Étiquetage de graphe et “cyclic-bandwidth problem”

Programmation par contraintes
Rapport projet

Khedaoudj HALIMI – Juanfer MERCIER

Table des matières

Université de Nantes — Faculté des Sciences et Technologies
Master Informatique — Parcours optimisation et recherche opérationnelle
Année académique 2021-2022

Lundi 11 Mars 2022

1 Introduction

Soit un graphe G défini par un couple $(V(G), E(G))$ où $V(G)$ est un ensemble de sommets et $E(G)$ est un ensemble de paires d'éléments de $V(G)$ appelé ensemble des arêtes. Nous noterons $n = |V(G)|$ le nombre de sommets de G.
Un étiquetage f d'un graphe G est une fonction bijective de $V(G)$ dans $\{1, \ldots, n\}$, c'est-à-dire, $f : V(G) \rightarrow \{1, \ldots, n\}$. Dans ce projet, on s'intéresse au problème de cyclic-bandwidth que nous définirons ci-après.

Soit G un graphe, f un étiquetage de G et (u, v) une arête de G. Le nombre $|f(v)-f(u)|$ est appelé distance entre les sommets u et v (où $|x|$ représente la valeur absolue de x). La "cyclic-bandwidth" du graphe G par rapport à l'étiquetage f est donnée par $CB(G, f) = \max\limits_{(u,v)\in E(G)} \{min\{|f(v) - f(u)|, n - |f(v) - f(u)|\}\}$. Le "cyclic-bandwidth problem (CBP)" consiste à trouver l'étiquetage g tel que $CB(G, g) = \min\limits_{f\in\varepsilon}\{CB(G, f)\}$ où ε représente l'ensemble des étiquetages.

1.1 Applications du CBP

Introduit pour la première fois pour formuler un problème de conception dans le domaine des réseaux d'interconnexion en anneau, le "cyclic-bandwidth problem (CBP)" s'est également révélé être un modèle pertinent pour un certain nombre d'autres applications, notamment la conception des circuits intégrés VLSI ("Very Large Scale Integration"), la représentation de structures de données, la conception de codes et les systèmes informatiques parallèles. En termes de complexité de calcul, la version de décision du CBP est NP-complet. Par conséquent, le CBP est un défi informatique pour les méthodes de résolution. (voir Références)

1.2 Structure du rapport

Dans la suite de ce rapport, nous présenterons six modèles pour représenter le CBP. Les trois premiers modèles sont :

- M1, un modèle pour minimiser la cyclic-bandwidth basé sur des variables "domaine fini" entières (FD)
- M2, un modèle également basé sur des variables "domaine fini" entières (FD) mais s'intéressant à la satisfaction (*i.e.* recherche d'un étiquetage de cyclic-bandwidth de valeur k donnée)
- M3, un modèle basé sur des variables booléennes et contraintes booléennes (SAT), s'intéressant également à la satisfaction

Les trois autres modèles sont en fait des variantes des modèles mentionnés ci-dessus dans lequel nous incluons des contraintes dans le but de casser d'éventuelles symétries. Les deux premiers modèles et leurs variantes sont programmés en PyCSP3 tandis que le troisième modèle et sa variante sont modélisés grâce à la représentation textuelle DIMACS CNF. Pour la résolution, le solveur ACE est utilisé pour les modèles M1 et M2. Le modèle M3 est résolu avec le solveur Glucose4. Nous présentons une étude comparative des six modèles et un algorithme dichotomique parallèle utilisant les modèles de satisfaction (M2, M3 et leurs variantes) pour obtenir la cyclic-bandwidth de valeur optimale.

2 Modélisation

Modèle 1

Ce modèle est basé sur des variables "domaine fini" entières (FD) et vise à minimiser la cyclic-bandwidth.

`Données:` Un graphe $G(V(G), E(G))$, $n = |V(G)|$ et $k \in [\![1, n]\!]$
`Variables:` x_i avec $i \in [\![1, n]\!]$
`Domaine des variables:` $x_i \in [\![1, n]\!]^n$
`Contraintes:` $\texttt{alldifferent}(x_i),\ \forall\, i \in [\![1, n]\!]$
`Fonction objectif:` $\min\{\max_{(u,v)\in E(G)} \{\min\{|x_i - x_j|, n - |x_i - x_i|\}\}\}$

Modèle 2

Ce modèle est basé sur des variables "domaine fini" entières (FD). Plus axé sur la satisfaction, il consiste à trouver, s'il existe, un étiquetage qui permet d'avoir une cyclic-bandwidth égale à une valeur k donnée.

`Données:` Un graphe $G(V(G), E(G))$, $n = |V(G)|$, $k \in [\![1, n]\!]$ et
$P = \{(i, j) \mid \min(|j - i|, n - |j - i|) \leq k \ \text{ et } \ i \neq j \ \text{ avec } \ (i, j) \in [\![1, n]\!]^2\}$
`Variables:` x_i avec $i \in [\![1, n]\!]$
`Domaine des variables:` $x_i \in [\![1, n]\!]^n$
`Contraintes:`

- $\texttt{alldifferent}(x_i),\ \forall\, i \in [\![1, n]\!]$
- $\forall\, (i, j) \in E(G),\ (x_i, x_j) \in P$

Modèle 3

Ce modèle est similaire au modèle 2 sauf qu'il est basé sur des variables et contraintes booléennes (SAT).

`Données:` Un graphe $G(V(G), E(G))$, $n = |V(G)|$, k la valeur de la cyclic-bandwidth à satisfaire et P un ensemble de booléens b_{ij} tel que

$$b_{ij} = \begin{cases} T & \text{si } min(|j - i|, n - |j - i|) \leq k,\ \forall\, (i, j) \in [\![1, n]\!]^2 \\ -T & \text{sinon.} \end{cases}$$

`Variables:` X une matrice de booléens indexée par $(i, j) \in [\![1, n]\!]^2$ tel que $x_{ij} = Vrai$ si le sommet i est étiqueté par j
`Domaine:` $x_{ij} \in \{Vrai, Faux\},\ \forall\, (i, j) \in [\![1, n]\!]^2$
`Contraintes:`

- T
- $\bigwedge_{i=1}^{n} \bigvee_{j=1}^{n} (x_{ij})$
- $\bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n} \bigwedge_{\substack{k=1 \\ k\neq i}}^{n} (\neg x_{ij} \vee \neg x_{kj})$
- $\bigwedge_{(u,v)\in E(G)} \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n} (\neg x_{ui} \vee \neg x_{vj} \vee P_{ij})$

Le modèle 3 présenté ici est en fait une version améliorée qui n'a pas de contrainte redondante. En effet, notre modèle original a une contrainte de plus :

- $\bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n} \bigwedge_{\substack{k=1 \\ k \neq j}}^{n} (\neg x_{ij} \vee \neg x_{ik})$

De plus, la première contrainte est rajoutée de sorte à ce que T soit toujours évalué à vrai.

2.1 Comparaison des modèles

Nous noterons par la suite M1, M2 et M3 les modèles que nous avons présenté. Pour comparer ces différents modèles, nous les avons testé sur 45 graphes du benchmark Harwell-Boeing 113. La Table 1 présente les caractéristiques des différentes instances utilisées (le nombre de sommets et le nombre d'arêtes des graphes). Nos expérimentations ont toutes été exécutées sur une machine ayant le système d'exploitation Ubuntu, 16Go de mémoire vive et un processeur Intel Core i7-10750H de $10^{\text{ème}}$ génération avec 6 Coeurs / 12 Threads et une fréquence de base de 2.6GHz (5.0GHz maximum). De plus, un temps limite de résolution (TIMEOUT) de 600 secondes (10 minutes) a été fixé. Enfin, étant donné que les modèles sont compilés pour une instance donnée (et pour un k donné pour M2 et M3), nous avons aussi donné les temps de compilation/génération des modèles dans les tables 2 et 3. Ainsi, lors de la présentation des temps de résolution, nous ne présentons pas les temps de génération des modèles mais ces temps de génération sont tout de même pertinents (nous expliquerons pourquoi dans les lignes qui suivent).

Les tables 6 et 7 rapportent les temps de résolution, en secondes, pour chaque instance avec les différents modèles et pour $k \in [\![1, 6]\!]$ pour M2 et M3. Nous observons dans un premier temps que M1 atteint souvent le temps limite (TIMEOUT) et il ne donne de résultats que pour trois instances (bcspwr01, pores_1 et will57). La raison en est simple : le solveur ACE que l'on utilise, utilise un algorithme de descente pour trouver la valeur k du cyclic-bandwidth optimale en considérant un intervalle tel que $k \in [-\infty, +\infty]$ (avec ∞ une valeur prédéfinie par le solveur). Cette recherche est inefficace et nous verrons par la suite comment améliorer les performances du M1 en modifiant les paramètres du solveur. En comparaison, M2 et M3 sont beaucoup plus performants que le M1 tel quel. Cependant, bien que les performances du M2 et M3 sont du même ordre de grandeur pour les "petites" instances, le M3 présente des difficultés sur les plus "grandes" instances. De plus, les tables 2 et 3 montrent que les temps de génération pour M3 sont aussi beaucoup plus importants que les temps de génération du M1 et M2. L'explosion du nombre de clauses lors de la génération des modèles pour les grandes instances explique ces temps de génération anormalement longs. Que ce soit en terme de mémoire vive ou d'espace de stockage, les instances intermédiaires CNF générées à partir du M3 sont lourdes. Le M3 que l'on a précédemment présenté comporte des clauses en moins par rapport à notre modèle 3 original mais l'augmentation du nombre de clauses selon la taille et la topologie de l'instance est toujours du même ordre. Nous avons considéré l'utilisation de différentes structures de données pour stocker les clauses ou l'utilisation de formats binaires (plutôt que textuels) pour améliorer les temps de génération pour M3 mais le gain de temps a été jugé négligeable.

3 Gestion des symétries

Un étiquetage ε étant un n-uplet d'entiers unique allant de 1 à n avec $n = |V(G)|$, un n-uplet $s(\varepsilon)$ symétrique à ε donne la même valeur de cyclic-bandwidth que ε. Ainsi, pour améliorer les performances des modèles, on peut réduire l'espace de recherche en "cassant des symétries". En d'autres mots, lorsque l'on considère un étiquetage ε d'une certaine valeur k alors on peut éviter de considérer les étiquetages $s(\varepsilon)$ symétriques à ε car de toute façon ils ont tous la même valeur de cyclic bandwidth CB. Nous considérons deux types de symétries : la symétrie centrale et la symétrie axiale.

3.1 Symétrie centrale

On dit que deux étiquetages sont symétriques par rotation si on peut obtenir l'un en faisant tourner l'autre dans un sens donné. Pour casser ce type de symétrie, il suffit de fixer la valeur de la fonction d'étiquetage pour un sommet donné. Par exemple, en imaginant un graphe étiqueté comme un cycle, si pour chaque étiquetage on fixe $\varepsilon(1) = 1$ alors le fait de fixer la valeur de l'étiquetage pour un sommet donné va empêcher la rotation. En d'autres mots, en fixant la valeur de l'étiquetage d'un sommet, on élimine $n - 1$ permutations de ε.

Exemple :
ε_1 et ε_2 deux étiquetages symétriques par rotation
$\varepsilon_1 = (3, 5, 4, 2, 1)$ et $\varepsilon_2 = (1, 3, 5, 4, 2)$

3.2 Symétrie axiale

On dit que deux étiquetages sont symétriques selon un axe de symétrie si on peut obtenir l'un en faisant tourner l'autre par rapport à un axe de symétrie passant par un sommet du graphe. Pour casser ce type de symétrie, il suffit de fixer un ordre entre deux valeurs de la fonction d'étiquetage. On peut, par exemple, donner pour chaque étiquetage $\varepsilon(2) < \varepsilon(n)$ car grâce à cet ordre on s'assure que $\varepsilon(2)$ soit toujours avant $\varepsilon(n)$ dans un sens donné si l'on considère que le graphe est étiqueté comme un cycle (on imagine que le graphe est "comme une roue").

Exemple :
ε_1 et ε_2 deux étiquetages symétriques par rotation (ici l'axe passe par le sommet 1)
$\varepsilon_1 = (3, 2, 4, 1)$ et $\varepsilon_2 = (3, 1, 4, 2)$

Contraintes à rajouter aux modèles pour casser les symétries

- $x_1 = 1$ (l'équivalent pour M3 est l'ajout de la clause $x_{1,1}$)
- $x_2 < x_n$ (nous n'avons pas rajouté d'équivalent de cette contrainte pour M3)

La valeur que l'on attribue à x_1 est arbitraire, nous avons considéré de fixer x_1 tel que $x_1 = \Delta(G)$ avec $\Delta(G)$ le degré du graphe mais les expérimentations n'ont pas montré des résultats concluants. De même, nous avons considéré de fixer l'étiquetage du sommet s du graphe avec le plus haut degré à 1 ($x_s = 1$), là encore sans succès.

3.3 Comparaison des différents modèles sans symétries

Nous noterons par la suite M1+, M2+ et M3+ les modèles sans symétries. Les tables 4 et 5 présentent les temps de génération des instances intermédiaires (instances XCSP3 ou fichiers DIMACS CNF). On observe que les temps de génération sont du même ordre de grandeur que les modèles avec symétries avec une légère amélioration pour la plupart des instances. Les tables 8 et 9 présentent les temps de résolution des modèles M1+, M2+ et M3+. La table 10 compare les modèles sans symétries avec les modèles avec symétries (avec $k = 6$ pour les modèles de satisfaction).

La table 10 montre que M1+ a autant de mal que M1 pour la plupart des instances. M1+ n'est plus rapide que M1 que pour l'instance pores_1. M2+ et M3+ sont respectivement plus performants que M1 et M2 (M2 n'étant plus rapides que M2+ que sur 3 instances et M3 n'étant plus rapides que M3+ que sur 6 instances). Par contre, nous remarquons que cette amélioration est parfois négligeable pour M3+ (e.g. instances bcsstk22, fs_183_6, ibm32, will57).

La conclusion de cette comparaison c'est que, même sans symétries, M1+ (et M1) est le modèle avec les pires performances tandis que M2+ et M3+ fournissent systématiquement une solution en un temps plus ou moins raisonnable pour un k donné. M2+ est le modèle qui présente les meilleurs résultats. Cependant, il est important de rappeler que la comparaison des modèles d'optimisation (M1 et M1+) avec les modèles de satisfaction (M2, M2+, M3 et M3+) est assez injuste. En effet, l'intérêt des modèles d'optimisation est de trouver une ou plusieurs solutions optimales tandis que l'intérêt des modèles de satisfaction est de savoir s'il existe un étiquetage qui permet d'avoir une cyclic-bandwidth égale à une valeur k donnée.

Dans la prochaine section, nous présenterons un algorithme dichotomique parallèle utilisant un encadrement de la valeur k et les modèles de satisfaction pour déterminer la valeur k_{opt} optimale à l'image de M1 et M1+. Pour comparer équitablement cet algorithme avec M1+, nous fournirons au solveur ACE l'encadrement de la valeur k et nous préciserons au solveur d'utiliser un algorithme dichotomique pour effectuer l'optimisation plutôt qu'un algorithme de descente. Ces paramètres peuvent être transmis au solveur en ligne de commande grâce aux options suivantes :

- "-lb=<valeur>" la valeur de la borne inférieur de l'encadrement
- "-ub=<valeur>" la valeur de la borne supérieur de l'encadrement
- "-os=<algorithme>" le type d'algorithme a utilisé. Par défaut, le type d'algorithme est DECREASING (algorithme de descente), nous utilisons DICHOTOMIC (algorithme dichotomique)

4 Algorithme dichotomique parallèle

Il est possible d'encadrer la valeur k du cyclic-bandwidth [1, 2, 3]. L'encadrement (que l'on utilise dans notre algorithme dichotomique) est tel que $\lceil \Delta(G)/2 \rceil \leqslant k \leqslant \lfloor n/2 \rfloor$ avec $\Delta(G)$ le degré du graphe G et $n = |V(G)|$ le nombre de sommets de G. Le principe de notre algorithme dichotomique parallèle est détaillé ci-dessous. L'algorithme 1 complète notre description.

Principe :

1. Le degré $\Delta(G)$ du graphe G est calculé au préalable.
2. Les bornes que l'on considère sont $lb = \lceil \Delta(G)/2 \rceil$ et $ub = \lfloor n/2 \rfloor$
3. L'algorithme débute *nproc* processus. Un processus par valeur k testée. Les valeurs k testées lors de la première itération sont $k_{mid} = \dfrac{lb + ub}{2}$; $(nproc - 1)/2$ autres valeurs k choisies telles que $k \in [\![lb, k_{mid} - 1]\!]$ et $(nproc - 1)/2$ autres valeurs k choisies telles que $k \in [\![k_{mid} + 1, ub]\!]$. On garde un historique H des valeurs k testées pour ne pas tester une valeur plus d'une fois.
4. Pour chaque processus, on lance le modèle de satisfaction avec la valeur k attribuée au processus :
   - Si le modèle indique qu'il n'existe pas d'étiquetage de valeur k alors on arrête le processus et tout les processus tel que $k_p \geqslant k$ avec k_p les valeurs de la cyclic-bandwidth attribuées aux autres processus. On enregistre dans l'historique H toutes les valeurs k telles que $k \in [\![k, ub]\!]$.
   - Si le modèle indique qu'il existe un étiquetage de valeur k. On modifie la borne supérieur de l'encadrement tel que $ub = k$.
5. On considère d'autres valeurs de k telles que $k \notin H$ et $k \in [\![lb, ub]\!]$ pour remplacer les processus déjà terminés ou tués et on reprend l'étape 4 tant que $lb \neq ub$, $lb < ub$ et qu'on a pas testé toutes les valeurs entre lb et ub.

4.1 Comparaison des modèles avec les algorithmes dichotomiques

La Table 11 présentent les temps de résolution des algorithmes dichotomiques présentés lorsqu'ils sont utilisés avec les modèles sans symétries M1+, M2+ et M3+. On remarque que c'est M2+ qui a les meilleurs temps de résolution en général. En d'autres mots, si pour chaque modèle on calcule la somme des temps de résolution pour chaque instance, c'est M2+ qui présente la meilleure performance. De plus, sur certaines instances où les modèles M1+ ou M3+ sont plus rapides que M2+, la différence de performance est négligeable (e.g. bcspwr01, bcsstk01, ibm32, pores_1, will57). En comparaison, M1+ est plus rapide que les deux autres modèles pour 14 instances. Malgré cela, ce modèle est sujet à de nombreux dépassements du budget temps (TIMEOUT) et c'est d'ailleurs le seul à avoir dépasser le budget de 600 secondes que l'on a établit. D'un autre côté, M3+ surpasse M1+ et M2+ sur 6 instances mais, en général, ce modèle a tendance à prendre beaucoup de temps pour arriver à une solution. Nous suspectons que ces temps assez longs sont dû à un usage accru de la mémoire vive ou des écritures/lectures sur disque qui sont eux même dû à un grand nombre de clauses à traiter.

Algorithm 1: Algorithme dichotomique parallèle pour minimiser la cyclic-bandwidth

Data: G un graphe, Δ_G le degré du graphe, n le nombre de sommets de G et M le modèle à utiliser

$nproc \leftarrow$ nombre de processus à utiliser ; ;
/* Pour le calcul des K_i on suppose que l'on suit la 3ème étape du principe de l'algorithme */
$K_i \leftarrow$ valeurs attribuées à chaque processus i (avec $i \in \{1, \ldots, nproc\}$);
$lb \leftarrow \Delta_G/2$;
$ub \leftarrow n/2$;
$H \leftarrow \emptyset$; // Historique des valeurs déjà testées
/* Pour chaque processus i... */
while $lb \neq ub$, $lb < ub$ *et* $[\![lb, ub]\!] \nsubseteq H$ **do**
    $result \leftarrow M(K_i)$;
    if *result indique qu'il existe un étiquetage de valeur* K_i **then**
        $ub \leftarrow K_i$;
    else
        $kill(\{k \mid k \geqslant K_i\})$; // On tue tous les processus avec un k plus grand que K_i
        $H \leftarrow H \cup \{k \mid k > K_i\}$;
    end
    $H \leftarrow H \cup \{k_i\}$;
    $replaceProc(K, H)$; // On remplace les valeurs attribuées au processus tués ou terminés par des valeurs k qui n'ont pas encore été testées ($k \notin H$)
end

5 Conclusion

La façon de modéliser un problème donné est très importante parce que, comme le montre nos expérimentations, le type du modèle influence significativement le temps de résolution. Peaufiner les modèles en cassant des symétries ou en utilisant l'algorithme de dichotomie a servi dans la plupart des cas à obtenir de meilleurs temps de résolution. Ce projet nous a montré une autre façon de résoudre des problèmes d'optimisation. En effet, nous avons vu comment utiliser l'équivalence entre problème d'optimisation et problème de décision ainsi qu'un algorithme (la recherche dichotomique dans notre cas) pour obtenir des performances équivalentes à une modélisation purement axée sur l'optimisation.

C'est la déconstruction du problème originale et des premiers modèles qui a permit d'améliorer les modèles car sans celle-ci il est difficile de mieux choisir les contraintes et les variables sur lesquelles on travaille. Les analyses et les expérimentations nous ont montré qu'il est dans notre intérêt de ne pas nous contenter de la première version d'un modèle et qu'il peut être intéressant de casser des symétries, retirer des contraintes redondantes, bien choisir l'approche que l'on utilise, etc.

References

[1] Yixun Lin. « The Cyclic Bandwidth Problem. » In : *Chinese Science Abstracts Series A*. T. 2. 14. 1995, p. 14.

[2] Sanming Zhou. « Bounding the bandwidths for graphs ». In : *Theoretical computer science* 249.2 (2000), p. 357-368.

[3] Hugues Déprés, Guillaume Fertin et Eric Monfroy. « Improved Lower Bounds for the Cyclic Bandwidth Problem ». In : *International Conference on Computational Science*. Springer. 2021, p. 555-569.

A Annexe

Dans cet annexe nous présentons les résultats de toutes les expérimentations que nous avons conduit. La Table 1 présente les caractéristiques des instances de graphe du benchmark Harwell-Boeing 113. Les autres tables présentent les résultats interprétés section 2.1, 3.3 et 4.1.

Instance	**Nb sommets**	**Nb arêtes**		**Instance**	Nb sommets	Nb arêtes
arc130	130	715		gre_216a	216	660
ash85	85	219		gre_216b	216	660
bcspwr01	39	46		ibm32	32	90
bcspwr03	118	179		impcol_a	206	557
bcspwr04	274	669		impcol_c	137	352
bcsstk01	48	176		impcol_e	225	1187
bcsstk02	66	2145		lns_131	123	275
bcsstk04	132	1758		lund_a	147	1151
bcsstk05	153	1135		lund_b	147	1147
bcsstk22	110	254		mcca	168	1662
can_144	144	576		nnc261	261	794
can_161	161	608		nos1	158	312
curtis54	54	124		nos4	100	247
dwt_209	209	767		pores_1	30	103
dwt_234	117	162		saylr1	238	445
dwt_245	245	608		steam3	80	424
fs_183_1	183	701		west0132	132	404
fs_183_4	183	701		west0156	156	371
fs_183_6	183	701		west0167	167	489
gent113	104	549		will199	199	660
gre_115	115	267		will57	57	127
gre_185	185	650				
Instance	**Nb sommets**	**Nb arêtes**		**Instance**	**Nb sommets**	**Nb arêtes**

Table 1 – Caractéristiques des instances du benchmark Harwell-Boeing 113

	M1	M2			M3		
Instances		**k = 1**	**k = 2**	**k = 3**	**k = 1**	**k = 2**	**k = 3**
arc130	3.55	0.22	0.36	0.51	71.14	71.6	51.21
ash85	0.56	0.14	0.23	0.19	9.24	9.18	7.14
bcspwr01	0.08	0.03	0.02	0.06	0.44	0.43	0.32
bcspwr03	0.43	0.17	0.14	0.37	15.83	15.68	11.64
bcspwr04	3.97	0.92	1.11	2.88	380.78	380.13	284.03
bcsstk01	0.37	0.07	0.11	0.07	2.21	2.13	1.58
bcsstk02	19.35	0.79	1.46	1.09	51.68	51.37	37.22
bcsstk04	15.28	0.59	1.12	2.13	181.23	176.74	132.74
bcsstk05	5.3	0.77	1.73	1.62	189.59	191.05	131.8
bcsstk22	0.71	0.2	0.2	0.46	18.94	18.48	13.42
can___144	2.94	0.47	0.88	0.83	84.48	84.55	51.73
can___161	2.26	0.57	1.01	1.5	113.76	118.84	87.49
curtis54	0.25	0.07	0.05	0.14	2.14	2.12	1.54
dwt___209	5.0	0.9	1.68	1.32	247.89	246.52	196.48
dwt___234	0.34	0.15	0.22	0.35	14.38	14.06	10.71
dwt___245	2.23	0.56	1.65	1.61	281.54	279.54	209.62
fs_183_1	3.52	0.41	0.64	0.89	174.02	173.63	132.23
fs_183_4	2.48	0.44	1.36	1.89	174.04	171.65	128.51
fs_183_6	2.75	0.73	1.14	1.94	173.29	170.91	127.87
gent113	1.92	0.18	0.63	0.9	37.51	34.41	25.64
gre_216a	3.77	0.41	1.53	1.12	251.94	230.13	174.53
gre_216b	2.83	0.79	1.37	1.15	227.06	238.07	179.57
gre___115	0.76	0.11	0.18	0.24	21.87	21.57	15.91
gre___185	1.97	0.68	1.27	1.78	164.76	165.81	121.99
ibm32	0.16	0.03	0.06	0.05	0.46	0.45	0.33
impcol_a	1.46	0.27	0.46	0.6	181.15	211.17	134.92
impcol_c	1.23	0.32	0.54	0.76	42.91	45.46	29.17
impcol_e	9.06	1.15	1.73	2.19	445.12	454.29	349.24
lns___131	0.43	0.24	0.4	0.56	25.41	25.85	18.78
lund_a	5.84	0.68	1.69	2.49	169.59	172.44	112.61
lund_b	9.15	0.47	1.66	1.37	170.12	171.35	109.19
mcca	23.13	0.74	1.45	2.06	330.2	332.83	249.95
nnc261	5.14	1.11	1.97	2.4	419.2	412.54	322.97
nos1	0.99	0.31	0.56	0.81	58.51	57.83	44.43
nos4	0.69	0.18	0.29	0.42	14.71	14.75	10.56
pores_1	0.18	0.04	0.06	0.07	0.45	0.47	0.32
saylr1	1.88	0.39	1.19	1.38	193.76	197.23	152.35
steam3	1.69	0.23	0.38	0.55	17.88	15.36	12.25
west0132	1.53	0.16	0.26	0.35	42.63	43.15	30.45
west0156	0.79	0.35	0.66	0.57	67.57	68.67	53.74
west0167	1.2	0.5	0.87	0.65	103.81	101.1	79.94
will199	3.8	0.73	0.88	2.02	194.86	199.88	146.64
will57	0.24	0.04	0.11	0.15	2.45	2.47	1.84
Instances		**k = 1**	**k = 2**	**k = 3**	**k = 1**	**k = 2**	**k = 3**
	M1+	M2+			M3+		

TABLE 2 – Temps de compilation (en secondes) des modèles avec symétries (première partie)

	M1	M2			M3		
Instances		**k = 4**	**k = 5**	**k = 6**	**k = 4**	**k = 5**	**k = 6**
arc130	3.55	0.67	0.99	1.22	55.6	52.77	52.65
ash85	0.56	0.41	0.43	0.57	7.56	7.27	7.81
bcspwr01	0.08	0.06	0.03	0.05	0.35	0.34	0.34
bcspwr03	0.43	0.47	0.51	0.64	11.77	12.54	12.07
bcspwr04	3.97	1.94	4.04	5.25	301.62	294.3	300.8
bcsstk01	0.37	0.18	0.18	0.12	1.64	1.67	1.62
bcsstk02	19.35	1.42	2.81	3.8	40.99	39.52	39.57
bcsstk04	15.28	2.02	2.31	2.91	136.1	135.16	133.65
bcsstk05	5.3	2.91	1.98	4.76	134.99	134.2	133.79
bcsstk22	0.71	0.37	0.54	0.82	14.15	14.2	13.7
can___144	2.94	1.57	1.8	1.67	53.98	55.96	55.87
can___161	2.26	1.49	2.24	2.79	92.4	92.65	88.2
curtis54	0.25	0.17	0.09	0.23	1.63	1.59	1.72
dwt___209	5.0	2.67	1.82	4.55	198.49	195.16	191.7
dwt___234	0.34	0.44	0.36	0.6	10.98	10.75	10.56
dwt___245	2.23	1.69	2.08	4.27	223.47	219.18	223.96
fs_183_1	3.52	1.16	2.79	3.22	134.95	134.51	134.1
fs_183_4	2.48	2.46	2.81	3.58	134.9	132.11	132.98
fs_183_6	2.75	1.63	1.85	3.5	135.82	143.44	137.35
gent113	1.92	0.59	1.28	1.62	25.73	25.66	26.53
gre_216a	3.77	2.74	2.14	2.59	204.78	184.85	184.13
gre_216b	2.83	2.34	3.07	1.94	190.06	185.35	179.14
gre___115	0.76	0.31	0.34	0.45	16.01	15.88	15.84
gre___185	1.97	2.32	2.75	2.47	130.79	127.74	138.89
ibm32	0.16	0.04	0.05	0.11	0.34	0.34	0.33
impcol_a	1.46	0.78	1.15	1.57	139.38	137.2	139.37
impcol_c	1.23	1.02	1.12	1.44	30.45	30.11	32.61
impcol_e	9.06	5.13	5.53	4.87	362.0	367.98	356.41
lns___131	0.43	0.36	0.81	0.89	19.23	19.47	19.35
lund_a	5.84	1.81	2.61	2.38	108.96	113.8	114.56
lund_b	9.15	2.38	1.99	4.68	108.38	107.36	114.97
mcca	23.13	3.25	4.61	4.19	264.8	259.53	261.94
nnc261	5.14	3.59	4.43	4.93	324.82	352.36	352.0
nos1	0.99	0.66	1.18	1.45	45.49	45.39	44.77
nos4	0.69	0.51	0.58	0.75	11.55	11.7	11.58
pores_1	0.18	0.08	0.09	0.11	0.34	0.33	0.32
saylr1	1.88	2.1	2.38	2.99	159.21	150.43	150.06
steam3	1.69	0.69	0.81	0.97	11.98	12.4	11.77
west0132	1.53	0.43	0.49	1.15	32.5	32.24	32.59
west0156	0.79	0.66	1.31	1.42	53.36	55.68	55.01
west0167	1.2	1.07	1.82	1.52	78.13	85.41	87.39
will199	3.8	1.38	2.93	3.31	152.65	156.77	155.25
will57	0.24	0.17	0.2	0.24	1.95	1.86	1.88
Instances		**k = 4**	**k = 5**	**k = 6**	**k = 4**	**k = 5**	**k = 6**
	M1+	M2+			M3+		

TABLE 3 – Temps de compilation (en secondes) des modèles avec symétries (seconde partie)

	M1+	M2+			M3+		
Instances		**k = 1**	**k = 2**	**k = 3**	**k = 1**	**k = 2**	**k = 3**
arc130	2.25	0.25	0.42	0.61	64.10	65.79	70.00
ash85	0.26	0.13	0.10	0.16	8.99	9.24	8.82
bcspwr01	0.04	0.02	0.02	0.03	0.46	0.35	0.38
bcspwr03	0.25	0.08	0.12	0.28	14.76	14.73	14.19
bcspwr04	1.87	0.58	0.95	1.26	336.24	336.02	339.06
bcsstk01	0.18	0.07	0.05	0.07	2.11	2.00	1.95
bcsstk02	14.01	0.39	0.63	1.02	53.34	52.58	54.43
bcsstk04	9.45	0.44	0.78	1.19	149.89	139.93	139.79
bcsstk05	5.06	0.42	0.79	1.18	168.59	170.32	177.38
bcsstk22	0.36	0.09	0.15	0.20	17.85	17.00	16.42
can___144	1.61	0.25	0.40	0.54	65.62	66.01	65.47
can___161	1.58	0.27	0.49	0.64	111.33	112.40	111.91
curtis54	0.12	0.03	0.05	0.06	1.95	1.99	1.94
dwt___209	2.45	0.49	0.71	1.15	258.14	273.63	252.86
dwt___234	0.22	0.07	0.12	0.15	12.22	12.56	12.61
dwt___245	1.53	0.43	0.74	1.03	272.90	285.76	279.10
fs_183_1	2.03	0.35	0.64	0.82	167.85	169.32	175.03
fs_183_4	2.03	0.44	0.58	0.93	172.92	171.34	166.22
fs_183_6	2.03	0.41	0.60	0.82	182.97	167.42	167.20
gent113	1.36	0.16	0.31	0.40	30.96	31.14	31.72
gre_216a	1.83	0.42	0.65	1.01	238.93	233.98	240.78
gre_216b	1.99	0.43	0.65	1.07	231.24	225.53	226.78
gre___115	0.36	0.10	0.18	0.22	19.45	18.56	18.54
gre___185	1.79	0.43	0.54	0.79	171.00	165.20	159.59
ibm32	0.08	0.02	0.05	0.04	0.44	0.41	0.40
impcol_a	1.32	0.38	0.55	0.80	178.43	180.20	175.66
impcol_c	0.74	0.19	0.33	0.42	39.15	40.22	38.85
impcol_e	5.73	0.64	1.20	1.78	471.53	448.82	441.85
lns___131	0.42	0.16	0.18	0.24	24.40	23.32	25.88
lund_a	5.19	0.47	0.83	1.20	141.49	136.35	145.07
lund_b	5.34	0.49	0.78	1.09	138.27	141.20	142.76
mcca	9.32	0.63	1.12	1.82	318.28	304.18	306.18
nnc261	2.57	0.54	1.01	1.52	403.97	414.91	404.48
nos1	0.48	0.16	0.30	0.35	59.90	56.85	57.91
nos4	0.34	0.09	0.13	0.18	14.75	14.43	14.66
pores_1	0.09	0.02	0.03	0.03	0.40	0.41	0.41
saylr1	0.88	0.29	0.53	0.72	175.29	180.43	174.80
steam3	0.85	0.11	0.18	0.23	15.87	15.60	16.00
west0132	0.59	0.13	0.23	0.30	39.29	37.91	38.76
west0156	0.67	0.17	0.28	0.40	66.15	66.80	67.23
west0167	1.05	0.38	0.38	0.53	123.41	102.81	100.12
will199	1.88	0.36	0.61	0.85	189.34	193.30	187.70
will57	0.13	0.03	0.05	0.08	2.36	2.34	2.33
Instances		**k = 1**	**k = 2**	**k = 3**	**k = 1**	**k = 2**	**k = 3**
	M1+	M2+			M3+		

TABLE 4 – Temps de compilation (en secondes) des modèles sans symétries (première partie)

	M1+	M2+			M3+		
Instances		**k = 4**	**k = 5**	**k = 6**	**k = 4**	**k = 5**	**k = 6**
arc130	2.25	0.79	1.10	1.16	65.35	67.69	67.33
ash85	0.26	0.17	0.21	0.24	8.93	8.93	9.31
bcspwr01	0.04	0.03	0.03	0.04	0.39	0.38	0.39
bcspwr03	0.25	0.26	0.26	0.29	14.38	14.61	14.46
bcspwr04	1.87	1.64	1.98	2.55	335.56	337.06	343.23
bcsstk01	0.18	0.10	0.10	0.12	2.01	1.97	2.71
bcsstk02	14.01	1.19	1.45	1.87	55.20	53.23	53.95
bcsstk04	9.45	1.48	1.87	2.18	134.64	138.52	134.02
bcsstk05	5.06	1.51	1.85	2.37	166.43	169.00	166.81
bcsstk22	0.36	0.25	0.33	0.36	16.95	16.85	17.27
can___144	1.61	0.74	0.97	1.09	66.27	66.92	70.17
can___161	1.58	0.85	1.07	1.29	109.13	124.29	111.65
curtis54	0.12	0.09	0.08	0.10	1.96	1.97	2.13
dwt___209	2.45	1.51	1.69	1.97	253.68	246.59	289.20
dwt___234	0.22	0.19	0.26	0.26	11.98	12.00	12.64
dwt___245	1.53	1.28	1.60	1.92	272.97	271.25	278.48
fs_183_1	2.03	1.13	1.41	1.60	178.61	170.45	183.99
fs_183_4	2.03	1.12	1.33	1.74	164.97	165.34	176.84
fs_183_6	2.03	1.14	1.43	1.61	166.00	167.69	167.07
gent113	1.36	0.57	0.60	0.71	31.50	31.38	31.36
gre_216a	1.83	1.30	1.56	1.82	231.52	225.14	223.07
gre_216b	1.99	1.24	1.55	1.91	226.20	224.64	263.49
gre___115	0.36	0.28	0.34	0.43	18.63	19.12	19.28
gre___185	1.79	1.01	1.25	1.64	156.72	163.75	156.93
ibm32	0.08	0.05	0.04	0.05	0.43	0.39	0.78
impcol_a	1.32	1.01	1.32	1.48	182.46	179.22	179.68
impcol_c	0.74	0.53	0.53	0.71	38.31	38.47	41.35
impcol_e	5.73	2.34	2.80	3.33	443.40	447.18	477.10
lns___131	0.42	0.31	0.37	0.47	22.79	23.17	22.64
lund_a	5.19	1.40	1.85	2.04	137.33	137.32	135.49
lund_b	5.34	1.39	1.72	2.12	137.71	138.68	163.08
mcca	9.32	2.21	2.73	3.25	303.08	302.83	318.19
nnc261	2.57	1.84	2.31	2.81	401.44	402.19	416.41
nos1	0.48	0.48	0.57	0.64	58.66	55.45	57.80
nos4	0.34	0.23	0.30	0.34	15.69	15.16	15.12
pores_1	0.09	0.04	0.05	0.10	0.40	0.37	0.39
saylr1	0.88	0.94	1.13	1.38	177.51	178.44	175.78
steam3	0.85	0.30	0.36	0.43	15.90	16.08	15.97
west0132	0.59	0.39	0.54	0.58	40.34	40.44	39.05
west0156	0.67	0.52	0.63	0.72	66.31	65.68	66.84
west0167	1.05	0.71	0.93	1.09	102.32	101.38	105.89
will199	1.88	1.17	1.50	1.76	192.82	189.13	187.82
will57	0.13	0.11	0.09	0.12	2.27	2.37	2.61
Instances		**k = 4**	**k = 5**	**k = 6**	**k = 4**	**k = 5**	**k = 6**
	M1+	M2+			M3+		

Table 5 – Temps de compilation (en secondes) des modèles sans symétries (première partie)

	M1	M2			M3		
Instances		**k = 1**	**k = 2**	**k = 3**	**k = 1**	**k = 2**	**k = 3**
arc130	TIMEOUT	1.31	1.54	1.50	22.64	21.58	22.32
ash85	TIMEOUT	1.32	1.30	1.53	2.66	2.91	2.64
bcspwr01	2.99	1.15	1.18	1.22	0.14	0.14	0.14
bcspwr03	TIMEOUT	1.40	1.49	1.76	4.43	5.21	5.28
bcspwr04	TIMEOUT	1.99	2.10	2.38	96.86	103.23	103.62
bcsstk01	TIMEOUT	1.17	1.24	1.25	0.72	0.69	0.67
bcsstk02	TIMEOUT	1.50	1.55	1.52	17.25	16.92	14.93
bcsstk04	TIMEOUT	1.61	1.68	1.69	55.95	56.33	54.72
bcsstk05	TIMEOUT	1.64	1.82	1.83	50.24	50.63	49.41
bcsstk22	TIMEOUT	1.48	1.44	1.57	5.85	5.75	5.33
can___144	TIMEOUT	1.54	1.53	1.69	24.05	20.13	22.66
can___161	TIMEOUT	1.54	1.24	1.64	29.78	30.52	29.02
curtis54	TIMEOUT	1.26	1.30	1.25	0.63	0.67	0.63
dwt___209	TIMEOUT	1.65	1.26	1.54	66.22	66.92	64.53
dwt___234	TIMEOUT	1.30	1.30	1.43	4.01	4.03	4.53
dwt___245	TIMEOUT	1.87	2.12	2.28	72.13	74.52	72.70
fs_183_1	TIMEOUT	1.54	1.59	0.85	44.42	45.70	44.58
fs_183_4	TIMEOUT	1.52	1.58	1.59	45.71	43.72	46.10
fs_183_6	TIMEOUT	1.51	1.62	1.66	45.32	46.29	45.77
gent113	TIMEOUT	1.37	1.49	1.50	10.88	10.99	9.75
gre_216a	TIMEOUT	1.65	2.09	2.17	60.59	62.22	61.45
gre_216b	TIMEOUT	1.75	2.05	2.37	61.86	59.77	61.08
gre___115	TIMEOUT	1.38	1.38	1.47	6.92	6.03	7.26
gre___185	TIMEOUT	1.68	1.65	1.82	45.74	43.71	44.25
ibm32	TIMEOUT	1.12	1.23	1.09	0.16	0.16	0.16
impcol_a	TIMEOUT	1.61	1.83	1.82	47.87	47.89	47.16
impcol_c	TIMEOUT	1.44	1.55	1.47	13.37	11.48	13.27
impcol_e	TIMEOUT	1.87	1.72	1.86	114.14	112.90	114.35
lns___131	TIMEOUT	1.40	1.53	1.55	7.72	8.56	8.04
lund_a	TIMEOUT	1.52	1.77	1.73	44.42	47.60	45.29
lund_b	TIMEOUT	1.59	1.84	1.80	43.83	46.71	44.80
mcca	TIMEOUT	1.78	1.90	2.02	86.44	88.95	87.54
nnc261	TIMEOUT	2.00	2.60	3.12	106.57	107.34	106.48
nos1	TIMEOUT	2.56	2.40	24.75	14.68	15.05	14.47
nos4	TIMEOUT	1.33	1.44	1.52	4.53	4.17	4.88
pores_1	108.15	1.11	1.20	1.21	0.16	0.15	0.16
saylr1	TIMEOUT	1.90	2.87	3.91	54.38	51.63	51.41
steam3	TIMEOUT	1.36	1.46	1.43	5.23	5.22	4.76
west0132	TIMEOUT	1.43	1.48	1.49	12.50	13.85	13.89
west0156	TIMEOUT	1.38	1.57	1.60	18.43	16.63	15.15
west0167	TIMEOUT	1.33	1.55	1.51	27.41	23.61	23.30
will199	TIMEOUT	1.57	1.59	1.72	52.14	53.82	51.93
will57	33.95	1.28	1.34	1.37	0.73	0.77	0.71
Instances		**k = 1**	**k = 2**	**k = 3**	**k = 1**	**k = 2**	**k = 3**
	M1	M2			M3		

TABLE 6 – Temps de résolution (en secondes) des modèles avec symétries (première partie)

	M1	M2			M3		
Instances		**k = 4**	**k = 5**	**k = 6**	**k = 4**	**k = 5**	**k = 6**
arc130	TIMEOUT	1.66	1.77	1.51	23.49	19.88	22.57
ash85	TIMEOUT	1.80	1.63	1.54	2.96	2.92	2.77
bcspwr01	2.99	1.43	1.38	1.15	0.15	0.14	0.15
bcspwr03	TIMEOUT	1.68	1.93	1.86	5.32	5.32	5.39
bcspwr04	TIMEOUT	3.12	3.87	3.54	101.98	104.37	107.59
bcsstk01	TIMEOUT	1.32	1.43	1.29	0.72	0.67	0.81
bcsstk02	TIMEOUT	1.50	1.78	1.65	14.44	17.11	17.66
bcsstk04	TIMEOUT	1.78	1.92	1.68	55.37	55.41	56.60
bcsstk05	TIMEOUT	1.78	2.13	2.22	49.98	50.57	51.38
bcsstk22	TIMEOUT	1.69	2.50	8.43	6.17	5.62	5.52
can___144	TIMEOUT	1.83	2.03	1.92	21.97	23.08	24.02
can___161	TIMEOUT	1.86	2.10	2.20	30.28	29.00	30.59
curtis54	TIMEOUT	1.31	1.42	1.32	0.70	0.63	0.65
dwt___209	TIMEOUT	2.02	2.42	2.32	66.17	65.82	69.15
dwt___234	TIMEOUT	1.49	1.76	1.55	4.07	4.07	4.31
dwt___245	TIMEOUT	2.38	3.22	2.85	69.90	74.15	78.91
fs_183_1	TIMEOUT	1.77	1.95	1.76	44.87	46.30	47.72
fs_183_4	TIMEOUT	1.62	1.93	1.68	45.54	44.69	48.74
fs_183_6	TIMEOUT	1.80	2.04	1.86	45.72	45.14	47.59
gent113	TIMEOUT	1.61	1.66	1.57	10.45	11.45	11.22
gre_216a	TIMEOUT	2.54	2.69	3.18	61.49	59.75	61.20
gre_216b	TIMEOUT	2.65	3.99	1.71	58.63	61.83	63.19
gre___115	TIMEOUT	1.49	1.75	1.76	6.76	7.28	6.80
gre___185	TIMEOUT	2.26	2.81	2.42	42.80	43.58	43.93
ibm32	TIMEOUT	1.26	1.31	1.23	0.16	0.16	0.17
impcol_a	TIMEOUT	2.00	2.83	2.62	48.75	46.08	48.47
impcol_c	TIMEOUT	1.61	1.90	1.64	12.38	12.89	11.59
impcol_e	TIMEOUT	2.12	2.14	2.04	112.82	119.50	119.47
lns___131	TIMEOUT	1.58	1.76	1.26	7.20	8.13	7.40
lund_a	TIMEOUT	1.95	2.13	1.94	47.14	46.45	48.39
lund_b	TIMEOUT	2.10	2.08	1.89	46.45	46.85	47.00
mcca	TIMEOUT	2.23	2.41	2.09	88.27	89.19	88.17
nnc261	TIMEOUT	3.66	4.51	4.41	108.91	108.68	112.18
nos1	TIMEOUT	398.75	3.20	588.99	15.91	15.66	15.54
nos4	TIMEOUT	0.90	1.74	1.69	4.17	4.64	4.73
pores_1	108.15	1.17	1.54	2.05	0.16	0.16	0.17
saylr1	TIMEOUT	4.40	5.24	5.90	49.46	51.04	53.68
steam3	TIMEOUT	1.38	1.64	2.15	4.35	4.38	4.56
west0132	TIMEOUT	1.23	1.79	1.56	13.50	12.30	14.07
west0156	TIMEOUT	1.71	1.91	1.60	19.12	16.36	18.45
west0167	TIMEOUT	1.56	1.77	1.50	25.60	27.18	28.89
will199	TIMEOUT	1.56	1.86	1.92	52.26	52.37	53.50
will57	33.95	1.51	1.54	1.47	0.71	0.71	0.74
Instances		**k = 4**	**k = 5**	**k = 6**	**k = 4**	**k = 5**	**k = 6**
	M1	M2			M3		

Table 7 – Temps de résolution (en secondes) des modèles avec symétries (seconde partie)

	M1+	M2+			M3+		
Instances		**k = 1**	**k = 2**	**k = 3**	**k = 1**	**k = 2**	**k = 3**
arc130	TIMEOUT	1.39	1.36	1.41	21.21	21.98	21.00
ash85	TIMEOUT	1.23	1.24	1.29	2.85	2.53	2.53
bcspwr01	3.94	1.14	1.18	1.21	0.12	0.12	0.13
bcspwr03	TIMEOUT	1.27	1.15	1.26	4.38	5.07	4.45
bcspwr04	TIMEOUT	1.47	1.40	1.47	94.51	93.76	98.53
bcsstk01	TIMEOUT	1.21	1.23	1.19	0.66	0.65	0.65
bcsstk02	TIMEOUT	1.49	1.47	1.46	16.82	16.38	14.27
bcsstk04	TIMEOUT	1.47	1.54	1.49	55.50	54.91	56.10
bcsstk05	TIMEOUT	1.36	1.51	1.44	47.42	50.56	48.54
bcsstk22	TIMEOUT	1.27	1.27	1.28	4.99	5.69	5.66
can___144	TIMEOUT	1.30	1.42	1.38	21.77	21.98	21.81
can___161	TIMEOUT	1.38	1.33	1.40	29.49	27.60	29.51
curtis54	TIMEOUT	1.16	1.22	1.23	0.58	0.59	0.59
dwt___209	TIMEOUT	1.44	1.45	1.44	62.90	63.27	62.07
dwt___234	TIMEOUT	1.25	1.22	1.24	3.75	3.72	3.70
dwt___245	TIMEOUT	1.34	1.41	1.42	69.83	69.21	72.99
fs_183_1	TIMEOUT	1.38	1.39	1.43	42.54	44.30	43.24
fs_183_4	TIMEOUT	1.38	1.41	1.41	44.12	43.78	42.57
fs_183_6	TIMEOUT	1.41	1.36	1.40	44.38	42.77	43.50
gent113	TIMEOUT	1.30	1.38	1.33	10.52	10.45	11.28
gre_216a	TIMEOUT	1.42	1.42	1.49	58.57	58.87	56.04
gre_216b	TIMEOUT	1.42	1.35	1.47	60.44	57.32	57.59
gre___115	TIMEOUT	1.27	1.26	1.29	6.42	5.70	5.68
gre___185	TIMEOUT	1.33	1.42	1.47	41.32	41.48	40.20
ibm32	TIMEOUT	1.18	1.20	1.18	0.15	0.15	0.15
impcol_a	TIMEOUT	1.38	1.41	1.47	44.37	44.21	44.00
impcol_c	TIMEOUT	1.32	1.30	1.31	12.77	12.23	10.60
impcol_e	TIMEOUT	1.46	1.47	1.55	111.18	111.78	112.87
lns___131	TIMEOUT	1.30	1.25	1.31	6.69	8.01	7.82
lund_a	TIMEOUT	1.46	1.43	1.42	44.87	45.28	47.69
lund_b	TIMEOUT	1.44	1.41	1.44	45.54	45.16	45.55
mcca	TIMEOUT	1.51	1.54	1.53	85.53	85.32	85.70
nnc261	TIMEOUT	1.48	1.53	1.45	102.78	103.49	101.45
nos1	TIMEOUT	1.31	1.31	3.22	13.52	14.61	14.66
nos4	TIMEOUT	1.21	1.26	1.25	3.97	4.62	4.65
pores_1	16.72	1.17	1.19	1.12	0.15	0.15	0.15
saylr1	TIMEOUT	1.40	1.40	1.42	49.33	48.36	48.27
steam3	TIMEOUT	1.31	1.29	1.27	4.71	4.95	4.39
west0132	TIMEOUT	1.33	1.35	1.30	12.72	13.06	12.64
west0156	TIMEOUT	1.32	1.32	1.37	17.47	17.49	16.76
west0167	TIMEOUT	1.35	1.34	1.39	26.50	26.30	26.08
will199	TIMEOUT	1.42	1.38	1.45	49.23	46.44	49.48
will57	58.95	1.23	1.20	1.21	0.73	0.75	0.67
Instances		**k = 1**	**k = 2**	**k = 3**	**k = 1**	**k = 2**	**k = 3**
	M1+	M2+			M3+		

Table 8 – Temps de résolution (en secondes) des modèles sans symétries (première partie)

	M1+	M2+			M3+		
Instances		**k = 4**	**k = 5**	**k = 6**	**k = 4**	**k = 5**	**k = 6**
arc130	TIMEOUT	1.35	1.32	1.43	20.89	21.57	21.62
ash85	TIMEOUT	1.31	1.34	1.34	2.54	2.54	2.54
bcspwr01	3.94	1.27	1.22	1.20	0.13	0.13	0.12
bcspwr03	TIMEOUT	1.26	1.32	1.28	4.16	4.41	4.94
bcspwr04	TIMEOUT	1.43	1.50	1.56	96.24	96.97	95.62
bcsstk01	TIMEOUT	1.23	1.22	1.19	0.73	0.64	0.65
bcsstk02	TIMEOUT	1.50	1.51	1.60	17.96	16.41	16.80
bcsstk04	TIMEOUT	1.50	1.61	1.42	57.20	55.99	56.45
bcsstk05	TIMEOUT	1.45	1.46	1.52	49.96	48.62	48.62
bcsstk22	TIMEOUT	1.24	1.51	11.16	5.47	4.97	5.56
can___144	TIMEOUT	1.33	1.37	0.98	21.12	22.31	22.75
can___161	TIMEOUT	1.43	1.37	1.42	29.46	26.63	29.96
curtis54	TIMEOUT	1.23	1.19	1.25	0.61	0.58	0.58
dwt___209	TIMEOUT	1.47	1.48	1.47	65.58	61.51	63.24
dwt___234	TIMEOUT	1.22	1.27	1.25	3.73	4.35	3.70
dwt___245	TIMEOUT	1.38	1.44	1.46	68.70	70.09	69.61
fs_183_1	TIMEOUT	1.46	1.43	1.46	43.02	43.12	44.30
fs_183_4	TIMEOUT	1.44	1.48	1.45	43.78	42.36	43.57
fs_183_6	TIMEOUT	1.47	1.38	1.46	43.89	43.45	48.08
gent113	TIMEOUT	1.38	1.31	1.32	11.18	10.56	10.91
gre_216a	TIMEOUT	1.44	1.53	1.57	59.01	56.02	59.37
gre_216b	TIMEOUT	1.46	1.51	1.55	58.76	59.15	58.69
gre___115	TIMEOUT	1.32	1.30	1.28	6.07	5.75	5.96
gre___185	TIMEOUT	1.39	1.49	1.39	41.45	37.79	41.64
ibm32	TIMEOUT	1.18	1.21	1.29	0.15	0.16	0.16
impcol_a	TIMEOUT	1.46	1.49	1.47	44.46	44.48	43.57
impcol_c	TIMEOUT	1.33	1.37	1.31	11.63	12.66	12.46
impcol_e	TIMEOUT	1.52	1.53	1.63	113.35	116.76	109.73
lns___131	TIMEOUT	1.25	1.31	1.31	6.72	7.13	7.62
lund_a	TIMEOUT	1.50	1.49	1.44	46.09	46.01	43.14
lund_b	TIMEOUT	1.44	1.45	1.44	45.91	45.22	44.93
mcca	TIMEOUT	1.58	1.62	1.50	85.79	85.94	89.71
nnc261	TIMEOUT	1.50	1.60	1.51	106.79	102.33	99.47
nos1	TIMEOUT	415.25	5.45	336.92	14.54	14.95	13.52
nos4	TIMEOUT	1.24	1.31	1.28	4.17	4.25	4.55
pores_1	16.72	1.18	1.22	1.28	0.15	0.15	0.16
saylr1	TIMEOUT	1.44	1.50	0.73	49.27	48.93	48.28
steam3	TIMEOUT	1.26	1.25	1.59	4.27	4.73	4.75
west0132	TIMEOUT	1.10	1.33	1.40	13.19	12.31	12.82
west0156	TIMEOUT	0.85	1.36	1.35	14.90	15.42	16.38
west0167	TIMEOUT	1.36	1.38	1.28	25.28	26.89	26.22
will199	TIMEOUT	1.41	1.43	1.44	48.74	47.81	49.28
will57	58.95	1.29	1.38	1.39	0.67	0.67	0.73
Instances		**k = 4**	**k = 5**	**k = 6**	**k = 4**	**k = 5**	**k = 6**
	M1+	M2+			M3+		

Table 9 – Temps de résolution (en secondes) des modèles sans symétries (seconde partie)

Instance	$\lvert V(G)\rvert$	$\lvert E(G)\rvert$	**M1**	**M1+**	**M2**	**M2+**	**M3**	**M3+**
arc130	130	715	TIMEOUT	TIMEOUT	1.51	1.43	22.57	21.62
ash85	85	219	TIMEOUT	TIMEOUT	1.54	1.34	2.77	2.54
bcspwr01	39	46	2.99	3.94	1.15	1.20	0.15	0.12
bcspwr03	118	179	TIMEOUT	TIMEOUT	1.86	1.28	5.39	4.94
bcspwr04	274	669	TIMEOUT	TIMEOUT	3.54	1.56	107.59	95.62
bcsstk01	48	176	TIMEOUT	TIMEOUT	1.29	1.19	0.81	0.65
bcsstk02	66	2145	TIMEOUT	TIMEOUT	1.65	1.60	17.66	16.80
bcsstk04	132	1758	TIMEOUT	TIMEOUT	1.68	1.42	56.60	56.45
bcsstk05	153	1135	TIMEOUT	TIMEOUT	2.22	1.52	51.38	48.62
bcsstk22	110	254	TIMEOUT	TIMEOUT	8.43	11.16	5.52	5.56
can_144	144	576	TIMEOUT	TIMEOUT	1.92	0.98	24.02	22.75
can_161	161	608	TIMEOUT	TIMEOUT	2.20	1.42	30.59	29.96
curtis54	54	124	TIMEOUT	TIMEOUT	1.32	1.25	0.65	0.58
dwt_209	209	767	TIMEOUT	TIMEOUT	2.32	1.47	69.15	63.24
dwt_234	117	162	TIMEOUT	TIMEOUT	1.55	1.25	4.31	3.70
dwt_245	245	608	TIMEOUT	TIMEOUT	2.85	1.46	78.91	69.61
fs_183_1	183	701	TIMEOUT	TIMEOUT	1.76	1.46	47.72	44.30
fs_183_4	183	701	TIMEOUT	TIMEOUT	1.68	1.45	48.74	43.57
fs_183_6	183	701	TIMEOUT	TIMEOUT	1.86	1.46	47.59	48.08
gent113	104	549	TIMEOUT	TIMEOUT	1.57	1.32	11.22	10.91
gre_115	115	267	TIMEOUT	TIMEOUT	3.18	1.57	61.20	59.37
gre_185	185	650	TIMEOUT	TIMEOUT	1.71	1.55	63.19	58.69
gre_216a	216	660	TIMEOUT	TIMEOUT	1.76	1.28	6.80	5.96
gre_216b	216	660	TIMEOUT	TIMEOUT	2.42	1.39	43.93	41.64
ibm32	32	90	TIMEOUT	TIMEOUT	1.23	1.29	0.17	0.16
impcol_a	206	557	TIMEOUT	TIMEOUT	2.62	1.47	48.47	43.57
impcol_c	137	352	TIMEOUT	TIMEOUT	1.64	1.31	11.59	12.46
impcol_e	225	1187	TIMEOUT	TIMEOUT	2.04	1.63	119.47	109.73
lns_131	123	275	TIMEOUT	TIMEOUT	1.26	1.31	7.40	7.62
lund_a	147	1151	TIMEOUT	TIMEOUT	1.94	1.44	48.39	43.14
lund_b	147	1147	TIMEOUT	TIMEOUT	1.89	1.44	47.00	44.93
mcca	168	1662	TIMEOUT	TIMEOUT	2.09	1.50	88.17	89.71
nnc261	261	794	TIMEOUT	TIMEOUT	4.41	1.51	112.18	99.47
nos1	158	312	TIMEOUT	TIMEOUT	588.99	336.92	15.54	13.52
nos4	100	247	TIMEOUT	TIMEOUT	1.69	1.28	4.73	4.55
pores_1	30	103	108.15	16.72	2.05	1.28	0.17	0.16
saylr1	238	445	TIMEOUT	TIMEOUT	5.90	0.73	53.68	48.28
steam3	80	424	TIMEOUT	TIMEOUT	2.15	1.59	4.56	4.75
west0132	132	404	TIMEOUT	TIMEOUT	1.56	1.40	14.07	12.82
west0156	156	371	TIMEOUT	TIMEOUT	1.60	1.35	18.45	16.38
west0167	167	489	TIMEOUT	TIMEOUT	1.50	1.28	28.89	26.22
will199	199	660	TIMEOUT	TIMEOUT	1.92	1.44	53.50	49.28
will57	57	127	33.95	58.95	1.47	1.39	0.74	0.73
Instance	$\lvert V(G)\rvert$	$\lvert E(G)\rvert$	**M1**	**M1+**	**M2**	**M2+**	**M3**	**M3+**

TABLE 10 – Comparaison entre les modèles avec symétries et les modèles sans symétries (avec $k = 6$ pour les modèles de satisfaction, $|V(G)|$ le nombre de sommet de l'instance et $|E(G)|$ le nombre d'arêtes de l'instance)

Instances	**Nb sommets**	**Nb arêtes**	**M1+**	**M2+**	**M3+**
arc130	130	715	TIMEOUT	123.05	110.93
ash85	85	219	TIMEOUT	4.71	18.91
bcspwr01	39	46	1.29	4.62	1.08
bcspwr03	118	179	3.76	4.50	21.36
bcspwr04	274	669	127.10	34.80	514.27
bcsstk01	48	176	2.27	2.96	3.59
bcsstk02	66	2145	TIMEOUT	0.03	0.00
bcsstk04	132	1758	TIMEOUT	254.25	209.23
bcsstk05	153	1135	TIMEOUT	25.93	207.14
bcsstk22	110	254	4.14	16.71	25.67
can___144	144	576	190.55	57.35	103.32
can___161	161	608	TIMEOUT	8.05	140.22
curtis54	54	124	2.00	5.84	4.96
dwt___209	209	767	TIMEOUT	12.82	286.08
dwt___234	117	162	3.72	3.92	20.76
dwt___245	245	608	4.08	11.41	378.42
fs_183_1	183	701	TIMEOUT	12.70	277.53
fs_183_4	183	701	TIMEOUT	116.70	214.80
fs_183_6	183	701	TIMEOUT	159.00	211.44
gent113	104	549	21.54	16.58	55.78
gre_216a	115	267	341.69	11.44	310.47
gre_216b	185	650	544.82	3.83	260.64
gre___115	216	660	TIMEOUT	24.45	43.27
gre___185	216	660	32.85	13.66	229.28
ibm32	32	90	TIMEOUT	4.54	0.91
impcol_a	206	557	88.93	23.13	221.54
impcol_c	137	352	20.34	33.20	58.17
impcol_e	225	1187	16.82	38.26	527.66
lns___131	123	275	5.49	5.60	42.43
lund_a	147	1151	4.37	26.68	177.97
lund_b	147	1147	4.57	17.20	207.50
mcca	168	1662	75.12	324.80	393.37
nnc261	261	794	TIMEOUT	4.43	474.46
nos1	158	312	2.56	4.62	81.83
nos4	100	247	88.14	12.95	20.95
pores_1	30	103	1.97	3.75	1.20
saylr1	238	445	135.29	18.23	225.20
steam3	80	424	164.39	19.62	25.23
west0132	132	404	TIMEOUT	38.99	62.40
west0156	156	371	TIMEOUT	5.27	99.45
west0167	167	489	264.50	12.47	144.23
will199	199	660	TIMEOUT	24.65	275.90
will57	57	127	2.06	5.51	3.59
Instances	**Nb sommets**	**Nb arêtes**	**M1+**	**M2+**	**M3+**

Table 11 – Temps de résolution (en secondes) des modèles sans symétries avec les algorithmes dichotomiques (l'algorithme utilisé pour M2+ et M3+ est parallélisé entre 5 processus)