))}{n_2 - n_1} \right).$$

Si n_1 est égal à n_2 , alors $c_{\mathcal{A}}(n_1, d(G))$ tel que calculé ci-haut est l'estimateur recherché.

3.1.2 Algorithme de résolution

Les expérimentations effectuées visent à étudier l'impact de séquences de transformations sur la structure des graphes et à comparer le coût de résolution du problème du stable maximum dans un graphe initial avec le coût de résolution dans un graphe résultant d'une séquence de transformations. Ainsi, l'algorithme de résolution de ce problème n'étant pas le sujet d'étude, le choix de DFMAX dont le fonctionnement est décrit au chapitre 1 (page 12) semble approprié.

Tout d'abord, cet algorithme exact constitue une référence pour la comparaison entre les implémentations de différents algorithmes exacts de recherche d'une clique maximum.

Le complémentaire d'un graphe s'obtenant en temps polynomial, on considérera pour la suite que DFMAX effectue implicitement la résolution dans le graphe complémentaire et que, par conséquent, il résout le problème du stable. Notons qu'une légère modification de DFMAX permet de résoudre directement le problème du stable. Néanmoins plusieurs essais de cette version modifiée ont révélé que la résolution du problème de la clique maximum dans le complémentaire du graphe s'exécutait à un coût moindre.

De plus, cet algorithme constitue la base de plusieurs autres algorithmes. Ces derniers tentent de réduire le coût de résolution en modifiant, par exemple, la stratégie d'élagage de solutions de DFMAX. On pourrait donc penser qu'une séquence de transformations améliorant la performance de DFMAX améliorerait celle de certains autres algorithmes.

Le coût d'exécution de DFMAX a été mesuré en nombre de retours arrières. Cependant, dans la plupart des articles proposant un algorithme de résolution du problème du stable maximum le coût de l'algorithme est mesuré en temps d'exécution de l'algorithme. On trouvera les résultats de la mesure du coût d'exécution par le temps en annexe sous forme de tableaux.

Cependant, lorsque des résultats rapportant le temps d'exécution d'un algorithme sont présentés, il est impératif de calibrer l'ordinateur employé afin d'effectuer une comparaison significative avec des résultats obtenus sur une autre machine. À cet effet, des graphes étalons sont disponibles sur le site Internet du Second DIMACS Challenge (Trick, 1993).

Les expérimentations avec DFMAX ont été effectuées sur un Pentium i686 sous Linux. Les résultats de sa calibration avec les graphes étalons r.100.5.b, r.200.5.b, r.300.5.b, r.400.5.b et r.500.5.b disponibles sur (Trick, 1993) dans lesquels il faut résoudre le problème de la clique maximum sont respectivement de 0,01, 0,14, 1,13, 7,08 et 27,39 secondes.

3.1.3 Échantillon de graphes

Les graphes générés sont regroupés en classes selon leur nombre de sommets et leur densité. Les graphes ayant servi aux expérimentations ont initialement 50, 60, 70, 80, 90 et 100 sommets. Pour chacun de ces nombres de sommets, dix graphes de densité 0,01, 0,02, ..., 0,15, 0,20, 0,30, ..., 0,80 et 0,90 ont été générés aléatoirement.

L'échantillon de graphes employé compte donc 138 classes de graphes et pour chacune de ces classes, dix graphes ont été choisis aléatoirement avec remise. Cependant, en observant les caractéristiques de chacun des graphes de l'échantillon, il est possible d'affirmer que tous les graphes de l'échantillon sont deux à deux différents.

Quelques séquences de transformations décrites à la section 3.3 ont été appliquées aux graphes de l'échantillon précédemment décrit. Pour chacune des séquences de transformations, on obtient donc 138 classes de graphes transformés correspondant au 138 classes de graphes de l'échantillon.

Notons que certains résultats concernant les graphes de référence du Second DIMACS Challenge (Trick, 1993) sont également présentés mais que les séquences de transformations n'ont pas été appliquées à ces graphes.

3.1.4 Module informatique

Afin d'effectuer les expérimentations relatives à la démarche expliquée ci-dessus, il est impératif de posséder un outil informatique. À cette fin, une bibliothèque permettant la manipulation des graphes autant pour les générer que pour les transformer a été développée. Cette bibliothèque est un ensemble de classes C++ qui ont été développées dans un environnement Linux. Elle comporte sept classes:

- *NombresAleatoires*,
- *Graphe*,
- *Operation*,
- *Transformation*,
- *StructionsInversees*,
- *SuppressionsSommetDominant*,
- *InsertionsArete*.

Comme son nom l'indique, la classe *NombresAleatoires* permet de générer des nombres pseudo-aléatoires. Elle génère uniformément un réel entre 0 et 1 ou un entier entre 1 et N où N est déterminé par l'utilisateur. Cette classe est utile pour la génération de graphes aléatoires ainsi que pour les séquences de transformations SIALEA1 et SIALEA2 décrites à la sous-section 3.3.1.

La classe *Graphe* implémente la génération d'un graphe, les demandes d'information pouvant être effectuées par rapport à un graphe et les manipulations simples pouvant être effectuées sur un graphe. La structure de données choisie pour représenter

un graphe est la matrice d'adjacence. Notons qu'il est possible de construire un graphe à partir d'un fichier produit selon le format texte de représentation d'un graphe du Second DIMACS Challenge. La sauvegarde d'un graphe s'effectue en produisant un tel fichier.

La classe *Operation* définit un type énumératif *codeOperation* qui codifie les différentes opérations pouvant être effectuées sur un graphe. Ces opérations ont été décrites au début du chapitre 2.

La classe *Transformation* implémente des transformations traitées au chapitre 2:

- suppression d'une arête (2.4.1),
- insertion d'une arête (2.4.2),
- suppression d'un sommet (2.4.3),
- identification de deux sommets (2.4.4),
- struction inversée (2.5.9).

Elle utilise les opérations de la classe *Operation* qu'elle insère dans une file. Cette insertion est effectuée dans l'ordre prescrit par les algorithmes correspondant à ces transformations.

Notons que ces cinq transformations sont suffisantes pour effectuer les séquences de transformations qui seront vues à la section 3.3. En effet, la suppression d'un sommet dominant est en fait la suppression d'un sommet possédant la caractéristique d'être dominant, mais les opérations effectuées pour le supprimer sont les mêmes que pour un sommet quelconque. Il en est de même pour l'insertion d'une arête lorsqu'il y a un bon triplet de sommets et l'insertion d'une arête quelconque.

Finalement, les trois dernières classes implémentent les séquences de transformations qui sont décrites à la section 3.3.

Puisque cette bibliothèque a été élaborée à l'extérieur du contexte de la maîtrise, les détails de conception ne sont pas fournis dans ce mémoire. Pour plus de détails sur celle-ci, un document de conception et le code source sont disponibles sur demande (Dufresne, 2002).

À partir de cette bibliothèque, le développement de programmes générant et transformant les graphes s'effectue aisément. Par exemple, pour générer un graphe, il suffit d'appeler le constructeur de graphe de la classe *Graphe* et pour le sauvegarder, il suffit d'appeler la fonction qui produit un fichier texte selon les normes du Second DIMACS Challenge .

Pour appliquer une transformation ou une des trois séquences de transformations à un graphe particulier, il s'agit tout simplement de construire le graphe à partir du fichier dans lequel il est sauvegardé puis d'appeler la transformation ou la séquence de transformations souhaitée puis de sauvegarder le graphe résultant.

3.2 Évaluation expérimentale des estimateurs du coût de résolution

Les graphiques de cette section, figures 3.2 à 3.6, présentent les résultats de la résolution du problème du stable maximum permettant de déterminer la valeur des estimateurs du coût de résolution de DFMAX. Ces résultats proviennent de l'exécution de la première étape du protocole visant à construire un ensemble d'estimateurs du coût de résolution du problème du stable maximum. Ils représentent la moyenne du coût de résolution de chacune des classes de graphes de l'échantillon présenté à la sous-section 3.1.3.

Notons qu'afin d'éviter la redondance, seuls les graphiques se rapportant aux graphes de 50 et 100 sommets sont présentés. Les graphiques concernant les autres graphes seraient similaires à ceux présentés. Les résultats pour l'ensemble des graphes de l'échantillon se trouvent sous forme de tableaux en annexe.

L'ensemble de ces valeurs semble montrer que, pour un même nombre de sommets, le coût de résolution de DFMAX est plus élevé pour des densités comprises dans l'intervalle $[d_1, d_2]$ où d_1 et d_2 dépendent du nombre de sommets. L'intervalle $[0, d_1[$, où d_1 est près de 0, décroît et l'intervalle $]d_2, 1]$ croît lorsque le nombre de sommets augmente. Ainsi, lorsqu'on se réfère aux faibles densités, on se réfère aux densités de l'intervalle $[d_1, d_2]$.

Signalons que le coût de résolution en fonction de la structure (n, d) a une valeur près du nombre de sommets n lorsque DFMAX est l'algorithme de résolution du problème du stable maximum utilisé et que la densité tend vers 0. Puisque la plus petite densité examinée est 0,01, ce fait n'apparaît pas sur le graphique 3.4.

De plus, pour une même densité, notons que la valeur du coût de résolution croît plus rapidement que le nombre de sommets.

Les graphiques des figures 3.2 et 3.3 présentent la valeur des estimateurs pour les graphes dont la structure comporte respectivement 50 et 100 sommets. Les points représentent la valeur calculée d'un estimateur $c_A(n, d)$ où $n = 50$ ou 100 sommets et la valeur de la densité d peut être lue sur l'axe des abscisses.

Le graphique de la figure 3.4 présente la valeur des estimateurs du coût de résolution $c_A(n, d)$ sous forme de courbes de niveau. Notons que l'échelle de ces courbes de niveau est logarithmique. Les estimateurs ont été calculés à partir de l'ensemble des graphes de l'échantillon. Ce graphique donne donc une idée générale du coût de résolution du problème du stable maximum à l'aide de DFMAX dans les graphes

de 50 à 100 sommets.

Finalement, le graphique de la figure 3.5 représente le nombre de stabilité en fonction de la densité et le graphique de la figure 3.6 représente le coût de résolution du problème du stable maximum en fonction du nombre de stabilité. Cela permet d'observer ce coût en fonction d'un autre paramètre du graphe que ceux considérés pour décrire sa structure.

Dans le premier, chaque point représente le nombre de stabilité d'un graphe de 100 sommets de l'échantillon selon la densité espérée et chaque point du nuage de points du second graphique représente le coût de résolution du problème du stable maximum pour un graphe de 100 sommets de l'échantillon. Tous les graphes de 100 sommets de l'échantillon sont représentés.

Le premier graphique montre que le nombre de stabilité décroît avec la densité pour les graphes de 100 sommets de l'échantillon. Notons que la décroissance est plus marquée pour les densités inférieures à 0, 2. Le second graphique semble révéler que la résolution du problème du stable maximum est plus coûteuse lorsque la valeur du nombre de stabilité est légèrement inférieure au nombre de sommets divisé par deux. C'est qu'on remarque que la majorité des points du nuage de points se situent dans l'intervalle $[24, 78]$ de l'axe des abscisses et que les points correspondant aux coûts de résolution les plus élevés ont leur abscisse près du milieu de cet intervalle.

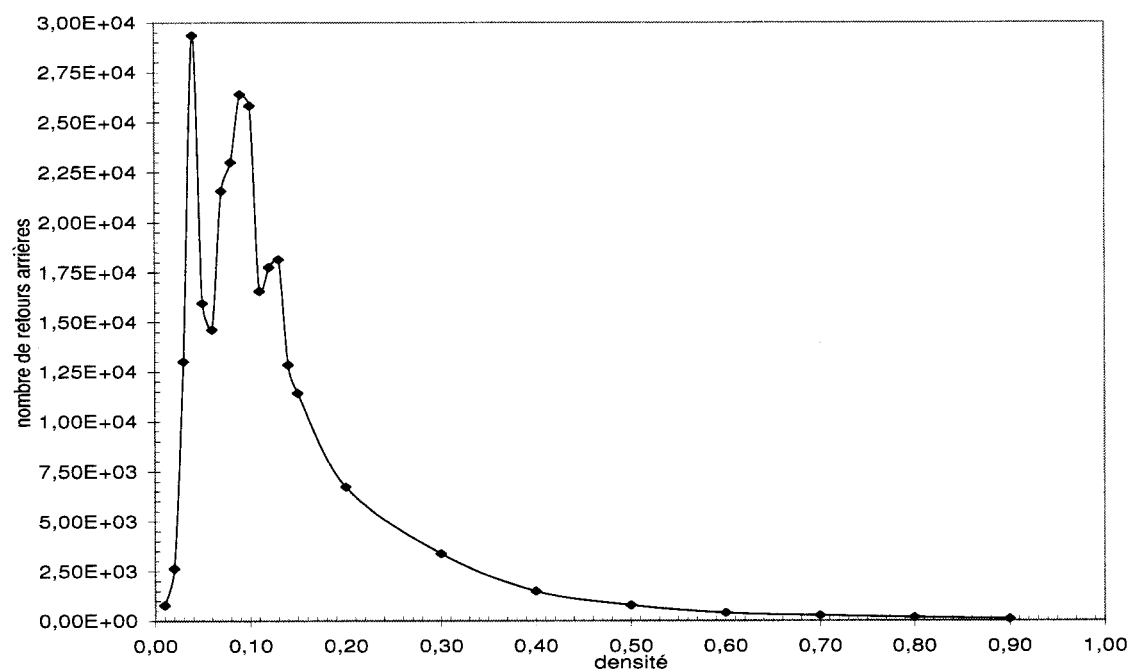


Figure 3.2 Coût de résolution du problème du stable maximum à l'aide de DFMAX en fonction de la densité espérée des graphes initiaux de 50 sommets

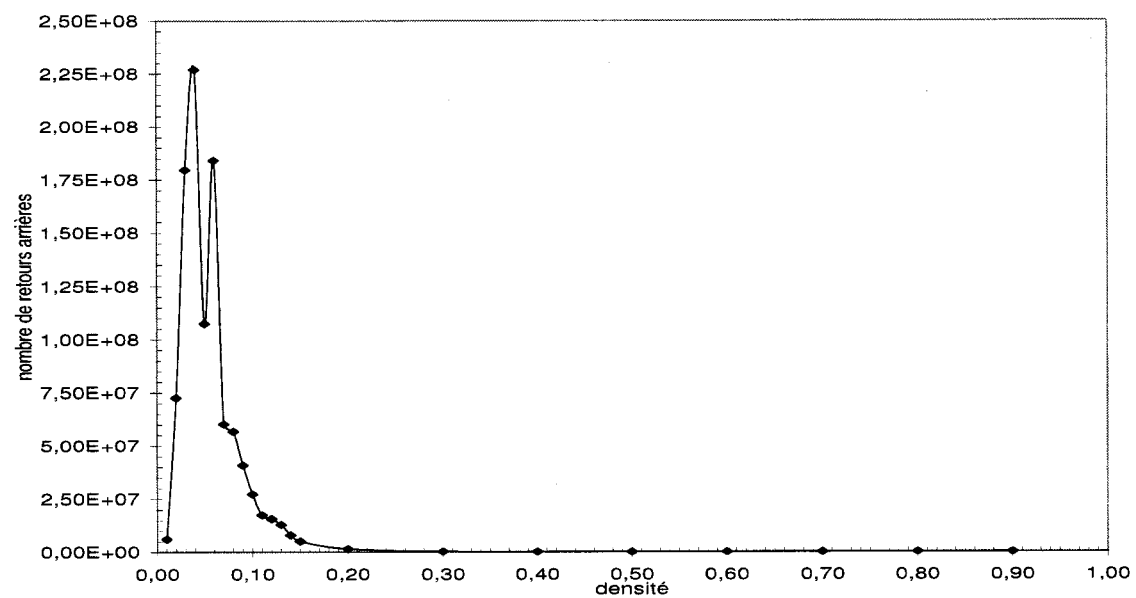


Figure 3.3 Coût de résolution du problème du stable maximum à l'aide de DFMAX en fonction de la densité espérée des graphes initiaux de 100 sommets

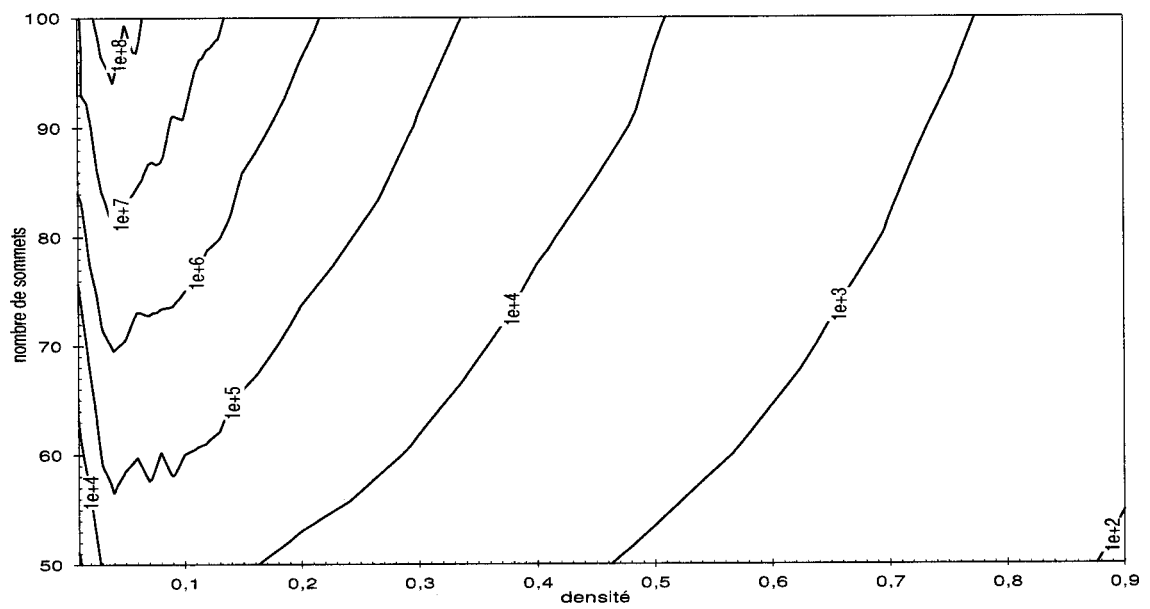


Figure 3.4 Courbe de niveau du coût de résolution du problème du stable maximum à l'aide de DFMAX en fonction de la structure des graphes

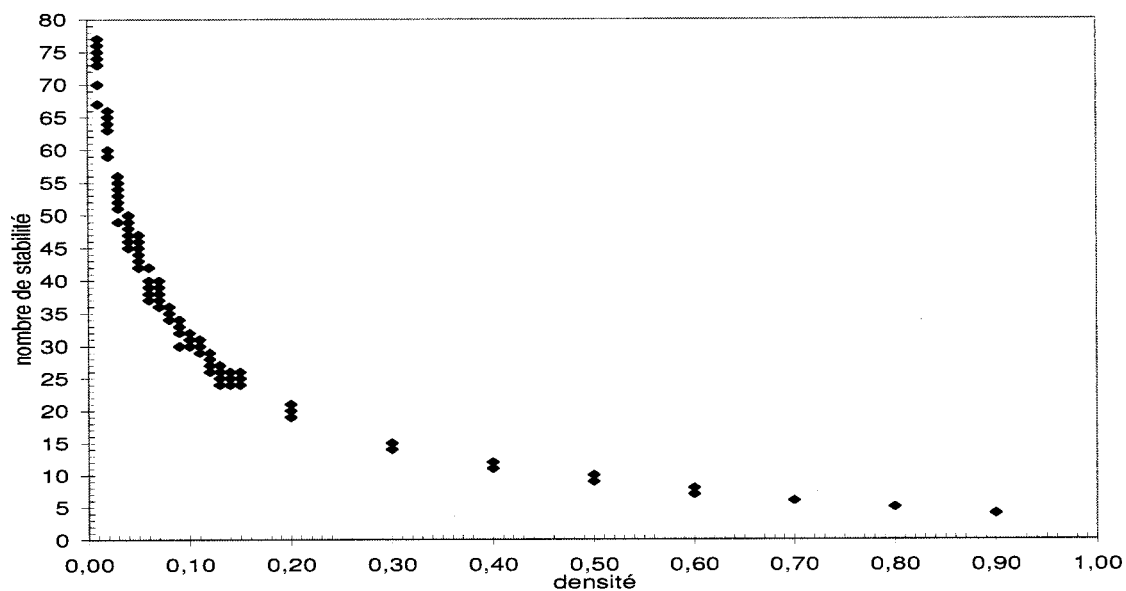


Figure 3.5 Nombre de stabilité en fonction de la densité des graphes de 100 sommets

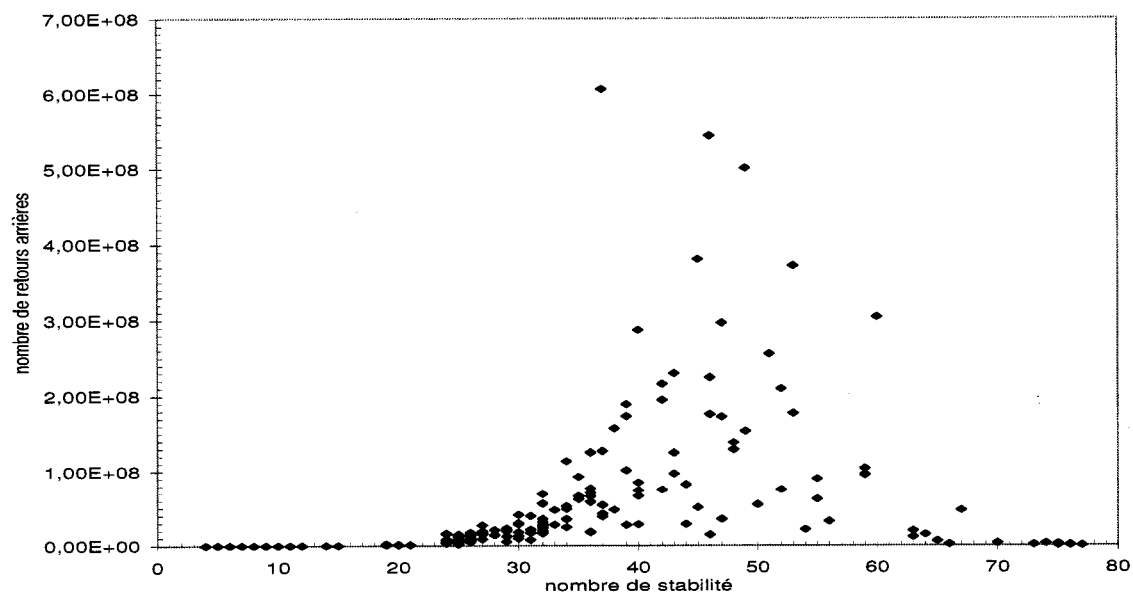


Figure 3.6 Coût de résolution du problème du stable maximum à l'aide de DFMAX en fonction du nombre de stabilité des graphes de 100 sommets

3.3 Séquences de transformations

Une séquence de transformations décrit la façon de choisir les graphes de la suite générée lors de l'application itérative d'une transformation à un graphe. La séquence de transformations est donc constituée de critères de choix des paramètres de transformation et d'un critère d'arrêt.

Dans cette recherche, trois transformations vues au chapitre 2 ont été étudiées:

- la struction inversée (2.5.9),
- la suppression d'un sommet dominant (2.5.3),
- l'insertion d'une arête lorsqu'il y a un bon triplet de sommets (2.5.7).

3.3.1 Séquence de structions inversées

Quatre séquences de transformations utilisant la struction inversée ont été expérimentées. Elles sont toutes constituées de la même façon.

Soit $G = (V, E)$ un graphe.

La première séquence de structions inversées, notée SI1, consiste à sélectionner l'arête $\{x, y\} \in E$ dont les sommets incidents, $x \in V$ et $y \in V$, possèdent le plus de voisins communs et à effectuer $G' = T_{SI}(G, \{x, y\})$. En cas d'égalité, la première arête dans l'ordre lexicographique est choisie. La procédure est recommencée sur le graphe résultant G' jusqu'à ce que le plus grand ensemble de voisins communs à deux sommets adjacents soit de cardinalité inférieure ou égale à 1.

Une autre séquence, notée SIALEA1, diffère de la première au niveau du choix de l'arête. À chaque itération, l'arête est sélectionnée selon les mêmes critères

mais à partir d'un sous-ensemble d'arêtes $F \in E$ tel que chaque arête de E a une probabilité de $\frac{1}{2}$ d'appartenir à F . Le choix des arêtes est aléatoire.

Les deux dernières séquences de structions inversées à présenter, notées SI2 et SIALEA2 diffèrent des précédentes au niveau du critère d'arrêt. Au lieu de terminer lorsque le plus grand ensemble de voisins communs à deux sommets adjacents est de cardinalité inférieure à 2, ces séquences se terminent lorsque la cardinalité de cet ensemble est inférieure ou égale à 2.

L'algorithme 3.1 est celui des séquences de structions inversées telles que décrites ci-dessus. Pour SI1 et SI2, la probabilité p est de 1 et le maximum de voisins communs qu'auront deux sommets adjacents dans le graphe résultant est respectivement 1 et 2 tandis que pour SIALEA1 et SIALEA2, la probabilité p est de $\frac{1}{2}$ et le maximum de voisins communs est respectivement 1 et 2.

<p>Entrée : Un graphe $G = (V, E)$, une probabilité p de choisir une arête et un maximum de voisins communs $MaxVoisins$</p> <p>Sortie : Un graphe G'</p> <p>début</p> <p style="padding-left: 20px;">$N \leftarrow 0$</p> <p style="padding-left: 20px;">répéter</p> <p style="padding-left: 40px;">pour chaque arête $\{u, v\} \in V$ faire</p> <p style="padding-left: 60px;">si $\{u, v\}$ est choisie selon la probabilité p alors</p> <p style="padding-left: 80px;">si $N(u) \cap N(v) > N$ alors</p> <p style="padding-left: 100px;">$N \leftarrow N(u) \cap N(v)$</p> <p style="padding-left: 100px;">$x \leftarrow u, y \leftarrow v$</p> <p style="padding-left: 60px;">si $N \geq MaxVoisins$ alors</p> <p style="padding-left: 80px;">$G \leftarrow T_{SI}(G, \{x, y\})$</p> <p style="padding-left: 40px;">jusqu'à $N \leq MaxVoisins$</p> <p>fin</p>
--

Algorithme 3.1 : Séquence de structions inversées

3.3.2 Séquence de suppressions de sommets dominants

Soit $G = (V, E)$ un graphe. La séquence de transformations utilisant la suppression d'un sommet dominant, notée SSD, consiste à construire l'ensemble $S \subset V$ contenant tous les sommets dominants dont l'ensemble de voisins contient au moins

un sommet n'appartenant pas à S puis, pour chaque sommet $u \in S$, à effectuer $G = T_{sSD}(G, u)$. La procédure est recommencée jusqu'à ce qu'il n'y ait plus de sommets dominants.

L'algorithme 3.2 présente la séquence de suppressions de sommets dominants utilisée.

```

Entrée : Un graphe  $G = (V, E)$ 
Sortie : Un graphe  $G'$ 
début
   $S \leftarrow \emptyset$ 
  pour chaque sommet dominant  $u \in V$  tel qu'il existe  $v \in N(u) - (N(u) \cap S)$  faire
     $S \leftarrow S \cup \{u\}$ 
  tant que  $S \neq \emptyset$  faire
    pour chaque sommet  $u \in S$  faire
       $G \leftarrow T_{sSD}(G, u)$ 
     $S \leftarrow \emptyset$ 
    pour chaque sommet  $u \in V$  tel qu'il existe  $v \in N(u) - (N(u) \cap S)$  faire
       $S \leftarrow S \cup \{u\}$ 
fin

```

Algorithme 3.2 : Séquence de suppressions d'un sommet dominant

Notons qu'il est possible d'approximer le nombre de sommets dominants que contient un graphe généré aléatoirement. Soit $G = (V, E)$ un graphe généré aléatoirement de n sommets et de densité p . Il y a $n(n-1)$ façon d'ordonner une paire de sommets de V et celle-ci forme une arête de G avec une probabilité p . De plus, considérons une paire de sommets ordonnée $(x, y) \subset V$. Pour chaque sommet $s \in V - \{x, y\}$, on a $s \notin N(x)$ avec une probabilité de $1-p$ et $s \in N(x)$ avec une probabilité p . Par conséquent, $N(y) \subseteq N(x)$ avec une probabilité de $(1 - (p(1-p)))^{n-2}$. Il est donc possible d'approximer le nombre de sommets dominants présents dans un graphe par

$$n_{SD}(G) \approx n \cdot (n-1) \cdot p \cdot (1-p+p^2)^{n-2}.$$

3.3.3 Séquence d'insertions d'arêtes

Soit $G = (V, E)$ un graphe. Supposons que les sommets de G sont numérotés de 1 à $|V|$. La séquence de transformations utilisant l'insertion d'une arête, notée IATS, consiste à parcourir chaque ensemble de trois sommets $\{u, v, w\} \subseteq V$ à partir de $\{1, 2, 3\}$. Pour chacun de ces ensembles, il s'agit d'évaluer les six triplets que peut former l'ensemble traité pour déterminer s'il y en a un bon et le cas échéant, insérer une arête. Lorsque le triplet $\{|V| - 2, |V| - 1, |V|\}$ a été traité, on recommence la recherche jusqu'à ce qu'il n'y ait plus de bons triplets de sommets.

L'algorithme 3.3 présente le détail de la séquence d'insertions d'arêtes.

<p>Entrée : Un graphe $G = (V, E)$ Sortie : Un graphe G' début</p> <div style="margin-left: 20px;"> <p>tant que il existe de bons triplets de sommets dans G faire</p> <div style="margin-left: 20px;"> <p>pour chaque triplet de sommets $\{u, v, w\} \subseteq V$ faire</p> <div style="margin-left: 20px;"> <p>si (u, v, w) forme un bon triplet de sommets alors</p> <div style="margin-left: 20px;"> <p>$G \leftarrow T'_{iAtS}(G, (u, v, w))$</p> </div> <p>sinon si (w, v, u) forme un bon triplet de sommets alors</p> <div style="margin-left: 20px;"> <p>$G \leftarrow T'_{iAtS}(G, (w, v, u))$</p> </div> <p>sinon si (u, w, v) forme un bon triplet de sommets alors</p> <div style="margin-left: 20px;"> <p>$G \leftarrow T'_{iAtS}(G, (u, w, v))$</p> </div> <p>sinon si (v, w, u) forme un bon triplet de sommets alors</p> <div style="margin-left: 20px;"> <p>$G \leftarrow T'_{iAtS}(G, (v, w, u))$</p> </div> <p>sinon si (v, u, w) forme un bon triplet de sommets alors</p> <div style="margin-left: 20px;"> <p>$G \leftarrow T'_{iAtS}(G, (v, u, w))$</p> </div> <p>sinon si (w, u, v) forme un bon triplet de sommets alors</p> <div style="margin-left: 20px;"> <p>$G \leftarrow T'_{iAtS}(G, (w, u, v))$</p> </div> </div> </div> </div> <p>fin</p>
--

Algorithme 3.3 : Séquence d'insertions d'arêtes

Notons qu'il est possible d'approximer le nombre de bons triplets de sommets que contient un graphe généré aléatoirement. Soit $G = (V, E)$ un graphe généré aléatoirement de n sommets et de densité p . Il y a $n(n-1)(n-2)$ façon d'ordonner un triplet de sommets de V et un triplet $(x, y, z) \subseteq V$ est tel que $\{x, z\} \in E$, $\{x, y\} \notin E$ et $\{y, z\} \notin E$ avec une probabilité $p(1-p)(1-p)$. De plus, pour chaque sommet $s \in V - \{x, y, z\}$, on a $s \notin N(x)$ avec une probabilité de $1-p$, $s \notin N(y)$

avec une probabilité de $1 - p$ et $s \in N(x)$ avec une probabilité p . Par conséquent, $N(y) \subseteq N(x) \cup N(y)$ avec une probabilité de $(1 - (p(1 - p)(1 - p)))^{n-3}$. Il est donc possible d'approximer le nombre de bon triplet de sommets présents dans un graphe par

$$n_{tS'}(G) \approx n \cdot (n - 1) \cdot (n - 2) \cdot p \cdot (1 - p) \cdot (1 - p) \cdot (1 - p + 2p^2 - p^3)^{n-3}.$$

3.3.4 Suite de séquences de transformations

Il est possible d'appliquer une suite de séquences de transformations à un graphe.

Soit G un graphe. Une première séquence \mathcal{T}_1 peut être appliquée à G et au graphe résultant, G' , une deuxième séquence \mathcal{T}_2 est appliquée d'où résulte G'' . Ainsi, G'' résulte de l'application de \mathcal{T}_1 suivie de celle de \mathcal{T}_2 à G . Notons qu'il est possible d'appliquer autant de séquences de transformations que désiré.

Les suites de séquences de transformations expérimentées sont:

- SIALEA1 suivie de SI1,
- SIALEA2 suivie de SI2,
- SI1 suivie de SSD.

3.3.5 Incidence sur la structure des graphes

Les graphiques de cette sous-section, figures 3.7 à 3.15, présentent la structure des graphes résultant de l'application de séquences de transformations par rapport à la structure des graphes initiaux. Ces résultats ont été produits afin de réaliser

la deuxième étape durant laquelle il faut déterminer l'impact de l'application de séquences de transformations sur la structure des graphes.

Les graphes transformés dont il est question résultent de l'application d'une séquence de transformations aux graphes de l'échantillon utilisés pour construire l'ensemble des estimateurs du coût de résolution du problème du stable maximum à l'aide de DFMAX.

Les courbes ont été tracées en utilisant la structure moyenne des graphes résultant de l'application d'une séquence de transformations. C'est-à-dire qu'à chaque classe de graphes initiaux utilisée correspond un ensemble de graphes transformés et la structure résultante moyenne correspondant à cette classe est déterminée en calculant la moyenne du nombre de sommets et de la densité de cet ensemble de graphes transformés.

Soit G_1, G_2, \dots, G_k k graphes initiaux de n sommets et de densité d . Soit les k graphes G'_1, G'_2, \dots, G'_k résultant de l'application d'une séquence de transformations \mathcal{T} aux graphes G_1, G_2, \dots, G_k telle que G'_i résulte de l'application de \mathcal{T} à G_i pour $i = 1, 2, \dots, k$. On note $n(G'_i)$ le nombre de sommets de G'_i et $d(G'_i)$ la densité de G'_i pour $i = 1, 2, \dots, k$. On suppose que G' représente les k graphes transformés. La structure moyenne résultant de l'application de \mathcal{T} aux graphes initiaux de n sommets et de densité d est

$$(n(G'), d(G')) = \left(\frac{\sum_{i=1}^k n(G'_i)}{k}, \frac{\sum_{i=1}^k d(G'_i)}{k} \right).$$

Les graphiques des figures 3.7 et 3.8 représentent le changement de structure des graphes respectivement de 50 et 100 sommets causé par l'application de SSD.

En général, on ne note pas de changement notable au niveau de la densité. Elle

décroît légèrement dans le domaine des très petites densités.

Cependant, pour certaines classes de graphes, le nombre de sommets diminue. En observant ces graphiques et les données présentées dans les tableaux en annexe, on constate que la longueur de l'intervalle de densités initiales pour lesquelles il n'y a aucun changement de structure suite à l'application de SSD croît avec le nombre de sommets initial. En général, il y a un changement de structure pour les densités initiales proches de 0 et 1. Autrement dit, il y a un changement de structure suite à l'application de SSD dans les intervalles de densité initiale $]0, d_1]$ et $[d_2, 1[$. Avec l'augmentation du nombre de sommets, d_1 s'approche de 0 et d_2 de 1.

Les graphiques des figures 3.9 et 3.10 représentent le changement de structure des graphes respectivement de 50 et 100 sommets causé par l'application de IATS. La dimension du nombre de sommets des graphes transformés est absente, car cette séquence de transformations n'affecte pas cet aspect de la structure des graphes auxquels elle est appliquée.

Ces graphiques et les données présentés dans les tableaux en annexe semblent montrer qu'en général, il y a une augmentation de la densité suite à l'application de IATS dans les intervalles de densité initiale $]0, d_1]$ et $[d_2, 1[$. Avec l'augmentation du nombre de sommets, d_1 s'approche de 0 et d_2 de 1.

L'observation de l'ensemble des résultats provenant de l'application de séquences de structions inversées permet de constater qu'en général, pour chacune des séquences de structions inversées expérimentées, le nombre de sommets des graphes transformés est croissant sur l'intervalle de densité initiale $[0, 01, 0, 7]$. Il y a une première exception concernant les graphes contenant initialement 70 et 80 sommets auxquels SI1 a été appliquée: la croissance cesse à la densité initiale 0,6. Une seconde exception concerne les graphes initialement de 50 sommets auxquels SI2 a été appliquée:

le nombre de sommets des graphes transformés est croissant sur la totalité du domaine de densités initiales expérimentées. En général, la croissance est plus forte pour les densités initiales supérieures à 0,2.

Le changement de structure au niveau de la densité est tel que tous les graphes résultant de l'application d'une séquence de structions inversées ont une densité inférieure ou égale à 0,1. La densité demeure identique ou diminue dans tous les cas. Sur un certain intervalle de densité initiale, la densité des graphes transformés est décroissante. Notons que cet intervalle ne correspond pas tout à fait à l'intervalle à l'intérieur duquel le nombre de sommets est croissant.

Les graphiques des figures 3.11, 3.12 et 3.13 représentent le changement de structure respectivement suite à l'application de SI2 à des graphes de 50 sommets et suite à l'application de SI1 à des graphes de 50 et 100 sommets. Notons que les axes de la densité des graphes transformés de ces graphiques ont des domaines réduits, ils ne couvrent pas l'intervalle $[0, 1]$ de densités possibles.

L'effet de ces séquences de structions inversées sur la structure étant similaire, les graphiques représentant le changement de structure des graphes de 100 sommets causé par l'application de SIALEA1, de SIALEA1 suivie de SI1, de SIALEA2 et de SIALEA2 suivie de SI2 ne sont pas présentés. Le lecteur est référé aux tableaux rapportant les résultats de ces séquences de structions inversées en annexe.

Le graphique de la figure 3.11 montre que SI2 a un impact sur la structure des graphes initialement de 50 sommets lorsque la densité initiale est supérieure ou égale à 0,08. De plus, pour les densités initiales supérieures ou égales à 0,15 le nombre de sommets des graphes transformés est croissante tandis que pour celles supérieures ou égales à 0,11, la densité des graphes transformés est décroissante.

Les graphiques des figures 3.12 et 3.13 et les résultats présents dans les tableaux

en annexe montrent que pour des densités initiales très petites, inférieure ou égale à 0,02, il ne semble pas y avoir de changement de structure des graphes causé par SI1.

Les résultats concernant SIALEA1, SIALEA1 suivie de SI1, SIALEA2 et SIALEA2 suivie de SI2 appliquées à des graphes initialement de 100 sommets semblent montrer que lorsque la probabilité de choisir une arête est 1, la séquence de structions inversées a une incidence sur des structures initiales de plus petites densités que lorsque cette probabilité est de $\frac{1}{2}$.

En effet, SIALEA1 a un impact sur les structures dont la densité est supérieure ou égale à 0,04 tandis que lorsqu'elle est suivie de SI1, il y a un impact à partir d'une densité de 0,03. De même, SIALEA2 a un impact sur les structures dont la densité est supérieure ou égale à 0,06 tandis que lorsqu'elle est suivie de SI2, il y a un impact à partir d'une densité de 0,05. De plus, on remarque que SIALEA1 et SI1 ont respectivement un impact sur la structure initiale à partir de densités plus petites que SIALE2 et SI2.

Les graphiques des figures 3.14 et 3.15 représentent le changement de structure des graphes respectivement de 50 et 100 sommets causé par l'application de SI1 suivie de celle de SSD.

En observant ces graphiques ainsi que les tableaux des résultats de l'application de SI1 suivie de SSD, il ne semble pas y avoir eu de changement de structure pour les graphes initiaux par rapport à l'application de SSD pour des densités initiales supérieures ou égales à 0,15 et inférieures à 0,8. Cependant, de façon similaire à SSD précédée d'aucune séquence de transformations, il y a un changement de structure suite à l'application de SSD dans les intervalles de densités initiales $]0, d_1]$ et $[d_2, 1[$. Avec l'augmentation du nombre de sommets initial, d_1 s'approche de 0

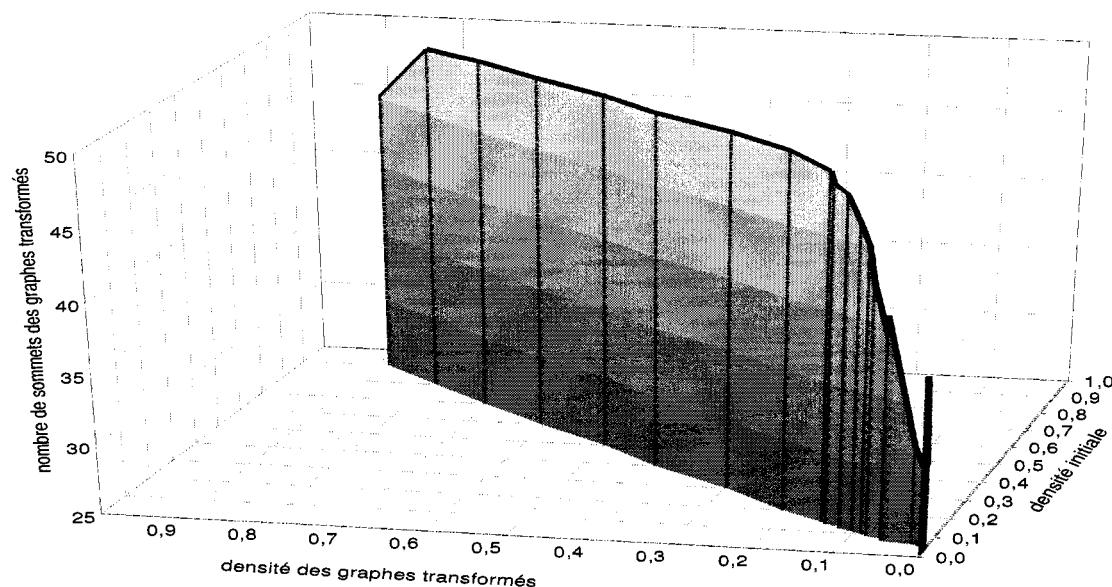


Figure 3.7 Structure résultant de l'application de SSD en fonction de la densité espérée des graphes initialement de 50 sommets

et d_2 de 1.

En résumé, il semble que SSD diminue le nombre de sommets sans trop affecter la densité et qu'il ait une grande incidence sur les graphes initialement de faible densité. IATS fait augmenter la densité tandis que le nombre de sommets demeure le même et il semble que cette séquence a également une plus grande incidence sur les graphes initialement de faible densité. Les séquences de structions inversées ont un effet contraire, elles diminuent la densité et augmentent le nombre de sommets. De plus, il semble que ces séquences aient une incidence plus grande sur les graphes d'assez forte densité.

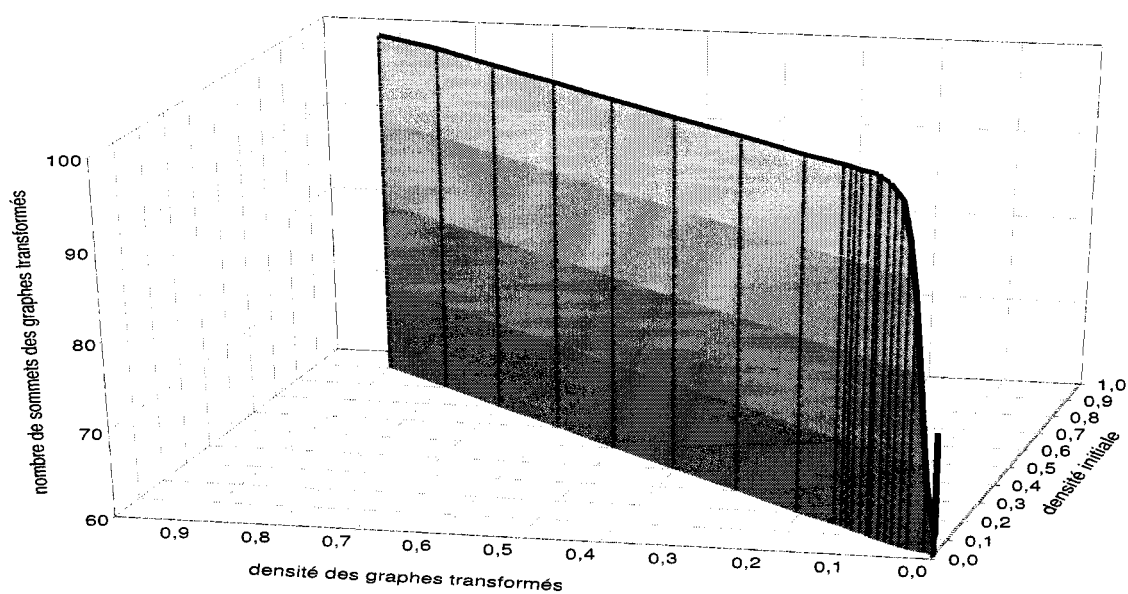


Figure 3.8 Structure résultant de l'application de SSD en fonction de la densité espérée des graphes initiaux initialement de 100 sommets

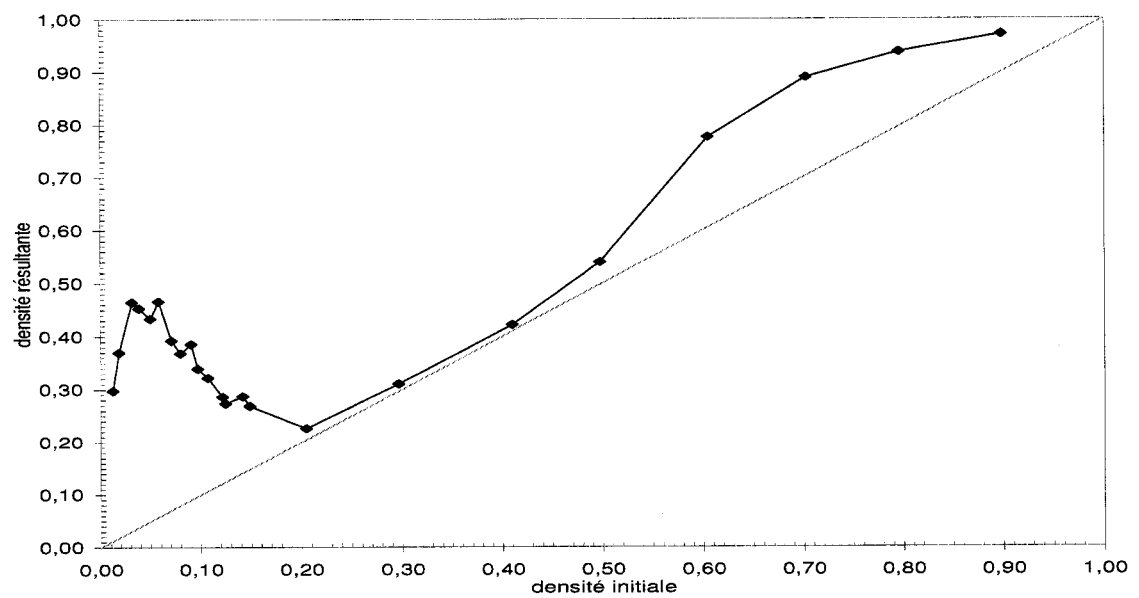


Figure 3.9 Structure résultant de l'application de IATS en fonction de la densité espérée des graphes de 50 sommets

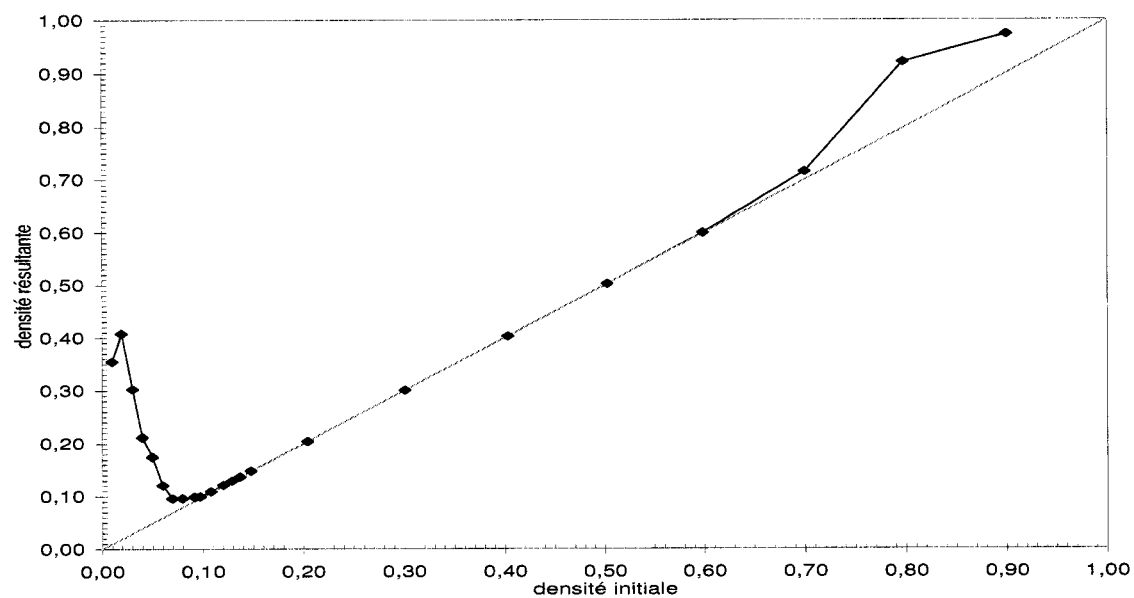


Figure 3.10 Structure résultant de l'application de IATS en fonction de la densité espérée des graphes de 100 sommets

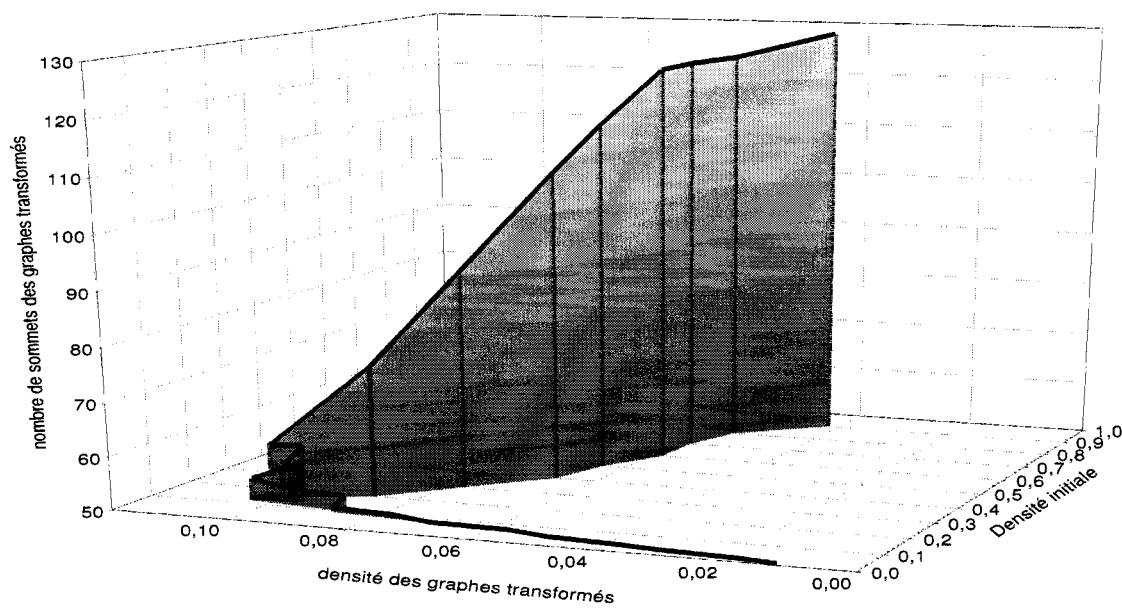


Figure 3.11 Structure résultant de l'application de SI2 en fonction de la densité espérée des graphes initialement de 50 sommets

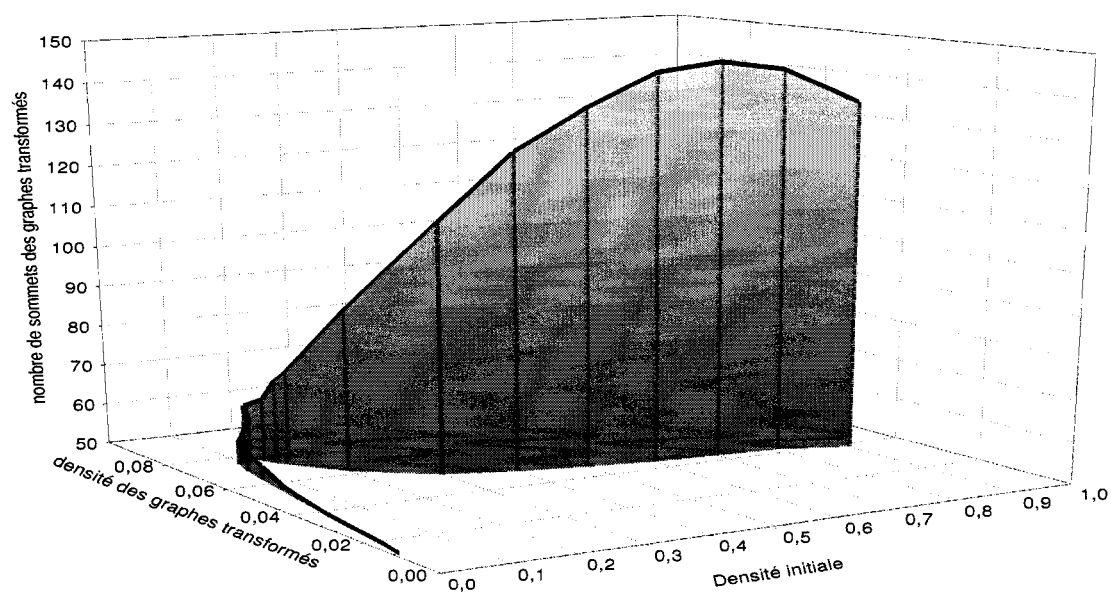


Figure 3.12 Structure résultant de l'application de SI1 en fonction de la densité espérée des graphes initialement de 50 sommets

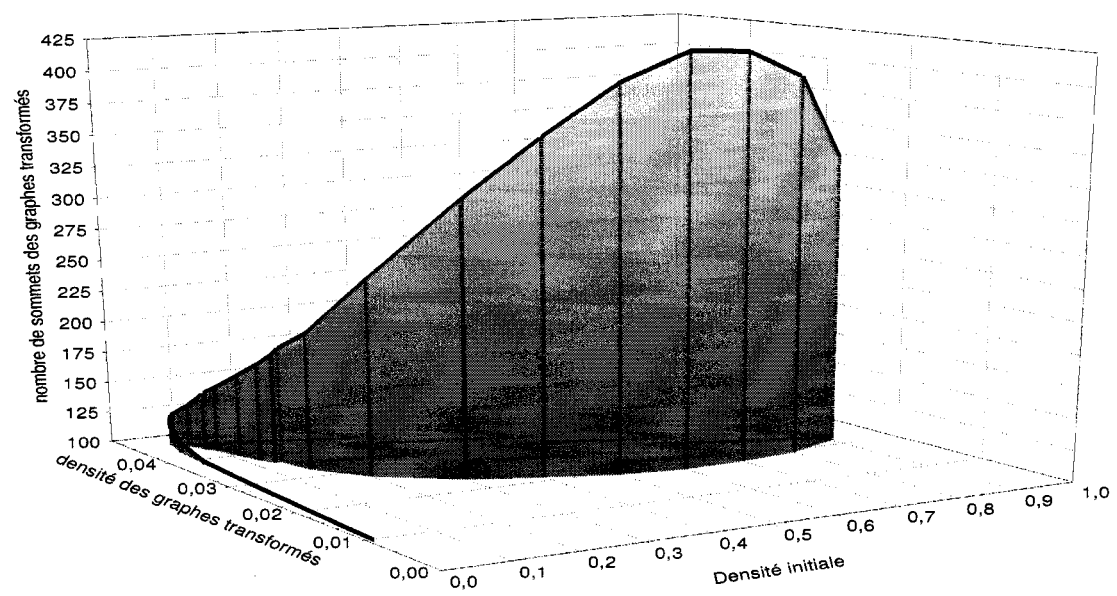


Figure 3.13 Structure résultant de l'application de SI1 en fonction de la densité espérée des graphes initialement de 100 sommets

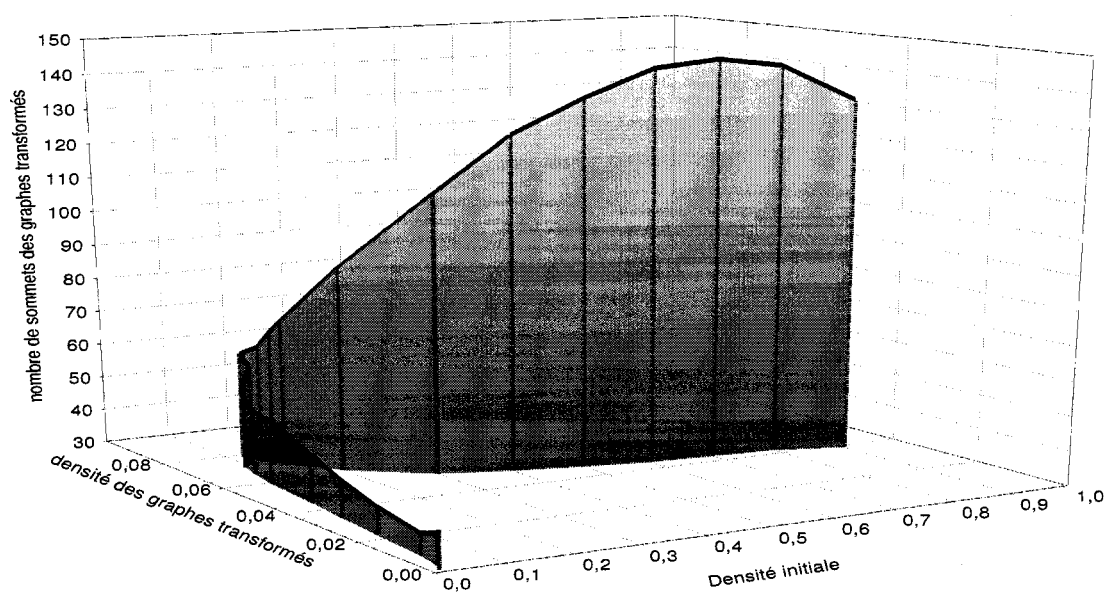


Figure 3.14 Structure résultant de l'application de SI1 suivie de SSD en fonction de la densité espérée des graphes initialement de 50 sommets

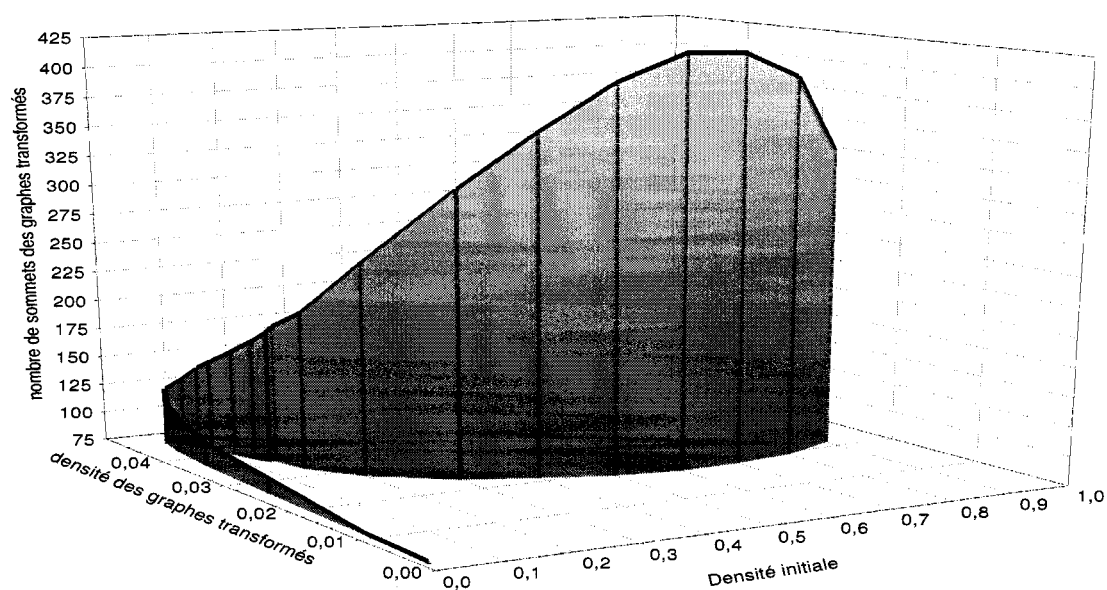


Figure 3.15 Structure résultant de l'application de SI1 suivie de SSD en fonction de la densité espérée des graphes initialement de 100 sommets

3.3.6 Graphes de référence du Second DIMACS Challenge

Le tableau 3.2 présente la structure, $(n(\overline{G}), d(\overline{G}))$, et le nombre de bons triplets de sommets, $n_{ts'}$, présents dans les graphes complémentaires des graphes de référence du Second DIMACS Challenge . Le nombre de sommets dominants n'est pas indiqué puisque qu'aucun de ces graphes n'en contient. Pour de plus amples informations sur les générateurs de ces graphes, voir (Trick, 1993).

Ces résultats servent à montrer que dans certaines classes de graphes générés de façon non aléatoire, il y a quelques fois des bons triplets de sommets. Cependant, aucune autre expérimentation n'a été effectuée sur ces graphes.

Notons que ce sont les graphes complémentaires qui nous intéressent, car ces graphes de référence sont habituellement utilisés pour la résolution de la clique maximum.

Tableau 3.2 Structure des graphes de référence du Second DIMACS Challenge

Identification du graphe	$(n(\overline{G}), d(\overline{G}))$	$n_{ts'}$
brock200-1.clq.b	(200, 0, 26)	0
brock200-2.clq.b	(200, 0, 50)	0
brock200-3.clq.b	(200, 0, 40)	0
brock200-4.clq.b	(200, 0, 34)	0
brock400-1.clq.b	(400, 0, 25)	0
brock400-2.clq.b	(400, 0, 25)	0
brock400-3.clq.b	(400, 0, 25)	0
brock400-4.clq.b	(400, 0, 25)	0
brock800-1.clq.b	(800, 0, 35)	0
brock800-2.clq.b	(800, 0, 35)	0

suite à la page suivante

Identification du graphe	$(n(\overline{G}), d(\overline{G}))$	$n_{tS'}$
brock800-3.clq.b	(800, 0, 35)	0
brock800-4.clq.b	(800, 0, 35)	0
c-fat200-1.clq.b	(200, 0, 92)	0
c-fat200-2.clq.b	(200, 0, 84)	0
c-fat200-5.clq.b	(200, 0, 57)	0
c-fat500-1.clq.b	(500, 0, 96)	0
c-fat500-2.clq.b	(500, 0, 63)	0
c-fat500-5.clq.b	(500, 0, 93)	0
c-fat500-10.clq.b	(500, 0, 81)	0
hamming6-2.clq.b	(64, 0, 10)	0
hamming6-4.clq.b	(64, 0, 65)	9408
hamming8-2.clq.b	(256, 0, 03)	0
hamming6-4.clq.b	(256, 0, 36)	0
hamming10-2.clq.b	(1024, 0, 01)	0
hamming10-4.clq.b	(1024, 0, 17)	0
johnson8-2-4.clq.b	(28, 0, 44)	0
johnson8-4-4.clq.b	(70, 0, 23)	0
johnson16-2-4.clq.b	(120, 0, 24)	0
johnson32-2-4.clq.b	(496, 0, 12)	0
keller4.clq.b	(171, 0, 35)	0
keller5.clq.b	(776, 0, 25)	0
keller6.clq.b	(3361, 0, 18)	n.d.
MANN_a9.clq.b	(45, 0, 073)	216
MANN_a27.clq.b	(378, 0, 001)	8424
MANN_a45.clq.b	(1035, 0, 004)	41580
MANN_a81.clq.b	(3321, 0, 001)	252720
p_hat300-1.clq.b	(300, 0, 76)	157
p_hat300-2.clq.b	(300, 0, 51)	32
p_hat300-3.clq.b	(300, 0, 26)	0

suite à la page suivante

Identification du graphe	$(n(\overline{G}), d(\overline{G}))$	$n_{tS'}$
p_hat500-1.clq.b	(500, 0, 75)	6
p_hat500-2.clq.b	(500, 0, 50)	4
p_hat500-3.clq.b	(500, 0, 25)	0
p_hat700-1.clq.b	(700, 0, 75)	1
p_hat700-2.clq.b	(700, 0, 50)	0
p_hat700-3.clq.b	(700, 0, 25)	0
p_hat1000-1.clq.b	(1000, 0, 76)	0
p_hat1000-2.clq.b	(1000, 0, 51)	0
p_hat1000-3.clq.b	(1000, 0, 26)	0
p_hat1500-1.clq.b	(1500, 0, 75)	0
p_hat1500-2.clq.b	(1500, 0, 50)	n.d.
p_hat1500-3.clq.b	(1500, 0, 25)	n.d.
san200_0.7_1.clq.b	(200, 0, 30)	0
san200_0.7_2.clq.b	(200, 0, 30)	77
san200_0.9_1.clq.b	(200, 0, 10)	0
san200_0.9_2.clq.b	(200, 0, 10)	0
san200_0.9_3.clq.b	(200, 0, 10)	0
san400_0.5_1.clq.b	(400, 0, 50)	51163
san400_0.7_1.clq.b	(400, 0, 30)	0
san400_0.7_2.clq.b	(400, 0, 30)	0
san400_0.7_3.clq.b	(400, 0, 30)	0
san400_0.9_1.clq.b	(400, 0, 10)	0
san1000.clq.b	(1000, 0, 50)	18460
sanr200_0.7.clq.b	(200, 0, 30)	0
sanr200_0.9.clq.b	(200, 0, 10)	0
sanr400_0.5.clq.b	(400, 0, 50)	0
sanr400_0.7.clq.b	(400, 0, 30)	0

3.4 Évaluation des gains escomptés et réels

Les graphiques et tableaux de cette section, figures 3.16 et 3.17 et tableaux 3.3 à 3.5, présentent quelques résultats provenant de la troisième étape de la méthode et permettant de justifier la validité de celle-ci. Ils présentent les résultats relatifs à l'estimation du gain escompté par les estimateurs du coût de résolution du problème du stable maximum à l'aide de DFMAX ainsi que la différence entre la valeur du coût estimé et la valeur du coût réel. Notons que plus le rapport des coûts de résolution est petit, plus le gain est grand.

Le coût de résolution du problème du stable maximum à l'aide de DFMAX est le coût moyen de résolution. Le calcul du coût moyen de résolution est identique à celui de la structure moyenne à la différence que le calcul implique les coûts de résolution au lieu des éléments de la structure.

Soit G_1, G_2, \dots, G_k k graphes initiaux de n sommets et de densité d . Soit les k graphes G'_1, G'_2, \dots, G'_k résultant de l'application d'une séquence de transformations \mathcal{T} aux graphes G_1, G_2, \dots, G_k telle que G'_i résulte de l'application de \mathcal{A} à G_i pour $i = 1, 2, \dots, k$. On note $C_{\mathcal{A}}(G'_i)$ le coût de résolution du problème du stable maximum dans G'_i . On suppose que G' représente les k graphes transformés. Le coût de résolution des graphes résultant de l'application de \mathcal{T} aux graphes initiaux de n sommets et de densité d est

$$C_{\mathcal{A}}(G') = \frac{\sum_{i=1}^k C_{\mathcal{A}}(G'_i)}{k}.$$

Les estimateurs du coût de résolution ne sont pas présents pour les graphes transformés ayant un nombre de sommets inférieur à 50 ou supérieur à 100 puisqu'aucun estimateur n'a été déterminé pour des structures dont le nombre de sommets est

inférieur à 50 ou supérieur à 100. De plus, lorsqu'un graphe transformé est tel que sa densité de 0,00, le coût de résolution est estimé par le nombre de sommets du graphe.

Le tableau 3.3 présente les résultats partiels de la résolution du problème du stable maximum dans quelques classes de graphes résultant de l'application de SSD. On remarque qu'en général le gain réel r'_A est supérieur ou égal au gain escompté r_A . Dans quelques cas, il y a un gain escompté alors qu'en réalité $r'_A \geq 1$. De plus, il arrive quelquefois que l'estimateur ne prédit pas le bon impact sur le coût de résolution, c'est-à-dire qu'il prédit une perte, mais il y a un gain et vice versa. Notons toutefois que, dans les cas de mauvaises prédictions, la différence entre la valeur du gain escompté et celle du gain réel est inférieure à 0,10 et que les deux valeurs r_A et r'_A sont très près de 1.

Le tableau 3.4 présente les résultats de la résolution du problème du stable maximum dans quelques classes de graphes résultant de l'application de IATS. On remarque qu'en général, le gain réel r'_A est supérieur ou égal au gain escompté r_A . Dans un cas, il y a un gain escompté alors que $r'_A > 1$. De plus, il arrive parfois qu'une perte est escomptée, alors que c'est un gain qui est obtenu. Notons toutefois que, dans les cas de mauvaises prédictions, la différence entre la valeur du gain escompté et celle du gain réel est inférieure à 0,25 et que les deux valeurs r_A et r'_A sont très près de 1.

Les graphiques des figures 3.16 et 3.17 montrent le coût de résolution du problème du stable maximum dans les graphes transformés par IATS appliquée à des graphes de respectivement 50 et 100 sommets. Les points représentent le coût de résolution dans les graphes transformés en fonction de la densité de ceux-ci tandis que la courbe représente les estimateurs du coût de résolution. Les étiquettes de données sur les points indiquent la densité des graphes avant leur transformation.

On remarque que l'écart entre les estimateurs et le coût réel semble moins important pour les graphes de 100 que pour ceux de 50 sommets. On sait que l'échelle du coût de résolution couvrant un plus grand intervalle, les données affichées sont moins précises.

Toutefois, en observant le tableau 3.4, on remarque qu'à l'exception des densités 0,01 à 0,03, l'estimateur du coût de résolution est plus près du coût réel pour les graphes de 100 sommets que pour ceux de 50 et 70 sommets. De plus, on remarque qu'à l'exception des densités 0,01 à 0,06, l'estimateur du coût de résolution est plus près du coût réel de résolution pour les graphes de 70 que pour ceux de 50.

Le tableau 3.5 présente les résultats de la résolution du problème du stable maximum dans certains graphes résultant de l'application de SI2 à des graphes initialement de 50 sommets. Il n'y a pas d'estimateur pour les graphes résultant de l'application de SI2 aux graphes de densité supérieure ou égale à 0,4, car les graphes résultant ont plus de 100 sommets. De plus, les résultats des graphes résultant de l'application de SI2 aux graphes de densité inférieure à 0,08 ne sont pas présentés puisque SI2 n'a pas d'incidence sur les graphes initialement de ces densités. Seuls les résultats concernant les graphes résultant de l'application de SI2 aux graphes de densité 0,05 sont présentés à titre d'exemple.

Puisque les rapports $r_{\mathcal{A}}$ et $r'_{\mathcal{A}}$ sont supérieurs ou égaux à 1, on parlera de perte escomptée et de perte réelle. Notons que plus le rapport des coûts de résolution est grand, plus la perte est grande.

On remarque que pour les graphes résultant de l'application de SI2 aux graphes initialement de densité supérieure ou égale à 0,12, la perte réelle $r'_{\mathcal{A}}$ est supérieure à la perte escomptée $r_{\mathcal{A}}$. La différence entre ces pertes semble s'accroître avec la densité initiale.

En résumé, bien que les estimateurs aient prédit une mauvaise incidence dans quelques cas, il semble toutefois qu'en général, la prédiction qu'ils font est bonne. Dans le cas de SI2, elle est toujours correcte et dans les cas de SSD et IATS, les quelques erreurs concernent, en général, des graphes tels que leur gain réel est proche de 1.

Les graphiques des figures 3.18 à 3.21 ainsi que les graphiques et tableaux précédemment mentionnés dans cette sous-section visent à montrer l'impact réel de l'application de séquences de transformations sur la résolution du problème du stable maximum. Ils constituent donc les résultats provenant de l'étape finale de la méthode.

Les graphiques des figures 3.18 et 3.19 semblent montrer que l'application de SSD aux graphes respectivement de 50 et 100 sommets procure un gain lors de la résolution du problème du stable maximum dans les graphes transformés par rapport aux graphes initiaux. Le tableau 3.3 et les tableaux en annexe semblent montrer la même chose. Remarquons que l'application de SSD ne procure jamais de perte véritable lors de la résolution du problème du stable maximum dans les graphes transformés par rapport aux graphes initiaux. Le rapport r'_A égale 1 lorsque SSD n'a aucune incidence sur la structure des graphes.

Les graphiques des figures 3.20 et 3.21 semblent montrer que l'application de IATS aux graphes respectivement de 50 et 100 sommets procure un gain lors de la résolution du problème du stable maximum dans les graphes transformés par rapport aux graphes initiaux de faible densité. Le tableau 3.4 et les tableaux en annexe semblent montrer la même chose. Remarquons que l'application de IATS produit quelquefois de légères pertes. Cependant, ces pertes se produisent pour certains graphes de densités initiales dans l'intervalle $[0, 13, 0, 60]$ et de nombre de sommets initial dans l'intervalle $[60, 90]$. Dans le cas où $r'_A \geq 1$, on remarque que la valeur

de r'_A est très proche de 1 et égale souvent 1.

Remarquons que les graphiques des figures 3.16 et 3.17 montrent également le gain réel grâce aux étiquettes de données qui, rappelons-le, indiquent la densité des graphes initiaux. Ainsi, sur la courbe, il est possible de voir le coût de résolution pour un graphe d'une certaine densité et de le comparer avec le coût associé au point dont l'étiquette indique cette densité. Ces graphiques permettent aussi de constater que la plupart du temps, le gain réel est supérieur au gain escompté donc que r'_A est supérieur à r_A .

Le tableau 3.5 présente une partie des résultats concernant l'application de SI2 aux graphes initialement de 50 sommets. Rappelons que la perte réelle semble s'accroître avec la densité initiale. De plus, on remarque que dans aucun cas, il n'y a de gain réel.

Les données relatives aux densités supérieures ou égales à 0,70 sont absentes, car la résolution du problème du stable dans les graphes transformés durait plus de dix heures alors que dans les graphes initiaux, elle ne prenait que quelques secondes. Afin de donner un aperçu sur la relation entre le coût de résolution mesuré en nombre de retours arrières et le coût de résolution mesuré en temps, mentionnons que, par exemple, 72 691 527 retours arrières prennent 70 secondes et que 44 307 102 581 retours arrières prennent 9 heures et 10 minutes.

Les résultats de la résolution du problème du stable dans les graphes résultant de l'application d'autres séquences de structions inversées ne sont pas présentés, car dans tous les cas où la séquence de structions inversées a une incidence, une perte est escomptée. Quelques essais ont appuyé le fait qu'il est beaucoup plus coûteux de résoudre le problème du stable dans les graphes transformés que dans les graphes initiaux. Ainsi, l'exemple de SI2 dresse un portrait de l'incidence des séquences

de structions inversées sur la résolution du problème du stable maximum à l'aide de DFMAX. On y remarque que plus la structure résultante de l'application de SI2 diffère de la structure du graphe initial, plus la perte est grande.

Finalement, les graphiques des figures 3.22 et 3.23 donnent une vue d'ensemble de l'incidence de SSD et IATS sur la résolution du problème du stable maximum dans des graphes initialement de 50 et 100 sommets respectivement. Notons que l'échelle du coût de résolution est logarithmique. Les valeurs, et donc les remarques, sont les mêmes que pour les graphiques des figures 3.18 à 3.21.

Somme toute, il semble que dans plusieurs cas, notamment ceux concernant les graphes initialement de faible densité, SSD et IATS procurent un gain réel. Les séquences de structions inversées semblent cependant beaucoup plus décevantes, lorsqu'il y a incidence d'une de ces séquences sur la structure du graphe initial, il y a une perte escomptée.

Tableau 3.3 Incidence de sSD sur la résolution du problème du stable maximum

Graphes (n, d)	Estimateurs $c_A(n, d)$	Graphes résultant de l'application de sSD				
		($n(G'), d(G')$)	$C_A(G')$	$c_A(n(G'), d(G'))$	r_A	r'_A
(50, 0, 01)	790	(37, 0, 00)	39	37	0,05	0,05
(50, 0, 02)	2641	(37, 0, 00)	36	37	0,01	0,01
(50, 0, 03)	13012	(33, 0, 00)	32	33	0,00	0,00
(50, 0, 04)	29365	(31, 0, 00)	30	31	0,00	0,00
(50, 0, 05)	15954	(31, 0, 01)	61	—	—	0,00
(50, 0, 06)	14626	(32, 0, 02)	99	—	—	0,01
(50, 0, 07)	21570	(41, 0, 06)	2350	—	—	0,11
(50, 0, 08)	23007	(38, 0, 06)	3286	—	—	0,14
(50, 0, 09)	26392	(43, 0, 08)	4649	—	—	0,18
(50, 0, 10)	25833	(45, 0, 09)	11618	—	—	0,45
(50, 0, 11)	16570	(46, 0, 10)	11358	—	—	0,69
(50, 0, 12)	17747	(48, 0, 12)	11241	—	—	0,63
(50, 0, 13)	18143	(48, 0, 12)	13587	—	—	0,75
(50, 0, 14)	12849	(49, 0, 14)	9932	—	—	0,77
(50, 0, 15)	11430	(49, 0, 15)	10298	—	—	0,90
(70, 0, 01)	26930	(55, 0, 00)	54	55	0,00	0,00
(70, 0, 02)	173204	(48, 0, 00)	47	48	0,00	0,00
(70, 0, 03)	743828	(44, 0, 00)	172	44	0,00	0,00
(70, 0, 04)	1087281	(46, 0, 02)	5639	—	—	0,01
(70, 0, 05)	897844	(58, 0, 04)	85560	159731	0,18	0,10
(70, 0, 06)	518004	(57, 0, 05)	58380	99411	0,19	0,11
(70, 0, 07)	575447	(65, 0, 07)	297713	368665	0,64	0,52
(70, 0, 08)	556621	(68, 0, 08)	389175	464531	0,83	0,70
(70, 0, 09)	551998	(68, 0, 09)	443391	469258	0,85	0,80
(70, 0, 10)	478176	(69, 0, 10)	445724	440308	0,92	0,93
(70, 0, 11)	390816	(70, 0, 11)	388581	390816	1,00	0,99
(70, 0, 12)	274804	(69, 0, 12)	255960	256011	0,93	0,93
(70, 0, 13)	247291	(70, 0, 13)	247291	247291	1,00	1,00
(70, 0, 14)	216050	(70, 0, 14)	216050	216050	1,00	1,00
(70, 0, 15)	169035	(70, 0, 15)	169035	169035	1,00	1,00
(100, 0, 01)	6133703	(74, 0, 00)	118	74	0,00	0,00
(100, 0, 02)	72691527	(62, 0, 00)	5330	62	0,00	0,00
(100, 0, 03)	179676999	(68, 0, 01)	5135645	22791	0,00	0,03
(100, 0, 04)	226993732	(88, 0, 04)	34144156	47261254	0,21	0,15
(100, 0, 05)	107466454	(94, 0, 05)	58094312	66164939	0,62	0,54
(100, 0, 06)	184115891	(98, 0, 06)	117257550	152806462	0,83	0,64
(100, 0, 07)	60215965	(99, 0, 07)	50686702	55733441	0,93	0,84
(100, 0, 08)	56777757	(100, 0, 08)	54108328	56777757	1,00	0,95
(100, 0, 09)	40842801	(100, 0, 09)	37604155	40842801	1,00	0,92
(100, 0, 10)	27153709	(100, 0, 10)	27153709	27153709	1,00	1,00
(100, 0, 11)	17407123	(100, 0, 11)	17407123	17407123	1,00	1,00
(100, 0, 12)	15668795	(100, 0, 12)	15668795	15668795	1,00	1,00
(100, 0, 13)	12937927	(100, 0, 13)	12937927	12937927	1,00	1,00
(100, 0, 14)	7966531	(100, 0, 14)	7966531	7966531	1,00	1,00
(100, 0, 15)	5139322	(100, 0, 15)	5139322	5139322	1,00	1,00

Tableau 3.4 Incidence de IATS sur la résolution du problème du stable maximum

Graphes (n, d)	Estimateurs $c_A(n, d)$	Graphes résultant de l'application de IATS				
		($n(G'), d(G')$)	$C_A(G')$	$c_A(n(G'), d(G'))$	r_A	r'_A
(50, 0, 01)	790	(50, 0, 30)	91	3378	4,28	0,11
(50, 0, 02)	2641	(50, 0, 37)	92	2060	0,78	0,03
(50, 0, 03)	13012	(50, 0, 46)	112	1071	0,08	0,01
(50, 0, 04)	29365	(50, 0, 45)	123	1142	0,04	0,00
(50, 0, 05)	15954	(50, 0, 43)	132	1283	0,08	0,01
(50, 0, 06)	14626	(50, 0, 47)	128	1001	0,07	0,01
(50, 0, 07)	21570	(50, 0, 39)	198	1683	0,08	0,01
(50, 0, 08)	23007	(50, 0, 37)	351	2060	0,09	0,02
(50, 0, 09)	26392	(50, 0, 39)	305	1683	0,06	0,01
(50, 0, 10)	25833	(50, 0, 34)	445	2625	0,10	0,02
(50, 0, 11)	16570	(50, 0, 32)	981	3001	0,18	0,06
(50, 0, 12)	17747	(50, 0, 29)	2055	3712	0,21	0,12
(50, 0, 13)	18143	(50, 0, 27)	3489	4381	0,24	0,19
(50, 0, 14)	12849	(50, 0, 29)	3762	3712	0,29	0,29
(50, 0, 15)	11430	(50, 0, 27)	2743	4381	0,38	0,24
(70, 0, 01)	26930	(70, 0, 32)	150	16950	0,63	0,01
(70, 0, 02)	173204	(70, 0, 39)	186	7527	0,04	0,00
(70, 0, 03)	743828	(70, 0, 45)	236	4414	0,01	0,00
(70, 0, 04)	1087281	(70, 0, 39)	244	7527	0,01	0,00
(70, 0, 05)	897844	(70, 0, 30)	1100	19642	0,02	0,00
(70, 0, 06)	518004	(70, 0, 33)	1000	15604	0,03	0,00
(70, 0, 07)	575447	(70, 0, 20)	12662	69111	0,12	0,02
(70, 0, 08)	556621	(70, 0, 17)	27179	129065	0,23	0,05
(70, 0, 09)	551998	(70, 0, 18)	81772	109081	0,20	0,15
(70, 0, 10)	478176	(70, 0, 14)	264671	216050	0,45	0,55
(70, 0, 11)	390816	(70, 0, 13)	167348	247291	0,63	0,43
(70, 0, 12)	274804	(70, 0, 17)	116140	129065	0,47	0,42
(70, 0, 13)	247291	(70, 0, 14)	146705	216050	0,87	0,59
(70, 0, 14)	216050	(70, 0, 15)	131330	169035	0,78	0,61
(70, 0, 15)	169035	(70, 0, 15)	140014	169035	1,00	0,83
(100, 0, 01)	6133703	(100, 0, 35)	286	105222	0,02	0,00
(100, 0, 02)	72691527	(100, 0, 41)	458	37420	0,00	0,00
(100, 0, 03)	179676999	(100, 0, 30)	6936	170086	0,00	0,00
(100, 0, 04)	226993732	(100, 0, 21)	433291	1310463	0,01	0,00
(100, 0, 05)	107466454	(100, 0, 17)	1342572	3658462	0,03	0,01
(100, 0, 06)	184115891	(100, 0, 12)	17194663	15668795	0,09	0,09
(100, 0, 07)	60215965	(100, 0, 10)	20751826	27153709	0,45	0,34
(100, 0, 08)	56777757	(100, 0, 10)	30034886	27153709	0,48	0,53
(100, 0, 09)	40842801	(100, 0, 10)	25914130	27153709	0,66	0,63
(100, 0, 10)	27153709	(100, 0, 10)	21813013	27153709	1,00	0,80
(100, 0, 11)	17407123	(100, 0, 11)	16451060	17407123	1,00	0,95
(100, 0, 12)	15668795	(100, 0, 12)	14642565	15668795	1,00	0,93
(100, 0, 13)	12937927	(100, 0, 13)	12729418	12937927	1,00	0,98
(100, 0, 14)	7966531	(100, 0, 14)	7780729	7966531	1,00	0,98
(100, 0, 15)	5139322	(100, 0, 15)	5037696	5139322	1,00	0,98

Tableau 3.5 Incidence de SI2 sur la résolution du problème du stable maximum

Classe de graphes (n, d)	Estimateur $c_A(n, d)$
(50, 0.05)	15954
(50, 0.08)	23007
(50, 0.09)	26392
(50, 0.10)	25833
(50, 0.11)	16570
(50, 0.12)	17747
(50, 0.13)	18143
(50, 0.14)	12849
(50, 0.15)	11430
(50, 0.20)	6722
(50, 0.30)	3378
(50, 0.40)	1495
(50, 0.50)	789
(50, 0.60)	399
(50, 0.70)	265
(50, 0.80)	168
(50, 0.90)	85

Graphes résultant de l'application de SI2				
$(n(G'), d(G'))$	$C_A(G')$	$c_A(n(G'), d(G'))$	r_A	r'_A
(50, 0.05)	15954	15954	1,00	1,00
(51, 0.08)	30006	30324	1,32	1,30
(51, 0.09)	30030	37583	1,42	1,14
(53, 0.09)	56194	59964	2,32	2,18
(53, 0.10)	34218	47932	2,89	2,07
(54, 0.10)	60328	55298	3,12	3,40
(56, 0.10)	92600	70030	3,86	5,10
(61, 0.10)	285764	137363	10,7	22,2
(60, 0.10)	359643	99495	8,70	31,5
(75, 0.09)	4934219	1699171	253	734
(91, 0.08)	63909529	21081412	6241	18919
(107, 0.07)	1607883490	—	—	1075507
(115, 0.06)	4851080271	—	—	6148391
(124, 0.06)	44307102581	—	—	111045370
(125, 0.06)	—	—	—	—
(125, 0.05)	—	—	—	—
(129, 0.04)	—	—	—	—

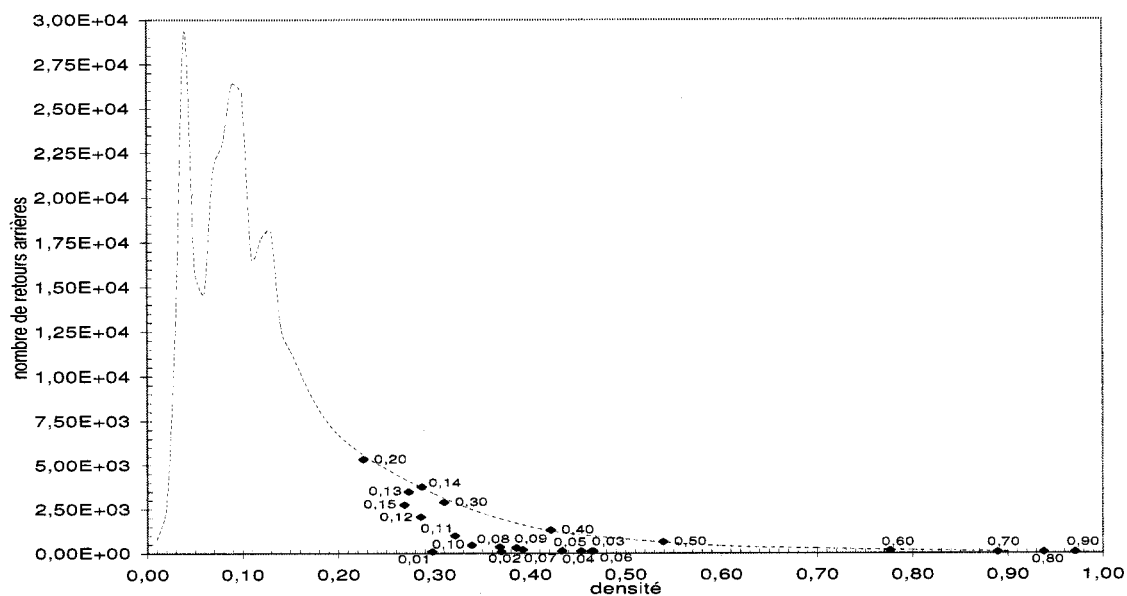


Figure 3.16 Coût de résolution du problème du stable maximum à l'aide de DFMAX en fonction de la densité des graphes résultant de l'application de IATS à des graphes initialement de 50 sommets

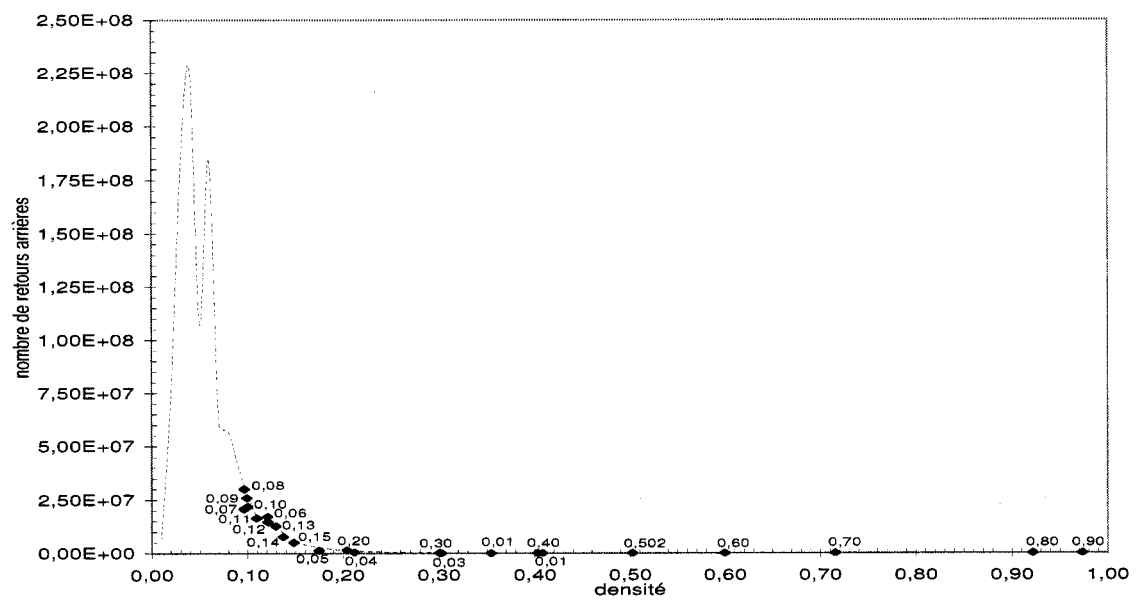


Figure 3.17 Coût de résolution du problème du stable maximum à l'aide de DFMAX en fonction de la densité des graphes résultant de l'application de IATS à des graphes initialement de 100 sommets

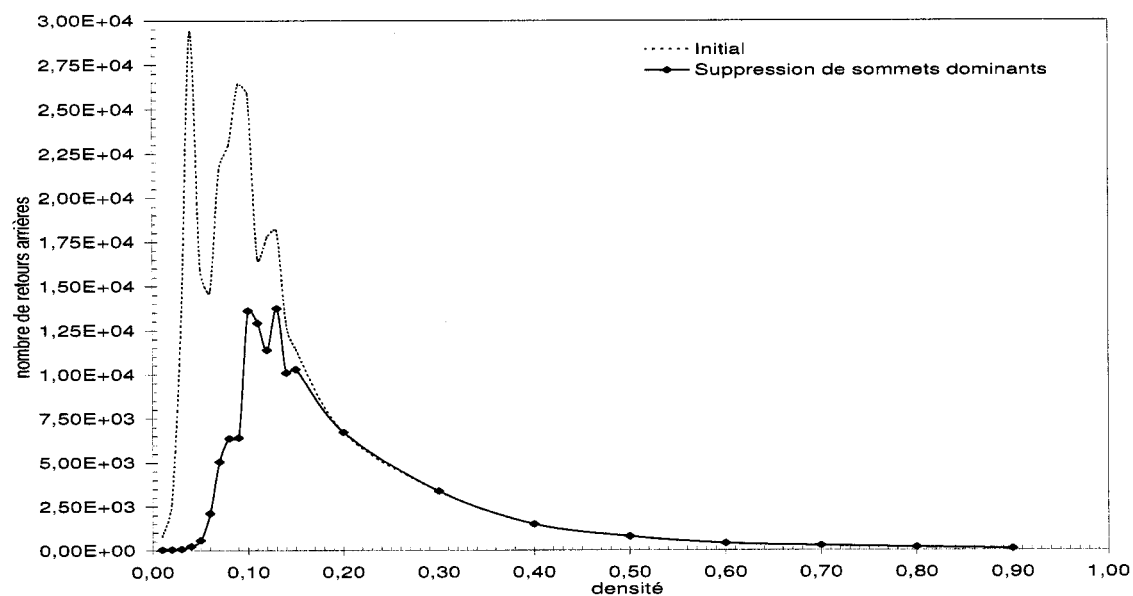


Figure 3.18 Coût de résolution du problème du stable maximum à l'aide de DFMAX appliqué aux graphes résultant de l'application de SSD en fonction de la densité espérée des graphes initialement de 50 sommets

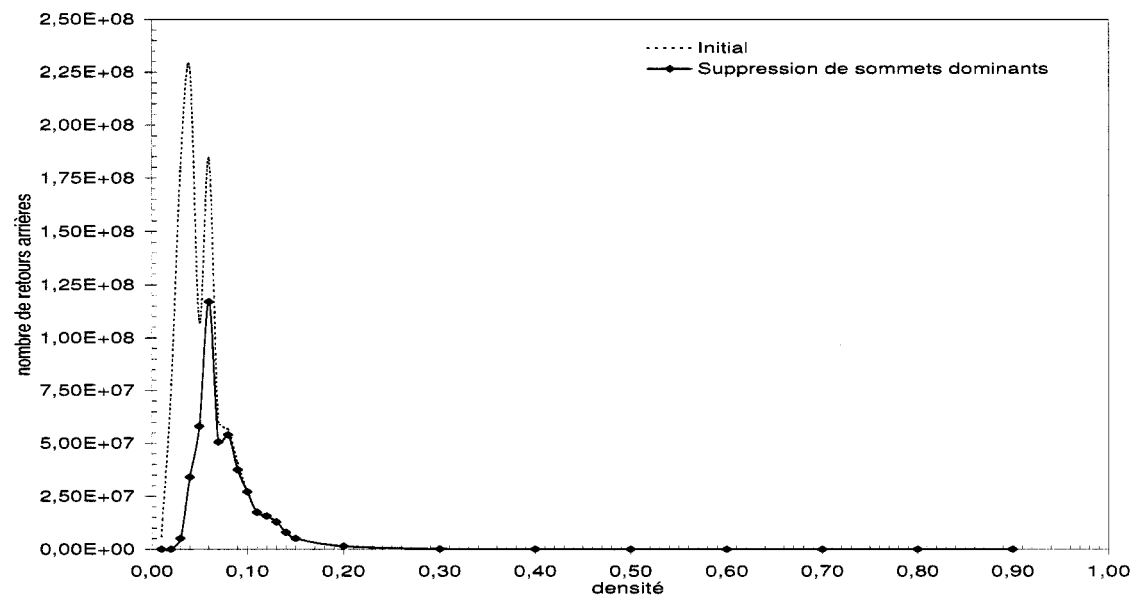


Figure 3.19 Coût de résolution du problème du stable maximum à l'aide de DFMAX appliqué aux graphes résultant de l'application de SSD en fonction de la densité espérée des graphes initialement de 100 sommets

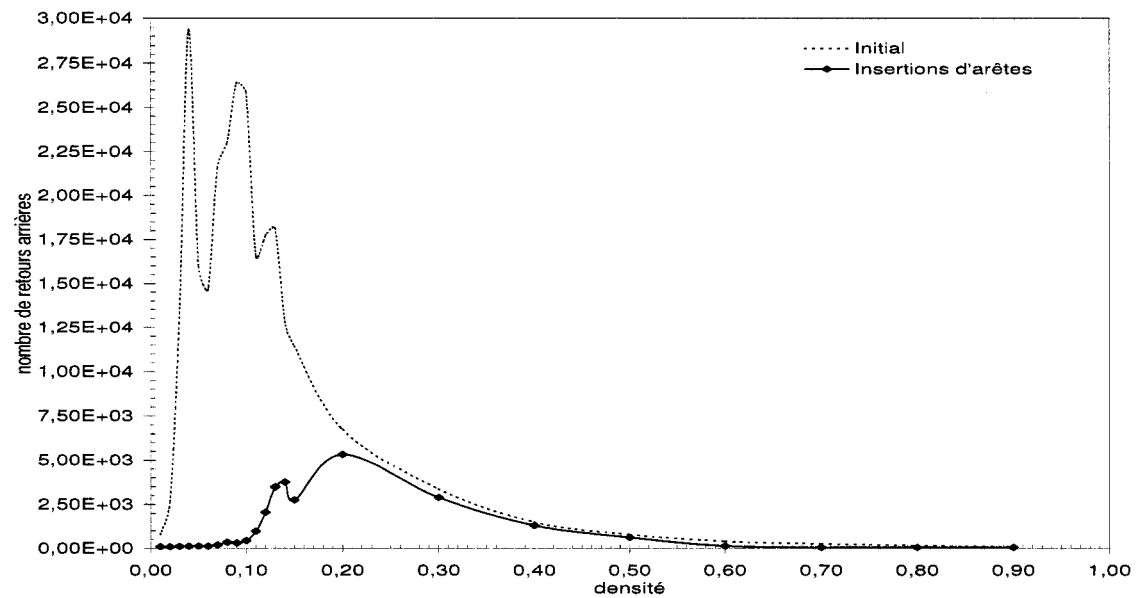


Figure 3.20 Coût de résolution du problème du stable maximum à l'aide de DFMAX appliqué aux graphes résultant de l'application de IATS en fonction de la densité espérée des graphes initialement de 50 sommets

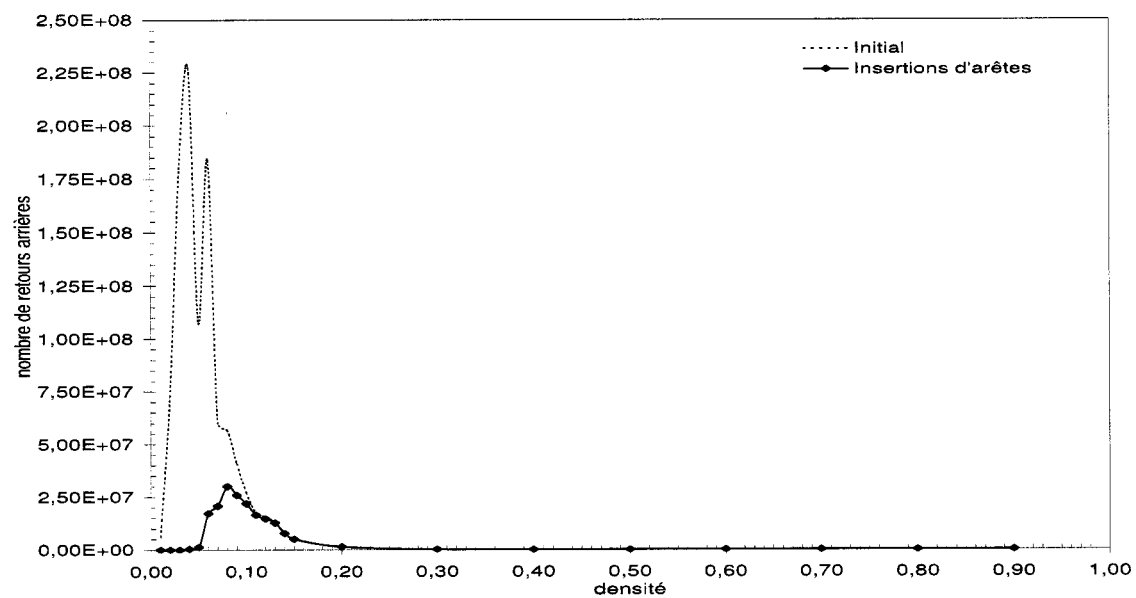


Figure 3.21 Coût de résolution du problème du stable maximum à l'aide de DFMAX appliqué aux graphes résultant de l'application de IATS en fonction de la densité espérée des graphes initialement de 100 sommets

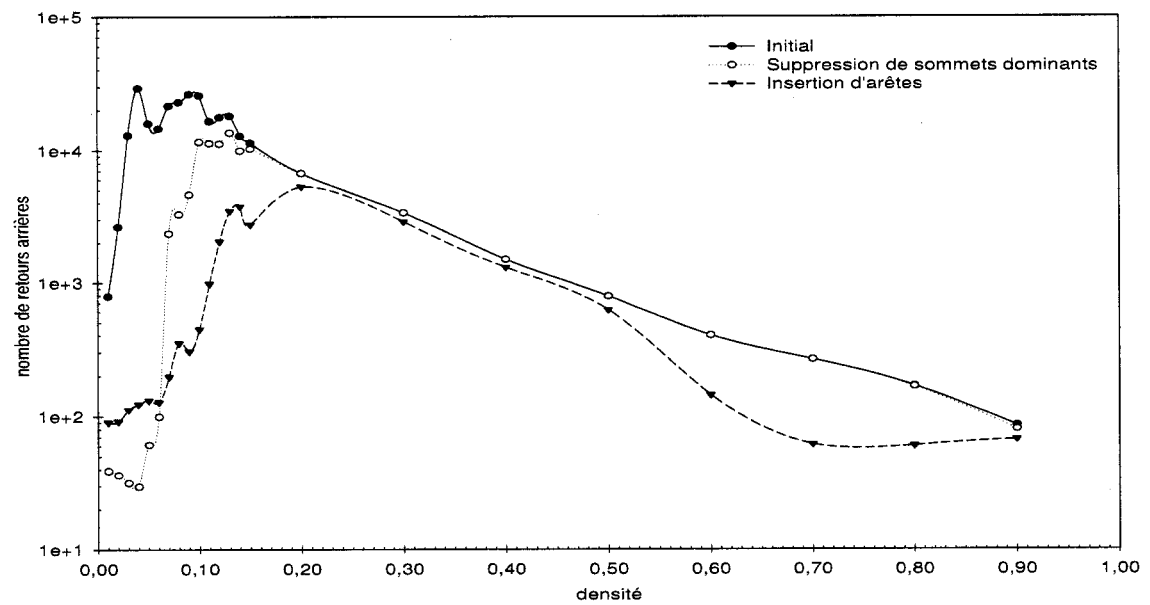


Figure 3.22 Coût de résolution du problème du stable maximum à l'aide de DFMAX appliqué aux graphes initialement de 50 sommets, aux graphes résultant de l'application de SSD et aux graphes résultant de l'application de IATS en fonction de la densité espérée des graphes initialement de 50 sommets

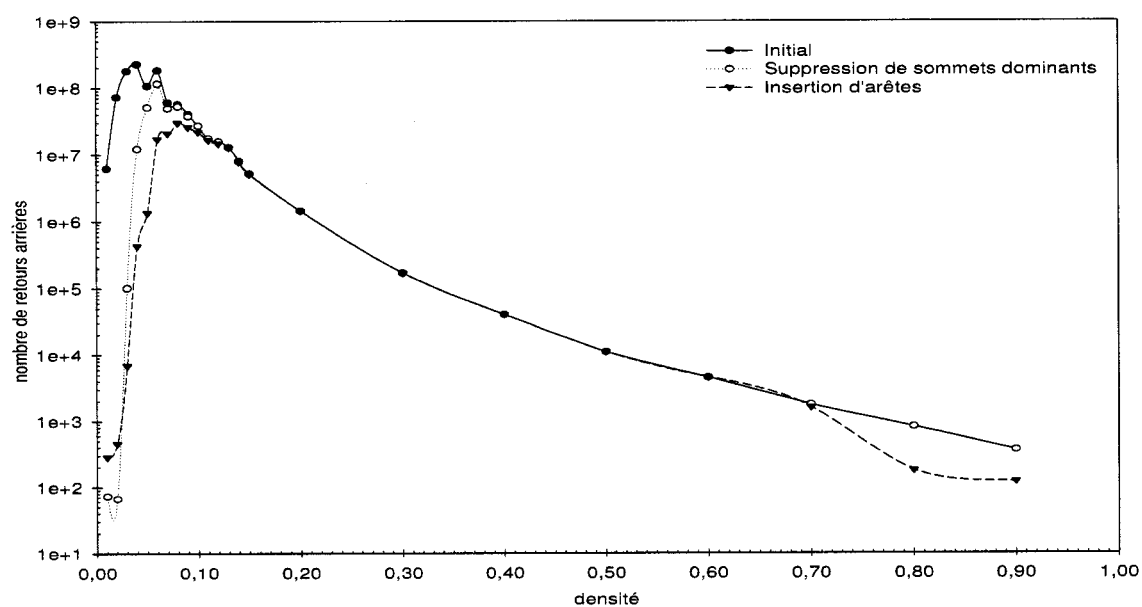


Figure 3.23 Coût de résolution du problème du stable maximum à l'aide de DFMAX appliqué aux graphes initialement de 100 sommets, aux graphes résultant de l'application de SSD et aux graphes résultant de l'application de IATS en fonction de la densité espérée des graphes initialement de 100 sommets

3.5 Discussion

Cette section vise à faire part des conclusions tirées à partir des résultats présentés précédemment et à discuter de la méthode élaborée en début de chapitre.

En premier lieu, SSD, IATS et les séquences de structions inversées seront jugées en fonction de l'objectif de réduction du coût de résolution du problème du stable maximum. Ensuite, quelques faiblesses de la méthode élaborée sont relatées et cette méthode est également analysée en fonction des résultats expérimentaux qu'elle a produits.

3.5.1 Séquence de transformations

Rappelons que l'objectif poursuivi par l'utilisation des transformations avant la résolution du problème du stable maximum est de réduire le coût de la résolution. Ainsi, on dira qu'une séquence de transformations est avantageuse lorsqu'elle atteint cet objectif, c'est-à-dire que $r'_A < 1$, et qu'elle est désavantageuse sinon, c'est-à-dire que $r'_A \geq 1$. Lorsque $r'_A = 1$, on note que la séquence de transformations n'a eu aucune incidence sur le coût de résolution.

On remarque que seules les séquences de transformations ayant des incidences sur la structure des graphes peuvent être avantageuses. Par ailleurs, les séquences de transformations désavantageuses incluent toutes les séquences de transformations n'ayant pas d'incidence sur la structure des graphes et certaines qui en ont une.

Lorsqu'elles ont une incidence sur la structure des graphes initialement de faible densité, SSD et IATS sont avantageuses. En effet, pour tous ces graphes qui ont subi une modification suite à l'application de SSD ou de IATS, on a obtenu un gain réel. Rappelons que selon les estimateurs, le problème du stable maximum

est plus coûteux à résoudre dans les graphes initialement de faible densité. Ainsi, ces deux séquences de transformations semblent avoir un effet avantageux pour les structures problématiques.

Il a été vu aux sous-sections 3.3.2 et 3.3.3 qu'il est possible d'approximer le nombre de sommets dominants et de bons triplets de sommets dans un graphe généré aléatoirement. Bien entendu, ces approximations ne peuvent prédire le nombre de transformations qui seront appliquées, car dès qu'une transformation de la séquence a été appliquée, le graphe résultant diffère du graphe initial. Cependant, lorsqu'il est approximé qu'il n'y a pas de sommets dominants ou de bons triplets de sommets dans les graphes de structure (n, d) , il est probable que respectivement SSD ou IATS n'ait aucune incidence sur la structure d'un graphe de n sommets et de densité d .

Les résultats relatifs à l'incidence de SSD et IATS sur la structure semblent confirmer les résultats prédits. En effet, lorsqu'il est prédit qu'aucun sommet dominant n'est présent dans un graphe d'une structure donnée, il n'y a en général aucune incidence sur cette structure. Il y a de rares exceptions où quelques suppressions de sommets dominants sont effectuées alors qu'il était prévu qu'il n'y avait aucun sommet dominant. Mais ceci n'a pas eu une incidence très importante sur la structure. La même réflexion s'effectue par rapport à IATS.

Notons aussi que sans être quantitativement fiable comme estimation, les nombres calculés de sommets dominants et de bons triplets en fonction de la structure sont près des nombres respectivement de sommets dominants supprimés suite à l'application de SSD et d'arêtes ajoutées suite à l'application de IATS. Une exception à cette observation se trouve au niveau des structures de densité 0,9 pour lesquelles le nombre calculé de sommets dominants est largement supérieur au nombre de sommets dominants supprimés suite à l'application de SSD.

Notons que les remarques sur le nombre de sommets dominants et de bons triplets de sommets s'appliquent en particulier aux graphes générés aléatoirement. Ces calculs ne s'appliquent pas nécessairement aux graphes qui sont générés selon certaines règles, comme les graphes de référence présentés à la sous-section 3.3.6. Pour plusieurs de ces graphes, il est possible d'approximer qu'il n'y a pas de sommets dominants ou de bons triplets de sommets et effectivement, il n'y en a pas. Mais dans d'autres cas, ces approximations ne semblent pas justes.

De plus, il est important de noter qu'une erreur algorithmique s'est glissée dans *sSD*. En effet, l'algorithme correspondant à cette séquence de transformations décrit une procédure telle qu'il faut construire l'ensemble D de tous les sommets dominants possédant au moins un voisin non dominant puis supprimer cet ensemble de sommets. L'algorithme est basé sur le fait que la suppression d'un sommet appartenant à D n'affecte pas la propriété de sommet dominant des autres sommets de D . Cette affirmation n'est toutefois pas vraie.

En effet, soit $G = (V, E)$ un graphe et soient $x \in V$ et $y \in V$ deux sommets dominants tels qu'il existe dans $N(x)$ et dans $N(y)$ au moins un sommet non dominant. Soit $G' = T_{sSD}(G, x)$. Si y n'est pas dominant dans G' , alors pour chaque $u \in N(y)$, on a $N(u) - \{y\} \not\subseteq N(y)$. Ainsi, dans G , y ne domine que x . Mais x est également dominant dans G . Par conséquent, dans G soit x domine un sommet différent de y , soit il domine y . La première option est impossible, car dans ce cas y dominerait également ce sommet. On a donc que dans G , x domine seulement y . Dans ce cas, la suppression de x annule la propriété de y d'être dominant.

Il est possible de déduire la configuration de sommets qui rend l'affirmation fausse. Il s'agit d'une paire de sommets dominants ayant le même voisinage tel que ce voisinage contient au moins un sommet non dominant et que chaque sommet de

ce voisinage est adjacent à au moins un sommet qui n'appartient pas à la paire de sommets dominants. Cependant, cette configuration de sommets est très peu probable dans les classes de graphes que nous avons étudiées.

Soit $G = (V, E)$ un graphe généré aléatoirement de densité espérée p . Posons $n = |V|$. Le nombre de paires de sommets possédant le même voisinage $n_{psv}(G)$ dans G peut être estimé par

$$n_{psv}(G) \approx n \cdot (n - 1) \cdot p \cdot (1 - 2p + 2p^2)^{n-2}.$$

En effectuant les calculs pour les classes de graphes expérimentées, on s'aperçoit que les probabilités que deux sommets adjacents aient le même voisinage est inférieure à 0,16. Cette probabilité maximum est atteinte pour la structure (50, 0, 90). Notons que pour la structure (50, 0, 80), la probabilité est de $1,8 \times 10^{-5}$ et que pour la structure (60, 0, 90), elle est de 0,03.

La probabilité estimée par $n_{psv}(G)$ ne tient pas compte du fait que tout sommet appartenant au voisinage de cette paire de sommets dominants doit posséder au moins un voisin non-adjacent aux sommets de cette paire. Ceci diminue les probabilités. En observant les données sur les graphes initiaux et les graphes résultant de l'application de SSD, il est juste d'affirmer qu'aucun sommet modifiant le nombre de stabilité n'a été supprimé suite à l'application de SSD.

Notons que dans les cas où l'application de ces séquences de transformations engendre une perte réelle, cette perte n'est pas très grande. La plupart du temps, elles sont désavantageuses lorsqu'elles n'ont aucune incidence sur la structure des graphes et donc, aucune incidence sur le coût de résolution du problème du stable maximum.

La struction inversée est chronologiquement la première transformation à avoir été

expérimentée. C'est cette transformation qui a motivé l'élaboration de la méthode présentée en début de chapitre.

Les différents essais de séquences de structions inversées ont montré que, dans tous les cas expérimentés, ce sont des séquences de transformations désavantageuses. En effet, pour tous les graphes ayant subi une modification suite à l'application d'une séquence de structions inversées, on a une perte réelle et pour les autres graphes, il n'y a aucune amélioration.

Notons qu'aucune approximation n'a été calculée par rapport au nombre de structions inversées qui pourraient être appliquées.

3.5.2 Méthode proposée

La méthode présentée à la sous-section 3.1.1 est proposée pour prévoir si l'application d'une transformation peut réduire le coût de résolution du problème du stable maximum. Tel qu'il sera vu à la fin de cette sous-section, elle semble avoir, en général, les résultats escomptés. Néanmoins, elle comprend quelques faiblesses.

Tout d'abord, la distribution de la valeur des coûts de résolution est plutôt dispersée pour les graphes dont la structure est la même mais la densité faible. C'est-à-dire que l'ensemble des valeurs du coût de résolution associées aux dix graphes expérimentés de même structure est réparti sur un intervalle. Cet intervalle est beaucoup plus grand pour les graphes possédant une faible densité. Ceci peut engendrer un biais sur les estimateurs du coût de résolution dans ces graphes.

En second lieu, les estimateurs pourraient ne pas être valides puisque les graphes transformés possèdent une structure particulière. Par structure particulière, il est entendu que ces graphes ne sont pas générés aléatoirement. Par conséquent, toutes

les arêtes du graphe résultant de l'application d'une séquence de transformations n'avaient pas une probabilité égale d'appartenir à ce graphe. Notons que la même remarque s'applique aux graphes de référence de la sous-section 3.3.6.

Enfin, cette méthode ne tient pas compte du coût de l'application de la séquence de transformations aux graphes. Bien qu'il ne soit pas difficile d'inclure cet aspect dans la méthode, ce coût n'a pas été pris en compte puisque les algorithmes implantant les séquences de transformations des graphes n'ont pas été optimisés. Par conséquent, considérer ce coût aurait pu biaiser les conclusions par rapport au possible avantage d'utiliser une séquence de transformations.

Il est vrai qu'une séquence de transformations est réellement avantageuse pour la résolution du problème du stable maximum dans un graphe initial seulement si le coût de son application additionné au coût de résolution dans le graphe résultant est inférieur au coût de résolution dans le graphe initial. Cependant, on remarque que l'application d'une séquence de transformations à un graphe s'effectue en un temps polynomial tandis que la résolution du problème du stable maximum s'effectue en un temps exponentiel. On peut penser que si une séquence de transformations semble avantageuse suite à l'application de notre méthode, alors pour des graphes de grande taille, elle sera réellement avantageuse.

De plus, notons que les coûts d'application de SSD et de IATS ont tout de même été observés. En les observant, on remarque que pour les graphes de 100 sommets, il semble réellement avantageux d'utiliser ces séquences de transformations pour les graphes possédant une faible densité.

Somme toute, les résultats obtenus suite à l'application de la méthode semblent malgré tout convenables.

En effet, DFMAX étant l'algorithme utilisé, il était possible de s'attendre à ce que,

pour une même densité, le coût de résolution augmente avec le nombre de sommets. Les estimateurs reflètent bien ce fait. Au niveau de la densité, les estimateurs montrent que, pour un même nombre de sommets, DFMAX est plus coûteux pour les petites densités, en général inférieures à 0,10.

Les résultats provenant de la résolution du problème du stable maximum dans les graphes résultant de l'application d'une séquence de transformations sont tels qu'ils semblent suivre ce qui était prédit par les estimateurs. Pour une même densité, le coût de résolution croît avec le nombre de sommets. Pour un même nombre de sommets, le coût de résolution est plus élevé pour un graphe de faible densité que pour un graphe de plus grande densité. Rappelons que faible densité fait référence aux densités comprises dans un intervalle $[d_1, d_2]$ près de 0, mais l'excluant.

En général, les estimateurs calculés ne sont pas trop différents des valeurs réelles. Par conséquent, les gains escomptés ne sont pas très différents des gains réels.

Même si la méthode proposée est plutôt approximative, il semble qu'elle arrive à prédire si l'application d'une séquence de transformations produira un gain ou une perte. Il y a eu quelques petites erreurs pour lesquels les rapports diffèrent de moins de 0,25 et leurs valeurs sont proches de 1 et deux plus grandes erreurs: les graphes de structure (50,0,01) et ceux de structure (60,0,01) auxquels on a appliqué IATS. Dans ces cas, une plus grande perte était prédite et un fort gain s'est produit.

Enfin, en général, la méthode a su prédire si la séquence de transformations produirait une perte ou non. Remarquons que dans certain cas, il a été prédit qu'il n'y aurait aucun effet, alors qu'il y en a eu un. L'effet, gain ou perte, est néanmoins assez faible, c'est-à-dire que r'_A est près de 1.

CHAPITRE 4

GRAPHES SÉPARABLES

Ce dernier chapitre traite de la propriété de certains graphes d'être séparables. La première section définit et précise cette propriété. Les deux autres sections présentent quelques classes de graphes séparables et quelques classes de graphes non-séparables.

4.1 Définitions et concepts

Propriété 39 Soit $G = (V, E)$ et une partition $\{V_1, V_2\}$ de V , alors

$$\alpha(G) \leq \alpha(G[V_1]) + \alpha(G[V_2]). \quad (4.1)$$

Cette propriété est valide pour tous les graphes. Cependant, dans le cas où 4.1 est une égalité, il est possible de définir une propriété caractérisant certains graphes.

Définition 40 Le graphe $G = (V, E)$ est dit **séparable** en $V_1 \subseteq V$ et $V_2 \subseteq V$ si $V_1 \neq \emptyset$, $V_2 \neq \emptyset$, $V_1 \cap V_2 = \emptyset$, $V_1 \cup V_2 = V$ et $\alpha(G) = \alpha(G[V_1]) + \alpha(G[V_2])$. Dans ce cas, V_1 et V_2 sont des **parties séparables** de G .

Lorsqu'une des parties séparables d'un graphe induit une clique, il est possible de déduire une transformation exacte telle qu'un graphe résultant de cette transformation possède moins de sommets que le graphe initial. Soit $G = (V, E)$ un graphe et $K \subseteq V$ une clique.

$$\begin{aligned}
& K \text{ est une partie séparable de } G \\
& \Leftrightarrow \alpha(G) = \alpha(G[K]) + \alpha(G[V - K]) \\
& \Leftrightarrow \alpha(G) = 1 + \alpha(G[V - K]) \\
& \Leftrightarrow \alpha(G[V - K]) = \alpha(G) - 1
\end{aligned}$$

Par conséquent, la suppression d'une clique qui est une partie séparable, notée T'_{sK} , consiste en la suppression de tous les sommets appartenant à cette clique.

Pour un graphe $G = (V, E)$ et une clique $K \subseteq V$ qui est une partie séparable de G , le graphe résultant de la transformation spécifique de suppression de la clique K est noté $T'_{sK}(G, K)$.

4.2 Quelques classes de graphes séparables

Dans cette section, quelques classes de graphes séparables sont présentées. Les graphes considérés sont tous supposés connexes. Pour un graphe non connexe $G = (V, E)$, une partie séparable correspond à une de ses composantes connexes, $C = (V_C, E_C)$, et l'autre partie séparable correspond au graphe $G[V - V_C]$.

De plus, notons qu'il est supposé que tous les graphes présentés dans cette section ne sont pas des cliques.

4.2.1 Arbres

Tel que vu à la section 2.5.1, un arbre est un graphe connexe sans cycle.

Proposition 41 *Un arbre est séparable.*

DÉMONSTRATION

Soit $G = (V, E)$ un arbre. Soit $V_1 = \{x, y\}$ où x est une feuille de G et y , l'unique voisin de x . Le sommet y domine le sommet x . Il a été vu à la section 2.5.3 que, dans ce cas, $\alpha(G) = \alpha(G[V - \{y\}])$.

Cependant, dans le graphe $G[V - \{y\}]$, le sommet x est isolé et, par conséquent, $\alpha(G[V - \{x, y\}]) = \alpha(G) - 1$.

Donc, G est séparable, car $\alpha(G[V_1]) = 1$ et $\alpha(G[V - V_1]) = \alpha(G) - 1$. ■

4.2.2 Cycles pairs

Un cycle pair est un cycle, tel que défini au chapitre 1, possédant un nombre pair de sommets.

Proposition 42 *Un cycle pair est séparable.*

DÉMONSTRATION

Soit C_{2k} un cycle de $2k$ sommets. Le nombre de stabilité de C_{2k} est k . Soit $V_1 = \{x, y\}$ où x et y sont adjacents dans C_{2k} . V_1 est une partie séparable de C_{2k} , car $\alpha(G[V_1]) = 1$ et, $G[V - V_1]$ étant une chaîne de longueur paire, $\alpha(G[V - V_1]) = \lceil \frac{2k-2}{2} \rceil = k - 1$. ■

Notons que le cas du cycle impair, qui sera vu à la sous-section 4.3.1, est un contre-exemple à l'affirmation disant qu'un cycle est séparable.

4.2.3 Cycle additionné à une chaîne

La somme cartésienne de graphes (Berge, 1970) se définit en considérant deux graphes $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$. Le graphe $G_1 + G_2$ est un graphe $G = (V, E)$ tel que

- $V = \{[x, y] \mid x \in V_1 \text{ et } y \in V_2\},$
- $E = \{([x, y], [z, w]) \mid (x = z \text{ et } \{y, w\} \in E_2) \text{ ou } (y = w \text{ et } \{x, z\} \in E_1)\}.$

Ainsi, pour additionner les deux graphes G_1 et G_2 , il s'agit de reproduire $|V_2|$ copies de G_1 où chacune représente un sommet de G_2 . Ensuite, chaque sommet u de G_1 est lié à ses copies de telle sorte que le sommet de la i^e copie représente le i^e sommet de G_2 . Donc, si les sommets i et j de G_2 sont adjacents, alors les sommets de la i^e copie de G_1 seront adjacents à leur copie dans la j^e copie de G_1 .

Proposition 43 *Le graphe $G_1 + G_2$ où G_1 est un cycle et G_2 est une chaîne est séparable.*

DÉMONSTRATION

Soit $C_m = (V_1, E_1)$ un cycle de m sommets et $P_n = (V_2, E_2)$ une chaîne de n sommets. Le graphe résultant de la somme cartésienne de ces deux graphes est constitué de n copies de C_m telles que les sommets homologues de la i^e copie et de la $(i + 1)^e$ copie de C_m sont adjacents pour $i = 1, 2, \dots, n - 1$. Le nombre de stabilité de ce graphe est $\alpha(C_m + P_n) = n \cdot \lfloor \frac{m}{2} \rfloor$.

Ainsi, le graphe $C_m + P_n$ est séparable en la n^e copie de C_m et le graphe $C_m + P_{n-1}$, car $\alpha(C_m) = \lfloor \frac{m}{2} \rfloor$ et $\alpha(C_m + P_{n-1}) = (n - 1) \cdot \lfloor \frac{m}{2} \rfloor$. ■

4.2.4 Graphes bipartis

La classe des graphes bipartis contient les arbres et les cycles pairs. Par conséquent, la proposition 47 qui suit généralise les propositions 41 et 42.

Définition 44 Un graphe $G = (V, E)$ est **biparti** si son ensemble de sommets V possède une partition en deux sous-ensembles V_1 et V_2 tels que chaque arête de G relie un sommet de V_1 à un sommet de V_2 .

Rappelons que la cardinalité du plus grand couplage est notée $\kappa(G)$ et que celle du plus grand transversal est noté $\tau(G)$ (chapitre 1).

Théorème 45 (König) Si $G = (V, E)$ est biparti, alors $\kappa(G) = \tau(G)$.

Définition 46 Soit $G = (V, E)$ un graphe et soit C un couplage maximum dans G . Un sommet $v \in V$ est dit **saturé** par le couplage C s'il est incident à une arête appartenant à ce couplage.

Proposition 47 Un graphe biparti est séparable.

DÉMONSTRATION

Soit $G = (V, E)$ un graphe biparti et C un couplage de cardinalité maximum dans G . Posons

$$V_1 = \{x, y\} \cup A \quad \text{et} \quad V_2 = V - V_1$$

où $\{x, y\} \in C$ et $A = \{v \in V \mid (\{x, v\} \in E \text{ ou } \{y, v\} \in E) \text{ et } \{v, u\} \notin C \forall u \in V\}$, c'est-à-dire que A est l'ensemble des sommets adjacents à x ou à y et non saturés.

Considérons les deux sous-graphes induits $G[V_1] = (V_1, E_1)$, $G[V_2] = (V_2, E_2)$ et posons $E' = E_1 \cup E_2$. Soit le sous-graphe partiel $G' = (V, E')$. Par construction, C est également un couplage maximum dans G' , car les arêtes appartenant à $E - E'$ n'appartiennent pas à C .

Posons $C_1 = C \cap E_1$ et $C_2 = C \cap E_2$. On a $C = C_1 \cup C_2$ et ainsi, C_1 et C_2 sont respectivement deux couplages maximums dans G_1 et G_2 puisque C est maximum dans G' et que $E_1 \cap E_2 = \emptyset$. On a donc

$$\kappa(G) = \kappa(G_1) + \kappa(G_2).$$

Par le théorème 45 et puisqu'il a été vu, au chapitre 1, que $\alpha(G) = |V| - \tau(G)$, on en conclut que les graphes bipartis sont séparables. ■

4.2.5 Graphes parfaits

La classe des graphes parfaits contient, entre autres, la classe des graphes bipartis. Par conséquent, la proposition 51 généralise la proposition 47.

Définition 48 Un graphe $G = (V, E)$ est **parfait** si pour tout sous-graphe induit G' de G ,

$$\chi(G') = \omega(G').$$

Proposition 49 Un graphe G est parfait si et seulement si son graphe complémentaire \overline{G} est parfait.

Corollaire 50 *Un graphe G est parfait si pour tout sous-graphe induit G' de G ,*

$$\theta(G') = \alpha(G').$$

Proposition 51 *Un graphe parfait est séparable.*

DÉMONSTRATION

Soit $G = (V, E)$ un graphe parfait. Soit Q un ensemble de cliques couvrant les sommets de G de cardinalité minimum, c'est-à-dire que $|Q| = \theta(G)$, et $K \in Q$ une clique.

Si $\alpha(G[V - K]) + \alpha(G[K]) = \alpha(G)$, alors G est séparable.

Supposons que G n'est pas séparable, alors $\alpha(G[V - K]) + \alpha(G[K]) \geq \alpha(G) + 1$. Puisque l'ensemble de sommets K forme une clique, $\alpha(G[V - K]) \geq \alpha(G)$.

Soit l'ensemble de cliques $Q - \{K\}$ de cardinalité $\theta(G) - 1$. Les sommets de $V - K$ sont couverts par cette famille de cliques. Donc, $\theta(G[V - K]) \leq \theta(G) - 1$. Par le corollaire 50, $\alpha(G[V - K]) \leq \alpha(G) - 1$ ce qui contredit $\alpha(G[V - K]) \geq \alpha(G)$. ■

La classe des graphes parfaits inclut plusieurs autres classes de graphes dont les graphes d'intervalle, les graphes de comparabilité, les graphes de permutation, les graphes parfaitement ordonnables, qui sont donc toutes des classes de graphes séparables. Pour d'autres sous-classes de la classe des graphes parfaits voir (Brandstädt, 2004).

4.2.6 Graphes contenant un point d'articulation

Définition 52 *Soit $G = (V, E)$ un graphe. Un sommet $x \in V$ est un **point d'articulation** si $G[V - \{x\}]$ contient plus d'une composante connexe.*

La théorie, la terminologie et les notations précédant l'énoncé de la proposition 54 ont été tirées d'un article de Mouloud Boulala et Jean-Pierre Uhry (Boulala et Uhry, 1979).

Soit $G = (V, E)$ un graphe contenant un point d'articulation $a \in V$. Supposons qu'un système de poids réels est défini sur les sommets de G . Considérons $G_1 = (V_1, E_1)$ et $G_2 = G[V - V_1 \cup \{a\}] = (V_2, E_2)$ où $a \in V_1$ et $G[V_1 - \{a\}]$ est une composante connexe de $G[V - \{a\}]$.

Notons S_a^1 (respectivement S_a^1) l'ensemble stable de poids maximum sur G_1 qui contient (respectivement qui ne contient pas) le sommet a et notons s_a^1 (respectivement s_a^1) la valeur de cet ensemble stable. De manière générale, les ensembles stables seront notés par des majuscules et leur valeur par une minuscule.

Théorème 53 *Soit S^2 un ensemble stable de poids maximum sur G_2 quand on affecte le poids $\delta_a = s_a^1 - s_a^1$ au sommet a , le poids des autres sommets de G_2 étant inchangé, et soit S l'ensemble stable de G obtenu de la manière suivante*

1. Si $a \in S^2$, $S = S^2 \cup S_a^1$.
2. Si $a \notin S^2$, $S = S^2 \cup S_a^1$.

Alors S est un ensemble stable de poids maximum dans G de valeur $s = s^2 + s_a^1$.

Donc, si le système de poids sur G est tel que chaque sommet a un poids égal à 1 et que S est un ensemble stable de poids maximum dans G , alors $\alpha(G) = s$.

Remarque 10 *Résoudre le problème du stable maximum dans un graphe équivaut à résoudre un problème d'ensemble stable de poids maximum dans le même graphe*

sur lequel on a défini un système de poids tel que chaque sommet a un poids égal à 1.

Proposition 54 *Un graphe contenant un point d'articulation est séparable.*

DÉMONSTRATION

Soit $G = (V, E)$ un graphe contenant un point d'articulation $a \in V$. Munissons G d'un système de poids tel que chaque sommet de V a un poids égal à 1.

Considérons $G_1 = (V_1, E_1)$ et $G_2 = G[V - V_1 \cup \{a\}] = (V_2, E_2)$ où $a \in V_1$ et $G[V_1 - \{a\}]$ est une composante connexe de $G[V - \{a\}]$.

Soit S_a^1 un ensemble stable de poids maximum sur G_1 qui contient le sommet a et soit S_a^1 un ensemble stable de poids maximum sur G_1 qui ne contient pas le sommet a . Posons $\delta_a = s_a^1 - s_a^1$.

Soit S^2 un ensemble stable de poids maximum sur G_2 lorsque la valeur δ_a est affectée au sommet a et soit S l'ensemble stable tel que défini au théorème 53. Par ce théorème, on sait que $s = s^2 + s_a^1$.

Montrons que G est séparable lorsque $\delta_a \leq 0$ et lorsque $\delta_a > 0$ ce qui montre que G est séparable dans tous les cas.

D'une part, si $\delta_a \leq 0$, il est possible de séparer G en G_1 et $G[V - V_1]$.

En effet, on a que $s_a^1 \leq s_a^1$ et ainsi, $\alpha(G_1) = s_a^1$.

De plus, il est possible de construire S^2 tel que $a \notin S^2$ puisque la valeur affectée au sommet a pour trouver S^2 est inférieure ou égale à 0. Supposons que $a \in S^2$. Si la valeur de $S^2 - \{a\}$ était supérieure à celle de S^2 , ce dernier ne serait pas un ensemble stable de poids maximum. Alors la valeur de $S^2 - \{a\}$ est égale à celle de

S^2 et ainsi, puisque S^2 est un ensemble stable de poids maximum, alors $S^2 - \{a\}$ l'est également. On en déduit que $\alpha(G[V - V_1]) = s^2$.

Ainsi, lorsque $\delta_a \leq 0$,

$$\alpha(G) = s = s^2 + s_a^1 = \alpha(G[V - V_1]) + \alpha(G_1).$$

D'autre part, si $\delta_a > 0$, il est possible de séparer G en G_2 et $G[V - V_2]$.

En effet, on a que $\alpha(G[V - V_2]) = \alpha(G[V_1 - \{a\}]) = s_a^1$.

De plus, on a que $\delta_a = 1$. Supposons que $\delta_a > 1$, alors on a que $s_a^1 - 1 > s_a^1$. Cependant, $S_a^1 - \{a\}$ est un ensemble stable ne contenant pas a de valeur $s_a^1 - 1$ ce qui est une contradiction. On en déduit que $\alpha(G_2) = s^2$.

Ainsi, lorsque $\delta_a > 0$,

$$\alpha(G) = s = s^2 + s_a^1 = \alpha(G_2) + \alpha(G[V - V_2]).$$

■

4.2.7 Graphes contenant une clique d'articulation

Définition 55 Soit $G = (V, E)$ un graphe. Un ensemble de sommets $U \subset V$ est un **ensemble d'articulation** si $G[V - U]$ contient plus d'une composante connexe.

Une clique d'articulation dans un graphe G est un ensemble d'articulation U dont le sous-graphe induit $G[U]$ est une clique.

Un sommet étant une clique, la proposition 57 généralise la proposition 54. De plus, le théorème 53 s'étend au cas où un graphe contient une clique d'articulation. Comme à la sous-section 4.2.6, la théorie, la terminologie et les notations précédant l'énoncé de la proposition 57 sont tirées de l'article (Boulala et Uhry, 1979).

Soit $G = (V, E)$ un graphe contenant une clique d'articulation $K \subseteq V$. Supposons qu'un système de poids réels est défini sur les sommets de G . Considérons $G_1 = (V_1, E_1)$ et $G_2 = G[V - V_1 \cup K] = (V_2, E_2)$ où $K \subseteq V_1$ et $G[V_1 - K]$ est une composante connexe de $G[V - K]$.

Notons par S_K^1 l'ensemble stable de poids maximum sur G_1 qui ne contient aucun sommet de K et notons par s_K^1 la valeur de cet ensemble stable. De plus, pour chaque sommet $a \in K$, notons S_a^1 l'ensemble stable de poids maximum sur G_1 qui contient a et s_a^1 la valeur de cet ensemble stable. Comme à la sous-section 4.2.6, les ensembles stables seront notés par des majuscules et leur valeur par une minuscule.

Théorème 56 *Soit S^2 l'ensemble stable de poids maximum sur G_2 quand on affecte le poids $\delta_a = s_a^1 - s_K^1$ à chaque sommet $a \in K$, le poids des autres sommets de G_2 étant inchangé, et soit S l'ensemble stable de G obtenu de la manière suivante*

1. Si $S^2 \cap K = \{a\}$, $S = S^2 \cup S_a^1$.
2. Si $S^2 \cap K = \emptyset$, $S = S^2 \cup S_K^1$.

Alors S est un ensemble stable de poids maximum dans G de valeur $s = s^2 + s_K^1$.

Rappelons la remarque 10 qui permet de conclure que si le système de poids sur G est tel que chaque sommet a un poids égal à 1 et que S est un ensemble de poids maximum dans G , alors $\alpha(G) = s$.

Proposition 57 *Un graphe contenant une clique d'articulation est séparable.*

DÉMONSTRATION

Soit $G = (V, E)$ un graphe contenant une clique d'articulation $K \subseteq V$. Munissons G d'un système de poids tel que chaque sommet de V a un poids égal à 1.

Considérons $G_1 = (V_1, E_1)$ et $G_2 = G[V - V_1 \cup K] = (V_2, E_2)$ où $K \subseteq V_1$ et $G[V_1 - K]$ est une composante connexe de $G[V - K]$.

Considérons pour chaque sommet $a \in K$, S_a^1 un ensemble stable de poids maximum sur G_1 qui contient le sommet $a \in K$ et soit S_K^1 un ensemble stable de poids maximum sur G_1 qui ne contient aucun sommet de K . Pour chaque $a \in K$, posons $\delta_a = s_a^1 - s_K^1$.

Soit S^2 un ensemble stable de poids maximum sur G_2 lorsque la valeur δ_a est affectée au sommet a pour chaque sommet $a \in K$ et soit S l'ensemble stable tel que défini au théorème 56. Par ce théorème, on sait que $s = s^2 + s_K^1$.

Montrons qu'il est possible de séparer G en

$$G[V_1 - \{v \in K \mid \delta_v > 0\}] \text{ et } G[V_2 - \{v \in K \mid \delta_v \leq 0\}].$$

En effet, d'une part, on a que $s_a^1 \leq s_K^1$ pour tout $a \in (V_1 - \{v \in K \mid \delta_v > 0\}) \cap K$. Ainsi, $\alpha(G[V_1 - \{v \in K \mid \delta_v > 0\}]) = s_K^1$.

D'autre part, montrons tout d'abord que pour chaque sommet $x \in \{v \in K \mid \delta_v > 0\}$, $\delta_x = 1$.

Supposons qu'il existe un $a \in K$ tel que $\delta_a > 1$, alors on a que $s_a^1 - 1 > s_K^1$. Cependant, $S_a^1 - \{a\}$ est un ensemble stable ne contenant aucun sommet appartenant à

K de valeur $s_a^1 - 1$, ce qui entraîne une contradiction.

Il est maintenant possible de construire S^2 tel que $S^2 \cap \{v \in K \mid \delta_v \leq 0\} = \emptyset$. Supposons qu'il existe $a \in \{v \in K \mid \delta_v \leq 0\}$ tel que $a \in S^2$. Si la valeur de $S^2 - \{a\}$ était supérieure à celle de S^2 , ce dernier ne serait pas un ensemble stable de poids maximum. Alors la valeur de $S^2 - \{a\}$ est égale à celle de S^2 et ainsi, puisque S^2 est un ensemble stable de poids maximum, alors $S^2 - \{a\}$ l'est également. On en déduit que

$$\alpha(G[V_2 - \{v \in K \mid \delta_v \leq 0\}]) = s^2.$$

On en conclut que

$$\alpha(G) = s = s^2 + s_K^1 = \alpha(G[V_2 - \{v \in K \mid \delta_v \leq 0\}]) + \alpha(G[V_1 - \{v \in K \mid \delta_v > 0\}]).$$

■

Définition 58 *Un graphe est **triangulé** si tous ses cycles de plus de trois sommets contiennent au moins une corde.*

Théorème 59 *(Berge, 1970) Un graphe connexe est triangulé si et seulement si tout ensemble d'articulation minimal induit une clique dans ce graphe.*

La proposition 57 permet donc de déduire que les graphes triangulés contenant au moins un ensemble d'articulation sont séparables.

4.2.8 Graphes série-parallèles

La théorie sous-jacente à la démonstration de la proposition 63 de même que la terminologie et les notations sont tirées de l'article (Boulala et Uhry, 1979) utilisé

également dans les sous-sections 4.2.6 et 4.2.7.

Définition 60 *Un graphe $G = (V, E)$ est dit **série-parallèle** s'il ne contient pas de sous-graphe partiel homéomorphe à la clique à quatre sommets.*

Un graphe est homéomorphe à K_4 , la clique à quatre sommets, si K_4 est obtenue de ce graphe en lui appliquant itérativement l'ajout d'une arête entre les voisins d'un sommet de degré 2 suivi de la suppression de ce sommet de degré 2 jusqu'à ce qu'il n'y ait plus de sommets de degré 2.

Un multigraphe est un graphe dont deux sommets peuvent être joints par plus d'une arête. Un graphe simple est un multigraphe, mais l'inverse n'est évidemment pas vrai. Les transformations qui suivent s'appliquent aux multigraphes.

Considérons deux transformations T_1 et T_2 où T_1 consiste à supprimer un sommet de degré 2 et à ajouter une arête entre ses deux voisins et T_2 consiste à supprimer une arête double entre deux sommets et à ajouter une arête simple entre ces sommets.

Plus précisément, pour un graphe $G = (V, E)$ et un sommet $x \in V$ dont l'ensemble de voisins est $\{y, z\} \subseteq V$, le graphe résultant de la suppression du sommet x et de l'ajout de l'arête $\{y, z\}$ est noté $T_1(G, x)$. Pour un graphe $G = (V, E)$ et pour deux sommets $x \in V$ et $y \in V$ tels que x et y sont joints par une arête double, le graphe résultant de la suppression de l'arête double entre x et y et de l'ajout de l'arête $\{x, y\}$ est noté $T_2(G, x, y)$.

Propriété 61 (Boulala et Uhry, 1979) *Soit $G = (V, E)$ un graphe série-parallèle connexe sans point d'articulation. On peut trouver une suite de transformations T_1 et T_2 appliquées à G de telle sorte que le graphe résultant soit une arête double.*

Considérons T_1^{-1} et T_2^{-1} les deux transformations respectivement réciproque de T_1 et T_2 . C'est-à-dire que T_1^{-1} consiste à supprimer une arête et à ajouter un sommet adjacent aux sommets incidents à l'arête supprimée et que T_2^{-1} consiste à supprimer une arête simple et à ajouter une arête double entre les sommets incidents à l'arête supprimée.

Plus précisément, pour un graphe $G = (V, E)$ et une arête $\{x, y\} \in E$, le graphe résultant de la suppression de l'arête $\{x, y\}$ et de l'ajout d'un sommet voisin de x et y est noté $T_1^{-1}(G, \{x, y\})$. Pour un graphe $G = (V, E)$ et une arête simple $\{x, y\} \in E$, le graphe résultant de la suppression de l'arête $\{x, y\}$ et de l'ajout d'une arête double entre x et y est noté $T_2^{-1}(G, \{x, y\})$.

Proposition 62 *Soit $G = (V, E)$ un graphe tel que V contient deux sommets et ces deux sommets sont joints par une arête double. De l'application itérative des transformations T_1^{-1} et T_2^{-1} résulte un graphe série-parallèle sans point d'articulation.*

Proposition 63 *Un graphe série-parallèle est séparable.*

DÉMONSTRATION

Soit $G = (V, E)$ un graphe série-parallèle.

Si G contient un point d'articulation, alors, par la proposition 54, G est séparable.

Sinon, par la propriété 61, on sait que l'application itérative des transformations T_1 et T_2 produit un graphe à deux sommets joints par une arête double.

En appliquant itérativement la transformation T_1 à G , il est possible d'obtenir une arête double d'extrémités $a' \in V'$ et $b' \in V'$ dans le graphe résultant $G' = (V', E')$. Les sommets $a \in V$ et $b \in V$ correspondant respectivement à $a' \in V'$ et $b' \in V'$

représentent un ensemble d'articulation à deux sommets dans G . De plus, l'arête double d'extrémité a' et b' dans G' correspond à deux chaînes d'extrémité a et b dans G . Notons $G_1 = (V_1, E_1)$ le sous-graphe de G induit par les sommets contenus dans une de ces chaînes et $G_2 = (V_2, E_2)$ le sous-graphe de G induit par les sommets contenus dans l'autre chaîne.

Il est possible de distinguer deux cas par rapport à la parité du nombre de sommets de G_1 et de G_2 :

- G_1 et G_2 possèdent tous deux un nombre pair de sommets,
- G_1 ou G_2 possède un nombre impair de sommets.

Afin de démontrer que G est séparable, montrons que dans ces deux cas G est séparable.

- G_1 et G_2 possèdent tous deux un nombre pair de sommets.

Si $V_1 = \{a, b\}$ ou $V_2 = \{a, b\}$, alors $\{a, b\}$ est une clique d'articulation dans G et, par la proposition 57, on déduit que G est séparable.

Sinon, montrons que G est séparable en $G[V_1 - \{a, b\}]$ et $G[V - V_1 \cup \{a, b\}]$.

Soit $x \in V_1 - \{a, b\}$ tel que $\{a, x\} \in E$ et soit $y \in V_1 - \{a, b\}$ tel que $\{b, y\} \in E$. On a que $G[V_1 - \{a, b\}]$ est une chaîne contenant un nombre pair de sommets. Notons S_{xy}^1 l'ensemble stable maximum dans $G[V_1 - \{a, b\}]$ contenant x et ne contenant pas y et S_{xy}^1 l'ensemble stable maximum dans $G[V_1 - \{a, b\}]$ ne contenant pas x et contenant y .

Il s'agit maintenant de construire un ensemble stable maximum dans $G[V - V_1 \cup \{a, b\}]$ contenant au plus un des sommets de l'ensemble $\{a, b\}$.

Soit S^2 un ensemble stable maximum dans $G[V - V_1 \cup \{a, b\}]$.

Si S^2 est tel que $a \notin S^2$ ou $b \notin S^2$, alors G est séparable.

En effet, d'une part si $a \notin S^2$, on a que $S^2 \cup S_{xy}^1 = S$ est un ensemble stable dans G et ainsi $|S| = |S^2 \cup S_{xy}^1| = |S^2| + |S_{xy}^1| \leq \alpha(G)$.

Donc $\alpha(G[V - V_1 \cup \{a, b\}]) + \alpha(G[V_1 - \{a, b\}]) \leq \alpha(G)$. Mais par la propriété 39, on a $\alpha(G[V - V_1 \cup \{a, b\}]) + \alpha(G[V_1 - \{a, b\}]) \geq \alpha(G)$. Par conséquent, $\alpha(G[V - V_1 \cup \{a, b\}]) + \alpha(G[V_1 - \{a, b\}]) = \alpha(G)$.

D'autre part, si $b \notin S^2$, on a que $S^2 \cup S_{xy}^1 = S$ est un ensemble stable dans G et, par le même raisonnement que pour $a \notin S^2$, on conclut que $\alpha(G[V - V_1 \cup \{a, b\}]) + \alpha(G[V_1 - \{a, b\}]) = \alpha(G)$.

Sinon, on a que $a \in S^2$ et $b \in S^2$. Cependant, par définition de G_2 , on sait qu'il existe une chaîne de a à b contenant un nombre pair de sommets telle que chaque sommet de cette chaîne, excluant a et b , est de degré 2 dans G et donc dans $G[V - V_1 \cup \{a, b\}]$.

Notons $S_{ab}^{G_2}$ le stable maximum dans G_2 contenant a et ne contenant pas b . On a que $S^2 - (S^2 \cap V_2) \cup S_{ab}^{G_2}$ est aussi un ensemble stable maximum dans $G[V - V_1 \cup \{a, b\}]$ et b n'appartient pas à cet ensemble stable. Donc, par le même raisonnement que lorsque $b \notin S^2$, on conclut que dans ce cas aussi G est séparable.

Ainsi, si G_1 et G_2 possèdent tous deux un nombre pair de sommets, G est séparable.

- G_1 ou G_2 possède un nombre impair de sommets.

Si $V_1 = \{a, b\}$ ou $V_2 = \{a, b\}$, alors $\{a, b\}$ est une clique d'articulation dans G et, par la proposition 57, on déduit que G est séparable.

Sinon, soit $G' = (V', E')$ où $V' = V_1 \cup V_2$ et $E' = E_1 \cup E_2$. Montrons par construction que G est séparable en G' et $G[V - V'] = G'' = (V'', E'')$.

G' est un cycle sans corde induit de G . Puisque $\{a, b\} \notin E'$ et qu'il existe un chemin entre a et b contenant un nombre impair de sommets dans G' , il existe un ensemble stable maximum S' dans G' tel que $a \notin S'$ et $b \notin S'$.

Puisque $E - (E' \cup E'') = \{\{x, y\} \in E \mid x \in \{a, b\} \text{ et } y \in V''\}$, on a que $S' \cup S''$ est un ensemble stable dans G quel que soit S'' l'ensemble stable maximum dans G'' . Donc, $|S| = |S' \cup S''| = |S'| \cup |S''| \leq \alpha(G)$ d'où $\alpha(G') + \alpha(G'') \leq \alpha(G)$. Mais par la propriété 39, $\alpha(G') + \alpha(G'') \geq \alpha(G)$.

Ainsi, si G_1 ou G_2 possède un nombre impair de sommets, G est séparable. ■

4.3 Quelques classes de graphes non-séparables

Les graphes non-séparables sont ceux qui ne possèdent aucun ensemble séparable. Les cliques sont un exemple trivial de graphes non-séparables. $\alpha(G) = 1$ pour tout graphe $G = (K, E)$ tel que l'ensemble des sommets de G forme une clique et $\alpha(G_1) + \alpha(G_2) = 2$ pour tout $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$ tels que $V_1 \neq \emptyset$, $V_2 \neq \emptyset$, $V_1 \cup V_2 = V$ et $V_1 \cap V_2 = \emptyset$.

4.3.1 Cycles impairs

Un cycle impair est un cycle possédant un nombre impair de sommets.

Proposition 64 *Un cycle impair est non-séparable.*

DÉMONSTRATION

La suppression d'une arête ne peut diminuer le nombre de stabilité d'un graphe. Par contre, elle peut l'augmenter d'une unité. Pour effectuer une séparation d'un cycle impair, un minimum de deux arêtes doivent être supprimées. Dans ce cas, le graphe est séparé en deux chaînes.

Considérons une séparation de G en deux chaînes. Une de ces chaînes P_i contient i sommets où $1 \leq i \leq 2k$ et l'autre P_j contient $j = 2k + 1 - i$ sommets. Une seule chaîne P_i ou P_j possède un nombre pair de sommets. Sans perte de généralité, supposons que $G[V_1] = P_i$ possède un nombre pair de sommets et que $G[V_2] = P_j$ en possède un nombre impair. Par conséquent, $\alpha(G[V_1]) = \frac{i}{2}$, $\alpha(G[V_2]) = \frac{(2k+1-i)+1}{2} = k + 1 - \frac{i}{2}$ et

$$\alpha(G[V_1]) + \alpha(G[V_2]) = \alpha(G) + 1 = k + 1.$$

On en conclut qu'un cycle impair n'est pas séparable. ■

4.3.2 Roues impaires

Une roue impaire est un graphe formé d'un cycle impair et d'un sommet adjacent à tous les sommets de ce cycle.

Lemme 65 *Soit un graphe $G = (V, E)$ avec $|V| > 2$ et x un sommet tel que $(x, u) \in E$ pour tout $u \in V - \{x\}$. Alors $\alpha(G) = \alpha(G[V - \{x\}])$.*

DÉMONSTRATION

Soit S un ensemble stable dans le graphe $G = (V, E)$ avec $|V| > 2$ et soit $x \in V$ un sommet adjacent à tous les autres sommets de V . Si $x \in S$, alors $|S| = 1$. Ainsi, puisque $\alpha(G) \geq 1$ pour tout graphe G , il est possible de trouver un ensemble stable S ne contenant pas x . ■

Proposition 66 *Une roue impaire est non-séparable.*

DÉMONSTRATION

Soit $G = (V, E)$ une roue impaire de $2k + 2$ sommets et soit $x \in V$ le sommet adjacent à chaque sommet $v \in V - \{x\}$. Ainsi, $\alpha(G) = \alpha(G[V - \{x\}])$ par le lemme 65.

Supposons que G soit séparable. Alors,

$$\alpha(G) = \alpha(G[V_1]) + \alpha(G[V_2]) \text{ avec } V_1 \cup V_2 = V \text{ et } V_1 \cap V_2 = \emptyset.$$

Sans perte de généralité, il est possible de supposer que $x \in V_1$. Par conséquent, x est adjacent à tous les autres sommets de V_1 donc $\alpha(G[V_1]) = \alpha(G[V_1 - \{x\}])$. On en conclut que $\alpha(G[V - \{x\}]) = \alpha(G[V_1 - \{x\}]) + \alpha(G[V_2])$ ce qui signifie que les cycles impairs sont séparables. Il y a contradiction, d'où on conclut que les roues impaires ne sont pas séparables. ■

4.3.3 Anti-trous

Définition 67 *Un **anti-trou** est le graphe complémentaire d'un cycle impair sans corde d'au moins cinq sommets.*

Proposition 68 *Un anti-trou n'est pas séparable.*

DÉMONSTRATION

Supposons qu'un anti-trou $G = (V, E)$ soit séparable. Ainsi,

$$\alpha(G[V_1]) + \alpha(G[V_2]) = \alpha(G) \text{ avec } V_1 \cup V_2 = V \text{ et } V_1 \cap V_2 = \emptyset.$$

\overline{G} , le graphe complémentaire de G , est un cycle impair. La plus grande clique dans \overline{G} est de cardinalité égale à 2, ce qui implique que $\alpha(G) = 2$. Par conséquent, pour qualifier G de séparable, il faut que $\alpha(G[V_1]) = \alpha(G[V_2]) = 1$. C'est donc dire que chacune des deux parties séparables de G est une clique.

G est séparable en deux cliques, alors \overline{G} est coloriable en deux couleurs. Cependant, un cycle impair est coloriable en un minimum de trois couleurs et \overline{G} étant un cycle impair, il ne peut être coloré en deux couleurs. On en conclut qu'un anti-trou n'est pas séparable. ■

CONCLUSION

Le premier objectif de ce travail était d'étudier les transformations de graphes et leurs effets sur le nombre de stabilité, plus précisément, de tenter de déterminer si l'application d'une transformation particulière réussit à réduire le coût de résolution du problème du stable maximum.

À cette fin, une étude théorique sur l'incidence des transformations de graphes sur le nombre de stabilité a été effectuée. Une transformation est qualifiée d'exacte lorsque son impact sur le nombre de stabilité est connu et constant indépendamment du graphe auquel elle est appliquée. Dans le cas contraire, elle est qualifiée d'in-exacte. Pour chacun des deux types de transformations, une stratégie les utilisant dans la résolution du problème du stable maximum a été proposée. Puis, des exemples de transformations exactes et inexactes ont été présentés.

Nous nous sommes davantage intéressés à l'utilisation des transformations exactes et, plus précisément, à l'utilisation de la suppression d'un sommet dominant, de l'insertion d'une arête lorsqu'il y a un bon triplet de sommets et de la struction inversée.

L'objectif a ensuite été poursuivi de manière expérimentale pour les trois transformations précédemment énumérées. Une méthode permettant d'évaluer leur incidence sur le coût de résolution du problème du stable a été élaborée. Cette méthode consiste en premier lieu à construire un ensemble d'estimateurs du coût de résolution du problème du stable maximum en fonction de la structure des graphes. Ensuite, étant donné un graphe, il s'agit de transformer ce graphe et d'observer l'impact sur sa structure. Enfin, le coût de résolution dans le graphe transformé est estimé puis calculé afin de déterminer le gain escompté et le gain

réel. Cette méthode se veut également un outil empirique permettant d'estimer si une transformation a des chances d'atteindre l'objectif de son utilisation ou non. Elle arrivera à estimer cela en comparant le gain escompté et le gain réel.

Une séquence de transformations est une suite de transformations appliquée selon des règles précises. Les trois transformations étudiées ont toutes été utilisées dans au moins une séquence de transformations.

Selon les résultats obtenus expérimentalement, la séquence de suppressions d'un sommet dominant et la séquence d'insertions d'une arête lorsqu'il y a un bon triplet de sommets semblent avantageuses dans certains cas. En général, pour les graphes de faible densité, il semble que l'emploi d'une de ces séquences préalablement à la résolution du problème du stable maximum diminue le coût de résolution de ce problème. Au contraire, toutes les séquences de structions inversées expérimentées se sont avérées désavantageuses qu'elles aient été appliquées seules ou dans une suite de séquences.

L'ensemble des résultats recueillis dans le cadre de ce travail semble indiquer qu'une transformation réduisant le nombre de sommets ou augmentant de façon considérable le nombre d'arêtes serait prometteuse, c'est-à-dire qu'une telle transformation pourrait avoir toutes les chances de réduire le coût de résolution du problème du stable maximum.

De plus, sous toute réserve, la méthode élaborée semble fiable. En effet, elle a su prédire l'avantage ou le désavantage de l'utilisation d'une transformation avant la résolution du problème du stable maximum dans tous les cas sauf un. Dans cet unique cas, elle a prédit que l'utilisation de la séquence d'insertion d'une arête lorsqu'il y a un bon triplet de sommets ne réduirait pas le coût de résolution, alors qu'en réalité, elle l'a réduit.

Le second objectif était d'amorcer l'étude des graphes séparables et non-séparables. Ce but a été poursuivi en déterminant si les graphes appartenant à certaines classes sont séparables ou non.

Comme il est possible de s'en douter à la lecture de ce mémoire, le sujet des transformations de graphes est aussi vaste que peu exploré. La plupart des recherches explorent une seule transformation et parfois limitent son application à une classe de graphes. Un article traitant d'une façon de formaliser les transformations préservant le nombre de stabilité a été publié depuis peu (Lozin, 2003). Expérimentalement, ce formalisme pourrait permettre la génération automatisée de transformations exactes. Ainsi, par exemple, il pourrait être possible de laisser le programme créer une ou des transformations causant une incidence choisie sur la structure des graphes.

Plus près de notre étude, nous avons choisi une manière d'appliquer les transformations sous forme de séquences. Cependant, deux séquences peuvent utiliser la même transformation mais être différentes car elles ont des critères de choix des paramètres ou un critère d'arrêt différents. Ainsi, avec les transformations que nous avons expérimentées, il est possible d'effectuer d'autres expériences avec des séquences différentes. Les incidences sur la structure des graphes et sur le coût de résolution pourraient également être différentes.

De plus, nous avons très peu exploré la combinaison de transformations différentes. Nous avons utilisé des séquences de transformations qui utilisent une seule transformation générique. Il pourrait être intéressant d'élaborer des séquences de transformation utilisant deux transformations génériques différentes.

Ensuite, pour déterminer de façon plus sûre si la séquence de suppressions d'un sommet dominant et celle d'insertions d'une arête lorsqu'il y a un bon triplet de

sommets sont avantageuses à utiliser avant la résolution du problème du stable maximum, les algorithmes correspondant à ces séquences pourraient être optimisés. Un fois optimisé, il serait possible d'inclure leur coût de résolution dans le calcul du gain réel.

Finalement, la sujet des graphes séparables en est au début de son étude. Par conséquent, plusieurs idées peuvent être développées. Entres autres, la classification des classes de graphes séparables peut être poursuivie en étudiant d'autres classes de graphes. Ensuite, une ou des transformations pourraient être développées à partir de cette propriété. Enfin, la propriété des graphes séparables pourrait être utilisée dans une approche "diviser pour régner " dans le but de résoudre le problème du stable maximum.

RÉFÉRENCES

- ALEKSEEV, V.E., LOZIN, V.V. (2004). Local transformations of graphs preserving independence number. *Discrete Applied Mathematics*. 135:1-3. P.17-30.
- BILLIONNET, A. (1981). Reductions and optimality conditions in the problem of a maximum weighted stable . *RAIRO Recherche Opérationnelle*. 15:3. P.213-231.
- BERGE, C. (1970). *Graphes et Hypergraphes*. Paris : Dunod. 502p.
- BOMZE, I.M., BUDINICH, M., PARDALOS, P.M., PELLILO, M. (1999). The maximum clique problem . *Handbook of Combinatorial Optimization - Suppl Vol. A*. D.Z. Du et P.M. Pardalos (eds). Boston, MA : Kluwer Academic Publishers. P.1-74.
- BOMZE, I.M., BUDINICH, M., PELLILO, M., ROSSI, C. (2002). Annealed replication: a new heuristic for the maximum clique problem . *Discrete Applied Mathematics*. 121:1-3. P.27-49.
- BOULALA, M., UHRY, J.-P. (1979). Polytope des indépendants d'un graphe série-parallèle . *Discrete Mathematics*. 27:3. P.225-243.
- BRANDSTÄDT, A., KNORR, S., LE, V.B., DE RIDDER, H.N., SZYMCAK, T., SIEGEMUND, F. et al. (2004). *Information System on Graph Class Inclusions*. Version 2.0. [Logiciel en ligne]. Rostock, Allemagne. <http://www.teo.informatik.uni-rostock.de/iscgi/>. (Page consultée le 15 novembre 2004)
- BUTZ, L., HAMMER, P.L., HAUSSMAN, D. (1982). Reduction methods for the vertex packing problem . *Proceedings of the 17th Conference on Probability Theory (Brasov 1982)*. Utrecht, Hollande : VNU Science Press. P.73-79.

- BUTENKO, S. (2003). *Maximum independent set and related problems, with applications*. 167p. Thèse de doctorat en mathématiques, Université de Floride.
- CARRAGHAN, R., PARDALOS, P.M. (1990). An exact algorithm for the maximum clique problem . *Operations Research Letters*. 9:6. P.375-382.
- DIRAC, G.A. (1961). On rigid circuit graphs . *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*. 25. P.71-76.
- DUFRESNE, K. (2002). *Projet sur les graphes et la recherche du nombre de stabilité*. [Disponible sur demande] Montréal, Canada. karine.dufresne@polymtl.ca. (Adresse valide le 2 février 2005)
- EBENEGGER, C., HAMMER, P.L., DE WERRA, D. (1984). Pseudo-boolean function and stability of graphs . *Annal of Discrete Mathematics*. 19. P.83-98.
- FRIDEN, C., HERTZ A., DE WERRA, D. (1989). STABULUS : a technique for finding stable sets in large graphs with tabu search . *Computing*. 42:1. P.35-44.
- GAREY, M.R., JOHNSON, D.S. (1979). *Computers and Intractability: A Guide to the Theory of NP-completeness*. New York : Freeman and Company. 340p.
- GERBER, M.U., HERTZ, A. (2001). A transformation which preserves the clique number . *Journal of Combinatorial Theory*. Serie B 83:2. P.320-330.
- HERTZ, A. (1997). On the use of boolean methods for the computation of the stability number . *Discrete Applied Mathematics*. 76:1-3. P.183-203.
- LOZIN, V.V. (2003). Stability preserving transformations of graphs . *Manuscript*. 13p.
- ÖSTERGÅRD, P.R.J. (2002). A fast algorithm for the maximum clique problem . *Discrete Applied Mathematics*. 120:1-3. P.197-207.

RÉGIN, J.-C. (2003). Solving the maximum clique problem with constraint programming . *Proceedings of the 5th International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (Montréal, 2003)* [En ligne]. <http://www.crt.umontreal.ca/cpaor/article-regin.pdf> (Page consulté le 15 février 2005)

TRICK, M. (1993). The Second DIMACS Challenge. In *Site de Carnegie Mellon University*. [En ligne]. <http://mat.gsia.cmu.edu/challenge.html> (Page consultée le 15 novembre 2004)

TRICK, M. (1993). The Second DIMACS Challenge. In *Site FTP de Rutgers, The State University of New Jersey*. [En ligne]. <ftp://dimacs.rutgers.edu/pub/challenge/> (Page consultée le 15 novembre 2004)

ZISSIMOPOULOS, V. (1995). On the performance guarantee of neural networks for NP-hard optimization problems . *Information Processing Letters*. 54 : 6. P.317-322.

ANNEXE I

TABLEAUX DE RÉSULTATS

À l'exception des résultats concernant l'application de SI2 sur les graphes de 50 sommets, tous les résultats recueillis lors des expérimentations décrites au chapitre 3 sont présentés sous forme de tableaux dans cet annexe. Les résultats concernant SI2 se trouvent à la page 105.

Pour connaître la signification des notations voir le chapitre 3 traitant du volet expérimental de ce travail.

Les deux premiers tableaux présentent les estimateurs du coût de résolution du problème du stable maximum à l'aide de DFMAX en fonction de la structure. Le premier contient ces estimateurs mesurés en nombre de retours arrières tandis que le second contient ceux mesurés en secondes.

Les tableaux I.3 à I.8 contiennent les résultats concernant l'application de SSD et les tableaux I.9 à I.14 contiennent les résultats concernant l'application de IATS. Les tableaux I.15 à I.20 et I.21 à I.26 contiennent respectivement les résultats relatifs à l'application de SI1 et ceux relatif à l'application de SI1 suivie de SSD. Finalement, les quatre derniers tableaux présentent les résultats relatifs à l'application de SIALEA1, à l'application de SIALEA1 suivie de SI1, à l'application de SIALEA2 et à l'application de SIALEA2 suivie de SI2.

Notons que lorsque “–” apparaît dans la colonne C_A , c'est que la résolution du problème du stable maximum prenait plus de 12 heures. Dans quelques cas “–” apparaît mais la résolution n'a pas été exécutée pendant 12 heures, car la

résolution dans le graphe initial ne prenait que quelques secondes. Par conséquent, il était inutile de laisser la résolution s'exécuter pendant plus d'une heure. Lorsque “- ” apparaît pour $c_{\mathcal{A}}(n(G'), d(G'))$, c'est qu'il n'y avait pas les estimateurs nécessaires à son calcul.

Tableau I.1 Estimateurs du coût de résolution du problème du stable maximum par DFMAX mesurés en nombre de retours arrières en fonction de la structure

		n					
		50	60	70	80	90	100
d	0, 01	790	6234	26930	261714	6959784	6133703
	0, 02	2641	20855	173204	1840429	8291193	72691527
	0, 03	13012	119958	743828	4148619	33210408	179676999
	0, 04	29365	192322	1087281	8419912	56971590	226993732
	0, 05	15954	135178	897844	5907759	38630596	107466454
	0, 06	14626	104846	518004	4285648	27568744	184115891
	0, 07	21570	161882	575447	4145824	15390722	60215965
	0, 08	23007	96172	556621	3142247	17115151	56777757
	0, 09	26392	138300	551998	2846343	8620442	40842801
	0, 10	25833	99495	478176	2125148	9096797	27153709
	0, 11	18438	90063	383572	1573204	5677460	18665748
	0, 12	19001	80606	291351	1196517	3724009	16307787
	0, 13	18143	78432	247291	1033081	3431334	12937927
	0, 14	12849	50975	216050	801176	2474981	7966531
	0, 15	11430	45868	169035	491399	1646192	5139322
	0, 20	6722	24750	69111	188201	576251	1437171
	0, 30	3378	8589	19642	45094	92735	170086
	0, 40	1495	3094	6181	11764	22605	40357
	0, 50	789	1558	2646	4378	7988	10988
	0, 60	399	798	1320	1926	3040	4526
	0, 70	265	428	626	953	1227	1759
	0, 80	168	255	356	487	655	817
	0, 90	85	118	170	217	279	360

Tableau I.2 Estimateurs du coût de résolution du problème du stable maximum par DFMAX mesurés en secondes en fonction de la structure

		n					
		50	60	70	80	90	100
d	0, 01	0, 00	0, 01	0, 03	0, 24	6, 37	6, 19
	0, 02	0, 00	0, 01	0, 14	1, 51	7, 25	70, 31
	0, 03	0, 01	0, 08	0, 51	3, 12	26, 52	153, 01
	0, 04	0, 01	0, 11	0, 68	5, 44	38, 66	166, 71
	0, 05	0, 01	0, 08	0, 52	3, 58	24, 28	73, 52
	0, 06	0, 01	0, 06	0, 30	2, 44	16, 54	113, 80
	0, 07	0, 01	0, 08	0, 31	2, 22	9, 00	36, 26
	0, 08	0, 01	0, 05	0, 29	1, 63	9, 07	32, 09
	0, 09	0, 01	0, 06	0, 26	1, 39	4, 44	21, 89
	0, 10	0, 01	0, 05	0, 22	1, 00	4, 48	14, 30
	0, 11	0, 01	0, 04	0, 17	0, 72	2, 76	9, 43
	0, 12	0, 01	0, 03	0, 13	0, 53	1, 73	7, 61
	0, 13	0, 01	0, 03	0, 11	0, 45	1, 54	5, 82
	0, 14	0, 01	0, 02	0, 09	0, 34	1, 09	3, 59
	0, 15	0, 00	0, 02	0, 07	0, 21	0, 72	2, 29
	0, 20	0, 00	0, 01	0, 03	0, 07	0, 22	0, 55
	0, 30	0, 00	0, 01	0, 01	0, 02	0, 03	0, 06
	0, 40	0, 00	0, 00	0, 00	0, 01	0, 01	0, 01
	0, 50	0, 00	0, 00	0, 00	0, 00	0, 00	0, 01
	0, 60	0, 00	0, 00	0, 00	0, 00	0, 00	0, 00
	0, 70	0, 00	0, 00	0, 00	0, 00	0, 00	0, 00
	0, 80	0, 00	0, 00	0, 00	0, 00	0, 00	0, 00
	0, 90	0, 00	0, 00	0, 00	0, 00	0, 00	0, 00

Tableau I.3 Incidence de SSD sur la résolution du problème du stable maximum dans des graphes initialement de 50 sommets

Structure initiale $(n(G), d(G))$	Structure résultante $((n(G'), d(G')))$	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(50, 0, 01)	(41, 0, 00)	43	41	0,05	0,05	0,00
(50, 0, 02)	(39, 0, 00)	51	39	0,02	0,02	0,00
(50, 0, 03)	(36, 0, 01)	77	—	—	0,01	0,00
(50, 0, 04)	(38, 0, 02)	240	—	—	0,01	0,00
(50, 0, 05)	(40, 0, 03)	575	—	—	0,04	0,00
(50, 0, 06)	(42, 0, 05)	2110	—	—	0,14	0,00
(50, 0, 07)	(45, 0, 07)	5058	—	—	0,23	0,00
(50, 0, 08)	(45, 0, 07)	6377	—	—	0,28	0,00
(50, 0, 09)	(45, 0, 09)	6410	—	—	0,24	0,00
(50, 0, 10)	(47, 0, 09)	13609	—	—	0,53	0,01
(50, 0, 11)	(48, 0, 10)	12912	—	—	0,78	0,01
(50, 0, 12)	(48, 0, 12)	11396	—	—	0,64	0,01
(50, 0, 13)	(48, 0, 12)	13742	—	—	0,76	0,01
(50, 0, 14)	(49, 0, 14)	10099	—	—	0,79	0,00
(50, 0, 15)	(49, 0, 15)	10298	—	—	0,90	0,00
(50, 0, 20)	(50, 0, 20)	6722	6722	1,00	1,00	0,00
(50, 0, 30)	(50, 0, 30)	3378	3378	1,00	1,00	0,00
(50, 0, 40)	(50, 0, 41)	1495	1424	0,95	1,00	0,00
(50, 0, 50)	(50, 0, 50)	789	789	1,00	1,00	0,00
(50, 0, 60)	(50, 0, 60)	399	399	1,00	1,00	0,00
(50, 0, 70)	(50, 0, 70)	265	265	1,00	1,00	0,00
(50, 0, 80)	(50, 0, 79)	166	177	1,06	0,99	0,00
(50, 0, 90)	(46, 0, 88)	80	—	—	0,94	0,00

Tableau I.4 Incidence de SSD sur la résolution du problème du stable maximum dans des graphes initialement de 60 sommets

Structure initiale ($n(G), d(G)$)	Structure résultante ($(n(G'), d(G'))$)	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(60, 0, 01)	(48, 0, 00)	51	48	0,01	0,01	0,00
(60, 0, 02)	(46, 0, 00)	69	46	0,00	0,00	0,00
(60, 0, 03)	(46, 0, 02)	235	—	—	0,00	0,00
(60, 0, 04)	(48, 0, 03)	1836	—	—	0,01	0,00
(60, 0, 05)	(51, 0, 04)	7305	45661	0,34	0,05	0,01
(60, 0, 06)	(53, 0, 05)	19753	51721	0,49	0,19	0,01
(60, 0, 07)	(55, 0, 07)	61378	91726	0,57	0,38	0,03
(60, 0, 08)	(56, 0, 08)	33482	66906	0,70	0,35	0,02
(60, 0, 09)	(58, 0, 09)	77731	115918	0,84	0,56	0,03
(60, 0, 10)	(59, 0, 10)	82184	92129	0,93	0,83	0,04
(60, 0, 11)	(59, 0, 11)	77527	84483	0,92	0,84	0,03
(60, 0, 12)	(60, 0, 12)	84450	86874	1,00	0,97	0,03
(60, 0, 13)	(60, 0, 13)	73515	78432	1,00	0,94	0,03
(60, 0, 14)	(60, 0, 14)	50975	50975	1,00	1,00	0,02
(60, 0, 15)	(60, 0, 15)	44713	45868	1,00	0,97	0,02
(60, 0, 20)	(60, 0, 20)	24750	24750	1,00	1,00	0,01
(60, 0, 30)	(60, 0, 30)	8589	8589	1,00	1,00	0,00
(60, 0, 40)	(60, 0, 41)	3094	2940	0,95	1,00	0,00
(60, 0, 50)	(60, 0, 50)	1558	1558	1,00	1,00	0,00
(60, 0, 60)	(60, 0, 59)	798	874	1,09	1,00	0,00
(60, 0, 70)	(60, 0, 70)	428	428	1,00	1,00	0,00
(60, 0, 80)	(60, 0, 80)	255	255	1,00	1,00	0,00
(60, 0, 90)	(58, 0, 90)	117	111	0,94	0,99	0,00

Tableau I.5 Incidence de SSD sur la résolution du problème du stable maximum dans des graphes initialement de 70 sommets

Structure initiale $(n(G), d(G))$	Structure résultante $((n(G'), d(G'))$	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(70, 0, 01)	(55, 0, 00)	69	55	0,00	0,00	0,00
(70, 0, 02)	(52, 0, 01)	97	1879	0,01	0,00	0,00
(70, 0, 03)	(55, 0, 02)	3327	11748	0,02	0,00	0,00
(70, 0, 04)	(58, 0, 03)	28505	98569	0,09	0,03	0,02
(70, 0, 05)	(63, 0, 05)	128755	363978	0,41	0,14	0,08
(70, 0, 06)	(64, 0, 06)	143190	270109	0,52	0,28	0,08
(70, 0, 07)	(67, 0, 07)	316153	451378	0,78	0,55	0,17
(70, 0, 08)	(68, 0, 08)	418070	464531	0,83	0,75	0,21
(70, 0, 09)	(69, 0, 09)	446313	510628	0,93	0,81	0,20
(70, 0, 10)	(69, 0, 10)	445724	440308	0,92	0,93	0,20
(70, 0, 11)	(70, 0, 11)	388581	390816	1,00	0,99	0,17
(70, 0, 12)	(69, 0, 12)	255960	256011	0,93	0,93	0,11
(70, 0, 13)	(70, 0, 13)	247291	247291	1,00	1,00	0,10
(70, 0, 14)	(70, 0, 14)	216050	216050	1,00	1,00	0,09
(70, 0, 15)	(70, 0, 15)	169035	169035	1,00	1,00	0,07
(70, 0, 20)	(70, 0, 21)	69111	64164	0,93	1,00	0,03
(70, 0, 30)	(70, 0, 30)	19642	19642	1,00	1,00	0,01
(70, 0, 40)	(70, 0, 41)	6181	5828	0,94	1,00	0,00
(70, 0, 50)	(70, 0, 50)	2646	2646	1,00	1,00	0,00
(70, 0, 60)	(70, 0, 59)	1320	1453	1,10	1,00	0,01
(70, 0, 70)	(70, 0, 70)	626	626	1,00	1,00	0,00
(70, 0, 80)	(70, 0, 79)	356	383	1,07	1,00	0,01
(70, 0, 90)	(68, 0, 90)	168	160	0,94	0,99	0,00

Tableau I.6 Incidence de SSD sur la résolution du problème du stable maximum sur des graphes initialement de 80 sommets

Structure initiale ($n(G), d(G)$)	Structure résultante ($(n(G'), d(G'))$)	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(80, 0, 01)	(62, 0, 00)	83	62	0,00	0,00	0,00
(80, 0, 02)	(59, 0, 01)	239	5690	0,00	0,00	0,00
(80, 0, 03)	(62, 0, 02)	23636	51325	0,01	0,01	0,02
(80, 0, 04)	(69, 0, 03)	162583	681441	0,08	0,02	0,12
(80, 0, 05)	(72, 0, 05)	844436	1899827	0,32	0,14	0,54
(80, 0, 06)	(76, 0, 06)	2004537	2778590	0,65	0,47	1,16
(80, 0, 07)	(77, 0, 07)	2131900	3074711	0,74	0,51	1,19
(80, 0, 08)	(79, 0, 08)	2478695	2883684	0,92	0,79	1,31
(80, 0, 09)	(79, 0, 09)	2284682	2616909	0,92	0,80	1,12
(80, 0, 10)	(79, 0, 10)	2048735	1960451	0,92	0,96	0,95
(80, 0, 11)	(80, 0, 11)	1560431	1610731	1,00	0,97	0,69
(80, 0, 12)	(80, 0, 12)	1140180	1185804	1,00	0,96	0,49
(80, 0, 13)	(80, 0, 13)	981483	1033081	1,00	0,95	0,42
(80, 0, 14)	(80, 0, 14)	795037	801176	1,00	0,99	0,33
(80, 0, 15)	(80, 0, 15)	491399	491399	1,00	1,00	0,20
(80, 0, 20)	(80, 0, 20)	188201	188201	1,00	1,00	0,07
(80, 0, 30)	(80, 0, 30)	45094	45094	1,00	1,00	0,02
(80, 0, 40)	(80, 0, 40)	11764	11764	1,00	1,00	0,01
(80, 0, 50)	(80, 0, 50)	4378	4378	1,00	1,00	0,00
(80, 0, 60)	(80, 0, 60)	1926	1926	1,00	1,00	0,00
(80, 0, 70)	(80, 0, 70)	953	953	1,00	1,00	0,00
(80, 0, 80)	(80, 0, 79)	487	534	1,10	1,00	0,00
(80, 0, 90)	(79, 0, 90)	219	212	0,98	1,01	0,00

Tableau I.7 Incidence de SSD sur la résolution du problème du stable maximum dans des graphes initialement de 90 sommets

Structure initiale ($n(G), d(G)$)	Structure résultante ($(n(G'), d(G'))$)	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(90, 0, 01)	(69, 0, 00)	100	69	0, 00	0, 00	0, 00
(90, 0, 02)	(70, 0, 01)	5026	26930	0, 00	0, 00	0, 01
(90, 0, 03)	(75, 0, 02)	217911	1006817	0, 03	0, 01	0, 18
(90, 0, 04)	(82, 0, 04)	3797569	18130248	0, 32	0, 07	2, 79
(90, 0, 05)	(85, 0, 05)	11793010	22269178	0, 58	0, 31	7, 76
(90, 0, 06)	(88, 0, 06)	13685095	22912125	0, 83	0, 50	8, 54
(90, 0, 07)	(89, 0, 07)	12811353	14266232	0, 93	0, 83	7, 59
(90, 0, 08)	(89, 0, 08)	15293356	15717861	0, 92	0, 89	8, 11
(90, 0, 09)	(90, 0, 09)	8605773	8620442	1, 00	1, 00	4, 42
(90, 0, 10)	(90, 0, 10)	9021425	9096797	1, 00	0, 99	4, 38
(90, 0, 11)	(90, 0, 11)	5446874	5585349	1, 00	0, 98	2, 58
(90, 0, 12)	(90, 0, 12)	3438099	3450419	1, 00	1, 00	1, 57
(90, 0, 13)	(90, 0, 13)	3431334	3431334	1, 00	1, 00	1, 52
(90, 0, 14)	(90, 0, 14)	2474981	2474981	1, 00	1, 00	1, 07
(90, 0, 15)	(90, 0, 15)	1646192	1646192	1, 00	1, 00	0, 70
(90, 0, 20)	(90, 0, 20)	576251	576251	1, 00	1, 00	0, 21
(90, 0, 30)	(90, 0, 30)	92735	92735	1, 00	1, 00	0, 03
(90, 0, 40)	(90, 0, 40)	22605	22605	1, 00	1, 00	0, 01
(90, 0, 50)	(90, 0, 50)	7988	7988	1, 00	1, 00	0, 00
(90, 0, 60)	(90, 0, 60)	3040	3040	1, 00	1, 00	0, 00
(90, 0, 70)	(90, 0, 70)	1227	1227	1, 00	1, 00	0, 00
(90, 0, 80)	(90, 0, 80)	655	655	1, 00	1, 00	0, 00
(90, 0, 90)	(89, 0, 90)	279	273	0, 98	1, 00	0, 00

Tableau I.8 Incidence de SSD sur la résolution du problème du stable maximum dans des graphes initialement de 100 sommets

Structure initiale $(n(G), d(G))$	Structure résultante $((n(G'), d(G'))$	Coût en nombre de retours arrières				Coût en secondes
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	C_A
(100, 0, 01)	(77, 0, 00)	118	77	0,00	0,00	0,00
(100, 0, 02)	(77, 0, 01)	5330	191279	0,00	0,00	0,01
(100, 0, 03)	(87, 0, 03)	5135645	24491871	0,14	0,03	4,69
(100, 0, 04)	(93, 0, 04)	34144156	107978233	0,48	0,15	25,82
(100, 0, 05)	(96, 0, 05)	58094312	79932111	0,74	0,54	41,32
(100, 0, 06)	(98, 0, 06)	117257550	152806462	0,83	0,64	73,15
(100, 0, 07)	(99, 0, 07)	50686702	55733441	0,93	0,84	30,87
(100, 0, 08)	(100, 0, 08)	54108328	56777757	1,00	0,95	30,49
(100, 0, 09)	(100, 0, 09)	37604155	40842801	1,00	0,92	20,06
(100, 0, 10)	(100, 0, 10)	27153709	27153709	1,00	1,00	14,18
(100, 0, 11)	(100, 0, 11)	17407123	17407123	1,00	1,00	8,60
(100, 0, 12)	(100, 0, 12)	15668795	15668795	1,00	1,00	7,28
(100, 0, 13)	(100, 0, 13)	12937927	12937927	1,00	1,00	5,76
(100, 0, 14)	(100, 0, 14)	7966531	7966531	1,00	1,00	3,55
(100, 0, 15)	(100, 0, 15)	5139322	5139322	1,00	1,00	2,26
(100, 0, 20)	(100, 0, 20)	1437171	1437171	1,00	1,00	0,54
(100, 0, 30)	(100, 0, 30)	170086	170086	1,00	1,00	0,06
(100, 0, 40)	(100, 0, 40)	40357	40357	1,00	1,00	0,01
(100, 0, 50)	(100, 0, 50)	10988	10988	1,00	1,00	0,01
(100, 0, 60)	(100, 0, 60)	4526	4526	1,00	1,00	0,00
(100, 0, 70)	(100, 0, 70)	1759	1759	1,00	1,00	0,00
(100, 0, 80)	(100, 0, 80)	817	817	1,00	1,00	0,00
(100, 0, 90)	(100, 0, 90)	360	360	1,00	1,00	0,00

Tableau I.9 Incidence de IATS sur la résolution du problème du stable maximum dans des graphes initialement de 50 sommets

Structure initiale ($n(G), d(G)$)	Structure résultante ($(n(G'), d(G'))$)	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(50, 0, 01)	(50, 0, 30)	91	3378	4,28	0,11	0,00
(50, 0, 02)	(50, 0, 37)	92	2060	0,78	0,03	0,00
(50, 0, 03)	(50, 0, 46)	112	1071	0,08	0,01	0,00
(50, 0, 04)	(50, 0, 45)	123	1142	0,04	0,00	0,00
(50, 0, 05)	(50, 0, 43)	132	1283	0,08	0,01	0,00
(50, 0, 06)	(50, 0, 47)	128	1001	0,07	0,01	0,00
(50, 0, 07)	(50, 0, 39)	198	1683	0,08	0,01	0,00
(50, 0, 08)	(50, 0, 37)	351	2060	0,09	0,02	0,00
(50, 0, 09)	(50, 0, 39)	305	1683	0,06	0,01	0,00
(50, 0, 10)	(50, 0, 34)	445	2625	0,10	0,02	0,00
(50, 0, 11)	(50, 0, 32)	981	2813	0,17	0,06	0,00
(50, 0, 12)	(50, 0, 29)	2055	3378	0,19	0,12	0,00
(50, 0, 13)	(50, 0, 27)	3489	4381	0,24	0,19	0,00
(50, 0, 14)	(50, 0, 29)	3762	3712	0,29	0,29	0,00
(50, 0, 15)	(50, 0, 27)	2743	4381	0,38	0,24	0,00
(50, 0, 20)	(50, 0, 23)	5339	5719	0,85	0,79	0,00
(50, 0, 30)	(50, 0, 31)	2891	3190	0,94	0,86	0,00
(50, 0, 40)	(50, 0, 42)	1305	1354	0,91	0,87	0,00
(50, 0, 50)	(50, 0, 54)	622	633	0,80	0,79	0,00
(50, 0, 60)	(50, 0, 78)	143	187	0,47	0,36	0,00
(50, 0, 70)	(50, 0, 89)	61	93	0,35	0,23	0,00
(50, 0, 80)	(50, 0, 94)	60	71	0,42	0,36	0,00
(50, 0, 90)	(50, 0, 97)	67	61	0,71	0,79	0,00

Tableau I.10 Incidence de IATS sur la résolution du problème du stable maximum dans des graphes initialement de 60 sommets

Structure initiale ($n(G), d(G)$)	Structure résultante ($(n(G'), d(G'))$)	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(60, 0, 01)	(60, 0, 31)	112	8040	1, 29	0, 02	0, 00
(60, 0, 02)	(60, 0, 37)	153	4743	0, 23	0, 01	0, 00
(60, 0, 03)	(60, 0, 42)	155	2787	0, 02	0, 00	0, 00
(60, 0, 04)	(60, 0, 40)	213	3094	0, 02	0, 00	0, 00
(60, 0, 05)	(60, 0, 40)	266	3094	0, 02	0, 00	0, 00
(60, 0, 06)	(60, 0, 35)	436	5842	0, 06	0, 00	0, 00
(60, 0, 07)	(60, 0, 31)	2053	8040	0, 05	0, 01	0, 01
(60, 0, 08)	(60, 0, 27)	3046	13437	0, 14	0, 03	0, 00
(60, 0, 09)	(60, 0, 25)	14744	16670	0, 12	0, 11	0, 01
(60, 0, 10)	(60, 0, 23)	17857	19902	0, 20	0, 18	0, 01
(60, 0, 11)	(60, 0, 21)	16444	23134	0, 25	0, 18	0, 01
(60, 0, 12)	(60, 0, 18)	27695	28974	0, 33	0, 32	0, 01
(60, 0, 13)	(60, 0, 19)	39709	26862	0, 34	0, 51	0, 02
(60, 0, 14)	(60, 0, 18)	32982	28974	0, 57	0, 65	0, 01
(60, 0, 15)	(60, 0, 18)	26311	28974	0, 63	0, 57	0, 01
(60, 0, 20)	(60, 0, 21)	23800	23134	0, 93	0, 96	0, 01
(60, 0, 30)	(60, 0, 30)	8563	8589	1, 00	1, 00	0, 00
(60, 0, 40)	(60, 0, 41)	2945	2940	0, 95	0, 95	0, 00
(60, 0, 50)	(60, 0, 51)	1505	1482	0, 95	0, 97	0, 01
(60, 0, 60)	(60, 0, 64)	608	650	0, 81	0, 76	0, 00
(60, 0, 70)	(60, 0, 88)	91	145	0, 34	0, 21	0, 00
(60, 0, 80)	(60, 0, 94)	73	95	0, 37	0, 29	0, 00
(60, 0, 90)	(60, 0, 97)	79	77	0, 65	0, 67	0, 00

Tableau I.11 Incidence de IATS sur la résolution du problème du stable maximum dans des graphes initialement de 70 sommets

Structure initiale $(n(G), d(G))$	Structure résultante $((n(G'), d(G'))$	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(70, 0, 01)	(70, 0, 32)	150	16950	0, 63	0, 01	0, 00
(70, 0, 02)	(70, 0, 39)	186	7527	0, 04	0, 00	0, 00
(70, 0, 03)	(70, 0, 45)	236	4414	0, 01	0, 00	0, 00
(70, 0, 04)	(70, 0, 39)	244	7527	0, 01	0, 00	0, 00
(70, 0, 05)	(70, 0, 30)	1100	19642	0, 02	0, 00	0, 00
(70, 0, 06)	(70, 0, 33)	1000	15604	0, 03	0, 00	0, 00
(70, 0, 07)	(70, 0, 20)	12662	69111	0, 12	0, 02	0, 01
(70, 0, 08)	(70, 0, 19)	27179	79103	0, 14	0, 05	0, 02
(70, 0, 09)	(70, 0, 18)	81772	89096	0, 16	0, 15	0, 04
(70, 0, 10)	(70, 0, 14)	264671	216050	0, 45	0, 55	0, 12
(70, 0, 11)	(70, 0, 13)	167348	247291	0, 63	0, 43	0, 08
(70, 0, 12)	(70, 0, 17)	116140	99088	0, 36	0, 42	0, 05
(70, 0, 13)	(70, 0, 14)	146705	216050	0, 87	0, 59	0, 06
(70, 0, 14)	(70, 0, 15)	131330	169035	0, 78	0, 61	0, 05
(70, 0, 15)	(70, 0, 15)	140014	169035	1, 00	0, 83	0, 05
(70, 0, 20)	(70, 0, 21)	65888	64164	0, 93	0, 95	0, 02
(70, 0, 30)	(70, 0, 30)	19894	19642	1, 00	1, 01	0, 01
(70, 0, 40)	(70, 0, 41)	5692	5828	0, 94	0, 92	0, 00
(70, 0, 50)	(70, 0, 50)	2760	2646	1, 00	1, 04	0, 00
(70, 0, 60)	(70, 0, 61)	1213	1251	0, 95	0, 92	0, 00
(70, 0, 70)	(70, 0, 86)	185	244	0, 39	0, 30	0, 00
(70, 0, 80)	(70, 0, 93)	86	140	0, 39	0, 24	0, 00
(70, 0, 90)	(70, 0, 97)	91	100	0, 59	0, 53	0, 00

Tableau I.12 Incidence de IATS sur la résolution du problème du stable maximum dans des graphes initialement de 80 sommets

Structure initiale ($n(G), d(G)$)	Structure résultante ($(n(G'), d(G'))$)	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(80, 0, 01)	(80, 0, 37)	192	21763	0,08	0,00	0,00
(80, 0, 02)	(80, 0, 42)	245	10287	0,01	0,00	0,01
(80, 0, 03)	(80, 0, 46)	273	7332	0,00	0,00	0,00
(80, 0, 04)	(80, 0, 37)	895	21763	0,00	0,00	0,00
(80, 0, 05)	(80, 0, 33)	6025	35095	0,01	0,00	0,00
(80, 0, 06)	(80, 0, 19)	182410	218521	0,05	0,04	0,11
(80, 0, 07)	(80, 0, 18)	164684	248841	0,06	0,04	0,10
(80, 0, 08)	(80, 0, 15)	189267	491399	0,16	0,06	0,11
(80, 0, 09)	(80, 0, 13)	982119	1033081	0,36	0,35	0,48
(80, 0, 10)	(80, 0, 12)	1486813	1185804	0,56	0,70	0,70
(80, 0, 11)	(80, 0, 13)	869690	1033081	0,64	0,54	0,39
(80, 0, 12)	(80, 0, 13)	790049	1033081	0,87	0,67	0,35
(80, 0, 13)	(80, 0, 14)	746304	801176	0,78	0,72	0,32
(80, 0, 14)	(80, 0, 14)	602900	801176	1,00	0,75	0,25
(80, 0, 15)	(80, 0, 15)	443051	491399	1,00	0,90	0,18
(80, 0, 20)	(80, 0, 21)	211693	173890	0,92	1,12	0,08
(80, 0, 30)	(80, 0, 30)	40837	45094	1,00	0,91	0,01
(80, 0, 40)	(80, 0, 40)	11468	11764	1,00	0,97	0,01
(80, 0, 50)	(80, 0, 50)	4362	4378	1,00	1,00	0,00
(80, 0, 60)	(80, 0, 60)	1933	1926	1,00	1,00	0,00
(80, 0, 70)	(80, 0, 77)	629	627	0,66	0,66	0,00
(80, 0, 80)	(80, 0, 93)	99	176	0,36	0,20	0,00
(80, 0, 90)	(80, 0, 97)	101	121	0,56	0,47	0,00

Tableau I.13 Incidence de IATS sur la résolution du problème du stable maximum dans des graphes initialement de 90 sommets

Structure initiale $(n(G), d(G))$	Structure résultante $((n(G'), d(G'))$	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(90, 0, 01)	(90, 0, 38)	233	36631	0,01	0,00	0,00
(90, 0, 02)	(90, 0, 35)	604	57670	0,01	0,00	0,00
(90, 0, 03)	(90, 0, 37)	536	43644	0,00	0,00	0,00
(90, 0, 04)	(90, 0, 26)	6125	286141	0,01	0,00	0,01
(90, 0, 05)	(90, 0, 23)	76436	431196	0,01	0,00	0,06
(90, 0, 06)	(90, 0, 13)	1227311	3431334	0,12	0,04	0,78
(90, 0, 07)	(90, 0, 12)	1308041	3450419	0,22	0,08	0,84
(90, 0, 08)	(90, 0, 13)	4326259	3431334	0,20	0,25	2,41
(90, 0, 09)	(90, 0, 11)	5736274	5585349	0,65	0,67	2,98
(90, 0, 10)	(90, 0, 10)	7760709	9096797	1,00	0,85	3,88
(90, 0, 11)	(90, 0, 13)	4188999	3431334	0,61	0,75	1,97
(90, 0, 12)	(90, 0, 13)	3001811	3431334	0,99	0,87	1,38
(90, 0, 13)	(90, 0, 13)	3981681	3431334	1,00	1,16	1,75
(90, 0, 14)	(90, 0, 14)	2176888	2474981	1,00	0,88	0,96
(90, 0, 15)	(90, 0, 15)	1716108	1646192	1,00	1,04	0,71
(90, 0, 20)	(90, 0, 20)	636191	576251	1,00	1,10	0,23
(90, 0, 30)	(90, 0, 30)	93575	92735	1,00	1,01	0,03
(90, 0, 40)	(90, 0, 40)	24486	22605	1,00	1,08	0,01
(90, 0, 50)	(90, 0, 50)	8104	7988	1,00	1,01	0,00
(90, 0, 60)	(90, 0, 60)	3166	3040	1,00	1,04	0,00
(90, 0, 70)	(90, 0, 73)	978	1055	0,86	0,80	0,00
(90, 0, 80)	(90, 0, 93)	120	222	0,34	0,18	0,00
(90, 0, 90)	(90, 0, 97)	107	147	0,53	0,38	0,00

Tableau I.14 Incidence de IATS sur la résolution du problème du stable maximum dans des graphes initialement de 100 sommets

Structure initiale ($n(G), d(G)$)	Structure résultante ($(n(G'), d(G'))$)	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(100, 0, 01)	(100, 0, 35)	286	105222	0,02	0,00	0,001
(100, 0, 02)	(100, 0, 41)	458	37420	0,00	0,00	0,001
(100, 0, 03)	(100, 0, 30)	6936	170086	0,00	0,00	0,008
(100, 0, 04)	(100, 0, 21)	433291	1310463	0,01	0,00	0,365
(100, 0, 05)	(100, 0, 17)	1342572	2547816	0,02	0,01	1,002
(100, 0, 06)	(100, 0, 12)	17194663	15668795	0,09	0,09	11,287
(100, 0, 07)	(100, 0, 10)	20751826	27153709	0,45	0,34	12,737
(100, 0, 08)	(100, 0, 10)	30034886	27153709	0,48	0,53	17,077
(100, 0, 09)	(100, 0, 10)	25914130	27153709	0,66	0,63	14,014
(100, 0, 10)	(100, 0, 10)	21813013	27153709	1,00	0,80	11,476
(100, 0, 11)	(100, 0, 11)	16451060	17407123	1,00	0,95	8,173
(100, 0, 12)	(100, 0, 12)	14642565	15668795	1,00	0,93	6,836
(100, 0, 13)	(100, 0, 13)	12729418	12937927	1,00	0,98	5,668
(100, 0, 14)	(100, 0, 14)	7780729	7966531	1,00	0,98	3,462
(100, 0, 15)	(100, 0, 15)	5037696	5139322	1,00	0,98	2,209
(100, 0, 20)	(100, 0, 20)	1441085	1437171	1,00	1,00	0,538
(100, 0, 30)	(100, 0, 30)	170086	170086	1,00	1,00	0,059
(100, 0, 40)	(100, 0, 40)	40316	40357	1,00	1,00	0,014
(100, 0, 50)	(100, 0, 50)	10980	10988	1,00	1,00	0,004
(100, 0, 60)	(100, 0, 60)	4525	4526	1,00	1,00	0,003
(100, 0, 70)	(100, 0, 72)	1607	1571	0,89	0,91	0,002
(100, 0, 80)	(100, 0, 92)	177	308	0,38	0,22	0,003
(100, 0, 90)	(100, 0, 97)	121	178	0,49	0,34	0,001

Tableau I.15 Incidence de SI1 sur la résolution du problème du stable maximum dans des graphes initialement de 50 sommets

Structure initiale $(n(G), d(G))$	Structure résultante $((n(G'), d(G'))$	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(50, 0, 01)	(50, 0, 01)	—	790	1, 00	—	—
(50, 0, 02)	(50, 0, 02)	—	2641	1, 00	—	—
(50, 0, 03)	(50, 0, 03)	—	13012	1, 00	—	—
(50, 0, 04)	(50, 0, 04)	—	29365	1, 00	—	—
(50, 0, 05)	(51, 0, 05)	—	27876	1, 75	—	—
(50, 0, 06)	(51, 0, 06)	—	23648	1, 62	—	—
(50, 0, 07)	(53, 0, 06)	—	41692	1, 93	—	—
(50, 0, 08)	(54, 0, 07)	—	77695	3, 38	—	—
(50, 0, 09)	(56, 0, 07)	—	105757	4, 01	—	—
(50, 0, 10)	(57, 0, 08)	—	74223	2, 87	—	—
(50, 0, 11)	(61, 0, 08)	—	142217	8, 58	—	—
(50, 0, 12)	(63, 0, 08)	—	234307	13, 20	—	—
(50, 0, 13)	(66, 0, 07)	—	410021	22, 60	—	—
(50, 0, 14)	(70, 0, 07)	—	575447	44, 78	—	—
(50, 0, 15)	(73, 0, 07)	—	1646560	144, 06	—	—
(50, 0, 20)	(89, 0, 06)	—	25240434	3754, 73	—	—
(50, 0, 30)	(111, 0, 05)	—	—	—	—	—
(50, 0, 40)	(127, 0, 05)	—	—	—	—	—
(50, 0, 50)	(137, 0, 05)	—	—	—	—	—
(50, 0, 60)	(144, 0, 04)	—	—	—	—	—
(50, 0, 70)	(146, 0, 04)	—	—	—	—	—
(50, 0, 80)	(144, 0, 04)	—	—	—	—	—
(50, 0, 90)	(136, 0, 04)	—	—	—	—	—

Tableau I.16 Incidence de SI1 sur la résolution du problème du stable maximum dans des graphes initialement de 60 sommets

Structure initiale $(n(G), d(G))$	Structure résultante $((n(G'), d(G'))$	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(60, 0, 01)	(60, 0, 01)	—	6234	1,00	—	—
(60, 0, 02)	(60, 0, 02)	—	20855	1,00	—	—
(60, 0, 03)	(60, 0, 03)	—	119958	1,00	—	—
(60, 0, 04)	(60, 0, 04)	—	192322	1,00	—	—
(60, 0, 05)	(60, 0, 05)	—	135178	1,00	—	—
(60, 0, 06)	(63, 0, 06)	—	228793	2,18	—	—
(60, 0, 07)	(65, 0, 06)	—	311425	1,92	—	—
(60, 0, 08)	(67, 0, 06)	—	394057	4,10	—	—
(60, 0, 09)	(72, 0, 06)	—	1271533	9,19	—	—
(60, 0, 10)	(74, 0, 07)	—	2003598	20,14	—	—
(60, 0, 11)	(80, 0, 06)	—	4285648	46,57	—	—
(60, 0, 12)	(82, 0, 07)	—	6394804	73,61	—	—
(60, 0, 13)	(86, 0, 06)	—	18255506	232,76	—	—
(60, 0, 14)	(91, 0, 06)	—	43223459	847,94	—	—
(60, 0, 15)	(96, 0, 06)	—	121497032	2648,83	—	—
(60, 0, 20)	(115, 0, 05)	—	—	—	—	—
(60, 0, 30)	(143, 0, 04)	—	—	—	—	—
(60, 0, 40)	(167, 0, 04)	—	—	—	—	—
(60, 0, 50)	(181, 0, 04)	—	—	—	—	—
(60, 0, 60)	(191, 0, 04)	—	—	—	—	—
(60, 0, 70)	(195, 0, 03)	—	—	—	—	—
(60, 0, 80)	(183, 0, 04)	—	—	—	—	—
(60, 0, 90)	(168, 0, 03)	—	—	—	—	—

Tableau I.17 Incidence de SI1 sur la résolution du problème du stable maximum dans des graphes initialement de 70 sommets

Structure initiale ($n(G), d(G)$)	Structure résultante ($(n(G'), d(G'))$)	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(70, 0, 01)	(70, 0, 01)	—	26930	1,00	—	—
(70, 0, 02)	(70, 0, 02)	—	173204	1,00	—	—
(70, 0, 03)	(70, 0, 03)	—	743828	1,00	—	—
(70, 0, 04)	(71, 0, 04)	—	1820544	1,67	—	—
(70, 0, 05)	(73, 0, 05)	—	2400819	2,67	—	—
(70, 0, 06)	(75, 0, 05)	—	3402802	6,57	—	—
(70, 0, 07)	(77, 0, 06)	—	3155355	5,48	—	—
(70, 0, 08)	(78, 0, 06)	—	3532119	6,35	—	—
(70, 0, 09)	(86, 0, 06)	—	18255506	33,07	—	—
(70, 0, 10)	(91, 0, 06)	—	43223459	90,39	—	—
(70, 0, 11)	(96, 0, 06)	—	121497032	310,88	—	—
(70, 0, 12)	(104, 0, 06)	—	—	—	—	—
(70, 0, 13)	(108, 0, 05)	—	—	—	—	—
(70, 0, 14)	(115, 0, 05)	—	—	—	—	—
(70, 0, 15)	(117, 0, 05)	—	—	—	—	—
(70, 0, 20)	(147, 0, 04)	—	—	—	—	—
(70, 0, 30)	(181, 0, 04)	—	—	—	—	—
(70, 0, 40)	(211, 0, 03)	—	—	—	—	—
(70, 0, 50)	(229, 0, 03)	—	—	—	—	—
(70, 0, 60)	(243, 0, 03)	—	—	—	—	—
(70, 0, 70)	(241, 0, 03)	—	—	—	—	—
(70, 0, 80)	(230, 0, 03)	—	—	—	—	—
(70, 0, 90)	(203, 0, 03)	—	—	—	—	—

Tableau I.18 Incidence de SI1 sur la résolution du problème du stable maximum dans des graphes initialement de 80 sommets

Structure initiale $(n(G), d(G))$	Structure résultante $((n(G'), d(G'))$	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(80, 0, 01)	(80, 0, 01)	—	261714	1,00	—	—
(80, 0, 02)	(80, 0, 02)	—	1840429	1,00	—	—
(80, 0, 03)	(80, 0, 03)	—	4148619	1,00	—	—
(80, 0, 04)	(81, 0, 04)	—	13275080	1,58	—	—
(80, 0, 05)	(84, 0, 05)	—	18996894	3,22	—	—
(80, 0, 06)	(86, 0, 05)	—	25541461	5,96	—	—
(80, 0, 07)	(90, 0, 06)	—	27568744	6,65	—	—
(80, 0, 08)	(96, 0, 06)	—	121497032	38,67	—	—
(80, 0, 09)	(105, 0, 05)	—	—	—	—	—
(80, 0, 10)	(113, 0, 05)	—	—	—	—	—
(80, 0, 11)	(121, 0, 05)	—	—	—	—	—
(80, 0, 12)	(127, 0, 05)	—	—	—	—	—
(80, 0, 13)	(135, 0, 05)	—	—	—	—	—
(80, 0, 14)	(140, 0, 05)	—	—	—	—	—
(80, 0, 15)	(146, 0, 04)	—	—	—	—	—
(80, 0, 20)	(181, 0, 04)	—	—	—	—	—
(80, 0, 30)	(222, 0, 03)	—	—	—	—	—
(80, 0, 40)	(261, 0, 03)	—	—	—	—	—
(80, 0, 50)	(282, 0, 03)	—	—	—	—	—
(80, 0, 60)	(300, 0, 02)	—	—	—	—	—
(80, 0, 70)	(296, 0, 02)	—	—	—	—	—
(80, 0, 80)	(283, 0, 03)	—	—	—	—	—
(80, 0, 90)	(243, 0, 03)	—	—	—	—	—

Tableau I.19 Incidence de SI1 sur la résolution du problème du stable maximum dans des graphes initialement de 90 sommets

Structure initiale ($n(G), d(G)$)	Structure résultante ($(n(G'), d(G'))$)	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(90, 0, 01)	(90, 0, 01)	—	6959784	1, 00	—	—
(90, 0, 02)	(90, 0, 02)	—	8291193	1, 00	—	—
(90, 0, 03)	(91, 0, 03)	—	47857067	1, 44	—	—
(90, 0, 04)	(92, 0, 04)	—	90976018	1, 60	—	—
(90, 0, 05)	(95, 0, 05)	—	73048525	1, 89	—	—
(90, 0, 06)	(102, 0, 05)	—	—	—	—	—
(90, 0, 07)	(106, 0, 05)	—	—	—	—	—
(90, 0, 08)	(116, 0, 05)	—	—	—	—	—
(90, 0, 09)	(125, 0, 05)	—	—	—	—	—
(90, 0, 10)	(130, 0, 05)	—	—	—	—	—
(90, 0, 11)	(141, 0, 04)	—	—	—	—	—
(90, 0, 12)	(156, 0, 04)	—	—	—	—	—
(90, 0, 13)	(158, 0, 04)	—	—	—	—	—
(90, 0, 14)	(167, 0, 04)	—	—	—	—	—
(90, 0, 15)	(173, 0, 04)	—	—	—	—	—
(90, 0, 20)	(215, 0, 03)	—	—	—	—	—
(90, 0, 30)	(276, 0, 03)	—	—	—	—	—
(90, 0, 40)	(310, 0, 02)	—	—	—	—	—
(90, 0, 50)	(341, 0, 02)	—	—	—	—	—
(90, 0, 60)	(355, 0, 02)	—	—	—	—	—
(90, 0, 70)	(362, 0, 02)	—	—	—	—	—
(90, 0, 80)	(342, 0, 02)	—	—	—	—	—
(90, 0, 90)	(286, 0, 02)	—	—	—	—	—

Tableau I.20 Incidence de SI1 sur la résolution du problème du stable maximum dans des graphes initialement de 100 sommets

Structure initiale $(n(G), d(G))$	Structure résultante $((n(G'), d(G'))$	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(100, 0, 01)	(100, 0, 01)	—	6133703	1,00	—	—
(100, 0, 02)	(100, 0, 02)	—	72691527	1,00	—	—
(100, 0, 03)	(101, 0, 03)	—	—	—	—	—
(100, 0, 04)	(102, 0, 04)	—	—	—	—	—
(100, 0, 05)	(108, 0, 04)	—	—	—	—	—
(100, 0, 06)	(115, 0, 05)	—	—	—	—	—
(100, 0, 07)	(123, 0, 05)	—	—	—	—	—
(100, 0, 08)	(135, 0, 04)	—	—	—	—	—
(100, 0, 09)	(147, 0, 04)	—	—	—	—	—
(100, 0, 10)	(153, 0, 04)	—	—	—	—	—
(100, 0, 11)	(167, 0, 04)	—	—	—	—	—
(100, 0, 12)	(179, 0, 04)	—	—	—	—	—
(100, 0, 13)	(188, 0, 04)	—	—	—	—	—
(100, 0, 14)	(193, 0, 04)	—	—	—	—	—
(100, 0, 15)	(210, 0, 03)	—	—	—	—	—
(100, 0, 20)	(256, 0, 03)	—	—	—	—	—
(100, 0, 30)	(318, 0, 02)	—	—	—	—	—
(100, 0, 40)	(363, 0, 02)	—	—	—	—	—
(100, 0, 50)	(402, 0, 02)	—	—	—	—	—
(100, 0, 60)	(423, 0, 02)	—	—	—	—	—
(100, 0, 70)	(421, 0, 02)	—	—	—	—	—
(100, 0, 80)	(399, 0, 02)	—	—	—	—	—
(100, 0, 90)	(332, 0, 02)	—	—	—	—	—

Tableau I.21 Incidence de SI1 suivie de SSD sur la résolution du problème du stable maximum dans des graphes initialement de 50 sommets

Structure initiale ($n(G), d(G)$)	Structure résultante ($(n(G'), d(G'))$)	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(50, 0, 01)	(41, 0, 00)	—	—	—	—	—
(50, 0, 02)	(39, 0, 00)	—	—	—	—	—
(50, 0, 03)	(36, 0, 01)	—	—	—	—	—
(50, 0, 04)	(38, 0, 02)	—	—	—	—	—
(50, 0, 05)	(41, 0, 03)	—	—	—	—	—
(50, 0, 06)	(44, 0, 04)	—	—	—	—	—
(50, 0, 07)	(48, 0, 06)	—	—	—	—	—
(50, 0, 08)	(49, 0, 07)	—	—	—	—	—
(50, 0, 09)	(51, 0, 07)	—	35601	1,35	—	—
(50, 0, 10)	(54, 0, 07)	—	77695	3,01	—	—
(50, 0, 11)	(59, 0, 07)	—	147851	8,92	—	—
(50, 0, 12)	(61, 0, 08)	—	142217	8,01	—	—
(50, 0, 13)	(64, 0, 07)	—	327308	18,04	—	—
(50, 0, 14)	(69, 0, 07)	—	534091	41,57	—	—
(50, 0, 15)	(73, 0, 07)	—	1646560	144,06	—	—
(50, 0, 20)	(89, 0, 06)	—	25240434	3754,73	—	—
(50, 0, 30)	(111, 0, 05)	—	—	—	—	—
(50, 0, 40)	(127, 0, 05)	—	—	—	—	—
(50, 0, 50)	(137, 0, 05)	—	—	—	—	—
(50, 0, 60)	(144, 0, 04)	—	—	—	—	—
(50, 0, 70)	(146, 0, 04)	—	—	—	—	—
(50, 0, 80)	(144, 0, 04)	—	—	—	—	—
(50, 0, 90)	(134, 0, 04)	—	—	—	—	—

Tableau I.22 Incidence de SI1 suivie de SSD sur la résolution du problème du stable maximum dans des graphes initialement de 60 sommets

Structure initiale $(n(G), d(G))$	Structure résultante $((n(G'), d(G'))$	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(60, 0, 01)	(48, 0, 00)	—	—	—	—	—
(60, 0, 02)	(46, 0, 00)	—	—	—	—	—
(60, 0, 03)	(46, 0, 02)	—	—	—	—	—
(60, 0, 04)	(49, 0, 03)	—	—	—	—	—
(60, 0, 05)	(51, 0, 04)	—	45661	0,34	—	—
(60, 0, 06)	(55, 0, 05)	—	75566	0,72	—	—
(60, 0, 07)	(60, 0, 06)	—	104846	0,65	—	—
(60, 0, 08)	(63, 0, 06)	—	228793	2,38	—	—
(60, 0, 09)	(70, 0, 06)	—	518004	3,75	—	—
(60, 0, 10)	(72, 0, 07)	—	1289522	12,96	—	—
(60, 0, 11)	(80, 0, 06)	—	4285648	46,57	—	—
(60, 0, 12)	(82, 0, 07)	—	6394804	73,61	—	—
(60, 0, 13)	(86, 0, 06)	—	18255506	232,76	—	—
(60, 0, 14)	(91, 0, 06)	—	43223459	847,94	—	—
(60, 0, 15)	(96, 0, 06)	—	121497032	2648,83	—	—
(60, 0, 20)	(115, 0, 05)	—	—	—	—	—
(60, 0, 30)	(143, 0, 04)	—	—	—	—	—
(60, 0, 40)	(167, 0, 04)	—	—	—	—	—
(60, 0, 50)	(181, 0, 04)	—	—	—	—	—
(60, 0, 60)	(191, 0, 04)	—	—	—	—	—
(60, 0, 70)	(195, 0, 03)	—	—	—	—	—
(60, 0, 80)	(183, 0, 04)	—	—	—	—	—
(60, 0, 90)	(167, 0, 03)	—	—	—	—	—

Tableau I.23 Incidence de SI1 suivie de SSD sur la résolution du problème du stable maximum dans des graphes initialement de 70 sommets

Structure initiale ($n(G), d(G)$)	Structure résultante ($(n(G'), d(G'))$)	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(70, 0, 01)	(55, 0, 00)	—	55	0, 00	—	—
(70, 0, 02)	(52, 0, 01)	—	1879	0, 01	—	—
(70, 0, 03)	(55, 0, 02)	—	11748	0, 02	—	—
(70, 0, 04)	(59, 0, 03)	—	109263	0, 10	—	—
(70, 0, 05)	(66, 0, 04)	—	729297	0, 81	—	—
(70, 0, 06)	(70, 0, 05)	—	897844	1, 73	—	—
(70, 0, 07)	(74, 0, 06)	—	2025062	3, 52	—	—
(70, 0, 08)	(76, 0, 06)	—	2778590	4, 99	—	—
(70, 0, 09)	(85, 0, 06)	—	15927196	28, 85	—	—
(70, 0, 10)	(90, 0, 06)	—	27568744	57, 65	—	—
(70, 0, 11)	(96, 0, 06)	—	121497032	310, 88	—	—
(70, 0, 12)	(104, 0, 05)	—	—	—	—	—
(70, 0, 13)	(108, 0, 05)	—	—	—	—	—
(70, 0, 14)	(115, 0, 05)	—	—	—	—	—
(70, 0, 15)	(117, 0, 05)	—	—	—	—	—
(70, 0, 20)	(147, 0, 04)	—	—	—	—	—
(70, 0, 30)	(181, 0, 04)	—	—	—	—	—
(70, 0, 40)	(211, 0, 03)	—	—	—	—	—
(70, 0, 50)	(229, 0, 03)	—	—	—	—	—
(70, 0, 60)	(243, 0, 03)	—	—	—	—	—
(70, 0, 70)	(241, 0, 03)	—	—	—	—	—
(70, 0, 80)	(230, 0, 03)	—	—	—	—	—
(70, 0, 90)	(203, 0, 03)	—	—	—	—	—

Tableau I.24 Incidence de SI1 suivie de SSD sur la résolution du problème du stable maximum dans des graphes initialement de 80 sommets

Structure initiale ($n(G), d(G)$)	Structure résultante ($(n(G'), d(G'))$)	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(80, 0, 01)	(62, 0, 00)	—	62	0,00	—	—
(80, 0, 02)	(59, 0, 01)	—	5690	0,00	—	—
(80, 0, 03)	(62, 0, 02)	—	51325	0,01	—	—
(80, 0, 04)	(70, 0, 03)	—	743828	0,09	—	—
(80, 0, 05)	(76, 0, 04)	—	5486860	0,93	—	—
(80, 0, 06)	(83, 0, 05)	—	15724610	3,67	—	—
(80, 0, 07)	(86, 0, 05)	—	25541461	6,16	—	—
(80, 0, 08)	(95, 0, 05)	—	73048525	23,25	—	—
(80, 0, 09)	(104, 0, 05)	—	—	—	—	—
(80, 0, 10)	(112, 0, 05)	—	—	—	—	—
(80, 0, 11)	(121, 0, 05)	—	—	—	—	—
(80, 0, 12)	(127, 0, 05)	—	—	—	—	—
(80, 0, 13)	(135, 0, 04)	—	—	—	—	—
(80, 0, 14)	(140, 0, 05)	—	—	—	—	—
(80, 0, 15)	(146, 0, 04)	—	—	—	—	—
(80, 0, 20)	(181, 0, 04)	—	—	—	—	—
(80, 0, 30)	(222, 0, 03)	—	—	—	—	—
(80, 0, 40)	(261, 0, 03)	—	—	—	—	—
(80, 0, 50)	(282, 0, 03)	—	—	—	—	—
(80, 0, 60)	(300, 0, 02)	—	—	—	—	—
(80, 0, 70)	(296, 0, 02)	—	—	—	—	—
(80, 0, 80)	(283, 0, 03)	—	—	—	—	—
(80, 0, 90)	(243, 0, 03)	—	—	—	—	—

Tableau I.25 Incidence de SI1 suivie de SSD sur la résolution du problème du stable maximum dans des graphes initialement de 90 sommets

Structure initiale ($n(G), d(G)$)	Structure résultante ($(n(G'), d(G'))$)	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(90, 0, 01)	(69, 0, 00)	—	69	0,00	—	—
(90, 0, 02)	(70, 0, 01)	—	26930	0,00	—	—
(90, 0, 03)	(76, 0, 02)	—	1173539	0,04	—	—
(90, 0, 04)	(84, 0, 04)	—	27840583	0,49	—	—
(90, 0, 05)	(91, 0, 04)	—	73973804	1,91	—	—
(90, 0, 06)	(100, 0, 05)	—	107466454	3,90	—	—
(90, 0, 07)	(106, 0, 05)	—	—	—	—	—
(90, 0, 08)	(116, 0, 05)	—	—	—	—	—
(90, 0, 09)	(125, 0, 05)	—	—	—	—	—
(90, 0, 10)	(130, 0, 05)	—	—	—	—	—
(90, 0, 11)	(141, 0, 04)	—	—	—	—	—
(90, 0, 12)	(156, 0, 04)	—	—	—	—	—
(90, 0, 13)	(158, 0, 04)	—	—	—	—	—
(90, 0, 14)	(167, 0, 04)	—	—	—	—	—
(90, 0, 15)	(173, 0, 04)	—	—	—	—	—
(90, 0, 20)	(215, 0, 03)	—	—	—	—	—
(90, 0, 30)	(276, 0, 03)	—	—	—	—	—
(90, 0, 40)	(310, 0, 02)	—	—	—	—	—
(90, 0, 50)	(341, 0, 02)	—	—	—	—	—
(90, 0, 60)	(355, 0, 02)	—	—	—	—	—
(90, 0, 70)	(362, 0, 02)	—	—	—	—	—
(90, 0, 80)	(342, 0, 02)	—	—	—	—	—
(90, 0, 90)	(286, 0, 02)	—	—	—	—	—

Tableau I.26 Incidence de SI1 suivie de SSD sur la résolution du problème du stable maximum dans des graphes initialement de 100 sommets

Structure initiale $(n(G), d(G))$	Structure résultante $((n(G'), d(G'))$	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(100, 0, 01)	(77, 0, 00)	—	77	0,00	—	—
(100, 0, 02)	(77, 0, 01)	—	191279	0,00	—	—
(100, 0, 03)	(88, 0, 03)	—	27398050	0,15	—	—
(100, 0, 04)	(95, 0, 04)	—	141982661	0,63	—	—
(100, 0, 05)	(104, 0, 04)	—	—	—	—	—
(100, 0, 06)	(114, 0, 05)	—	—	—	—	—
(100, 0, 07)	(122, 0, 05)	—	—	—	—	—
(100, 0, 08)	(134, 0, 04)	—	—	—	—	—
(100, 0, 09)	(147, 0, 04)	—	—	—	—	—
(100, 0, 10)	(153, 0, 04)	—	—	—	—	—
(100, 0, 11)	(167, 0, 04)	—	—	—	—	—
(100, 0, 12)	(179, 0, 04)	—	—	—	—	—
(100, 0, 13)	(188, 0, 04)	—	—	—	—	—
(100, 0, 14)	(193, 0, 04)	—	—	—	—	—
(100, 0, 15)	(210, 0, 03)	—	—	—	—	—
(100, 0, 20)	(256, 0, 03)	—	—	—	—	—
(100, 0, 30)	(318, 0, 02)	—	—	—	—	—
(100, 0, 40)	(363, 0, 02)	—	—	—	—	—
(100, 0, 50)	(402, 0, 02)	—	—	—	—	—
(100, 0, 60)	(423, 0, 02)	—	—	—	—	—
(100, 0, 70)	(421, 0, 02)	—	—	—	—	—
(100, 0, 80)	(399, 0, 02)	—	—	—	—	—
(100, 0, 90)	(332, 0, 02)	—	—	—	—	—

Tableau I.27 Incidence de SIALEA1 sur la résolution du problème du stable maximum dans des graphes initialement de 100 sommets

Structure initiale $(n(G), d(G))$	Structure résultante $((n(G'), d(G'))$	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(100, 0, 01)	(100, 0, 01)	—	6133703	1,00	—	—
(100, 0, 02)	(100, 0, 02)	—	72691527	1,00	—	—
(100, 0, 03)	(100, 0, 03)	—	179676999	1,00	—	—
(100, 0, 04)	(101, 0, 04)	—	—	—	—	—
(100, 0, 05)	(105, 0, 05)	—	—	—	—	—
(100, 0, 06)	(111, 0, 05)	—	—	—	—	—
(100, 0, 07)	(121, 0, 05)	—	—	—	—	—
(100, 0, 08)	(130, 0, 05)	—	—	—	—	—
(100, 0, 09)	(143, 0, 05)	—	—	—	—	—
(100, 0, 10)	(148, 0, 05)	—	—	—	—	—
(100, 0, 11)	(164, 0, 04)	—	—	—	—	—
(100, 0, 12)	(172, 0, 04)	—	—	—	—	—
(100, 0, 13)	(181, 0, 04)	—	—	—	—	—
(100, 0, 14)	(190, 0, 04)	—	—	—	—	—
(100, 0, 15)	(202, 0, 03)	—	—	—	—	—
(100, 0, 20)	(250, 0, 03)	—	—	—	—	—
(100, 0, 30)	(316, 0, 02)	—	—	—	—	—
(100, 0, 40)	(366, 0, 02)	—	—	—	—	—
(100, 0, 50)	(401, 0, 02)	—	—	—	—	—
(100, 0, 60)	(417, 0, 02)	—	—	—	—	—
(100, 0, 70)	(420, 0, 02)	—	—	—	—	—
(100, 0, 80)	(402, 0, 02)	—	—	—	—	—
(100, 0, 90)	(339, 0, 02)	—	—	—	—	—

Tableau I.28 Incidence de SIALEA1 suivie de SI1 sur la résolution du problème du stable maximum dans des graphes initialement de 100 sommets

Structure initiale ($n(G), d(G)$)	Structure résultante ($(n(G'), d(G'))$)	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(100, 0, 01)	(100, 0, 01)	—	6133703	1,00	—	—
(100, 0, 02)	(100, 0, 02)	—	72691527	1,00	—	—
(100, 0, 03)	(101, 0, 03)	—	—	—	—	—
(100, 0, 04)	(102, 0, 04)	—	—	—	—	—
(100, 0, 05)	(108, 0, 04)	—	—	—	—	—
(100, 0, 06)	(115, 0, 05)	—	—	—	—	—
(100, 0, 07)	(124, 0, 05)	—	—	—	—	—
(100, 0, 08)	(135, 0, 05)	—	—	—	—	—
(100, 0, 09)	(148, 0, 04)	—	—	—	—	—
(100, 0, 10)	(153, 0, 04)	—	—	—	—	—
(100, 0, 11)	(168, 0, 04)	—	—	—	—	—
(100, 0, 12)	(179, 0, 04)	—	—	—	—	—
(100, 0, 13)	(187, 0, 04)	—	—	—	—	—
(100, 0, 14)	(196, 0, 03)	—	—	—	—	—
(100, 0, 15)	(209, 0, 03)	—	—	—	—	—
(100, 0, 20)	(254, 0, 03)	—	—	—	—	—
(100, 0, 30)	(322, 0, 02)	—	—	—	—	—
(100, 0, 40)	(371, 0, 02)	—	—	—	—	—
(100, 0, 50)	(406, 0, 02)	—	—	—	—	—
(100, 0, 60)	(421, 0, 02)	—	—	—	—	—
(100, 0, 70)	(423, 0, 02)	—	—	—	—	—
(100, 0, 80)	(405, 0, 02)	—	—	—	—	—
(100, 0, 90)	(340, 0, 02)	—	—	—	—	—

Tableau I.29 Incidence de SIALEA2 sur la résolution du problème du stable maximum dans des graphes initialement de 100 sommets

Structure initiale $(n(G), d(G))$	Structure résultante $((n(G'), d(G'))$	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(100, 0, 01)	(100, 0, 01)	—	6133703	1,00	—	—
(100, 0, 02)	(100, 0, 02)	—	72691527	1,00	—	—
(100, 0, 03)	(100, 0, 03)	—	179676999	1,00	—	—
(100, 0, 04)	(100, 0, 04)	—	226993732	1,00	—	—
(100, 0, 05)	(100, 0, 05)	—	107466454	1,00	—	—
(100, 0, 06)	(101, 0, 06)	—	—	—	—	—
(100, 0, 07)	(102, 0, 07)	—	—	—	—	—
(100, 0, 08)	(108, 0, 07)	—	—	—	—	—
(100, 0, 09)	(114, 0, 07)	—	—	—	—	—
(100, 0, 10)	(120, 0, 07)	—	—	—	—	—
(100, 0, 11)	(128, 0, 07)	—	—	—	—	—
(100, 0, 12)	(140, 0, 06)	—	—	—	—	—
(100, 0, 13)	(145, 0, 06)	—	—	—	—	—
(100, 0, 14)	(154, 0, 05)	—	—	—	—	—
(100, 0, 15)	(159, 0, 05)	—	—	—	—	—
(100, 0, 20)	(202, 0, 04)	—	—	—	—	—
(100, 0, 30)	(257, 0, 04)	—	—	—	—	—
(100, 0, 40)	(301, 0, 03)	—	—	—	—	—
(100, 0, 50)	(326, 0, 03)	—	—	—	—	—
(100, 0, 60)	(340, 0, 03)	—	—	—	—	—
(100, 0, 70)	(347, 0, 03)	—	—	—	—	—
(100, 0, 80)	(330, 0, 03)	—	—	—	—	—
(100, 0, 90)	(291, 0, 03)	—	—	—	—	—

Tableau I.30 Incidence de SIALEA2 suivie de SI2 sur la résolution du problème du stable maximum dans des graphes initialement de 100 sommets

Structure initiale $(n(G), d(G))$	Structure résultante $((n(G'), d(G'))$	Coût en nombre de retours arrières				Coût en secondes C_A
		C_A	$c_A(n(G'), d(G'))$	r_A	r'_A	
(100, 0, 01)	(100, 0, 01)	—	6133703	1,00	—	—
(100, 0, 02)	(100, 0, 02)	—	72691527	1,00	—	—
(100, 0, 03)	(100, 0, 03)	—	179676999	1,00	—	—
(100, 0, 04)	(100, 0, 04)	—	226993732	1,00	—	—
(100, 0, 05)	(101, 0, 05)	—	—	—	—	—
(100, 0, 06)	(102, 0, 06)	—	—	—	—	—
(100, 0, 07)	(105, 0, 06)	—	—	—	—	—
(100, 0, 08)	(111, 0, 06)	—	—	—	—	—
(100, 0, 09)	(117, 0, 07)	—	—	—	—	—
(100, 0, 10)	(123, 0, 06)	—	—	—	—	—
(100, 0, 11)	(134, 0, 06)	—	—	—	—	—
(100, 0, 12)	(143, 0, 06)	—	—	—	—	—
(100, 0, 13)	(149, 0, 06)	—	—	—	—	—
(100, 0, 14)	(159, 0, 05)	—	—	—	—	—
(100, 0, 15)	(165, 0, 05)	—	—	—	—	—
(100, 0, 20)	(206, 0, 04)	—	—	—	—	—
(100, 0, 30)	(259, 0, 03)	—	—	—	—	—
(100, 0, 40)	(306, 0, 03)	—	—	—	—	—
(100, 0, 50)	(329, 0, 03)	—	—	—	—	—
(100, 0, 60)	(345, 0, 03)	—	—	—	—	—
(100, 0, 70)	(350, 0, 03)	—	—	—	—	—
(100, 0, 80)	(332, 0, 03)	—	—	—	—	—
(100, 0, 90)	(292, 0, 03)	—	—	—	—	—