Initiation à la Recherche : lecture d'article

HEURISTICS FOR A MATRIX SYMMETRIZATION PROBLEM

Article de

Bora UÇAR

Présenté par

Mohamed Amine EL AFRIT

PLAN

- Contexte Générale et problématique
- Principe de l’algorithme
- Anomalies
- Améliorations
- Résultats et expériences

PLAN

- Contexte Générale et problématique
- Principe de l’algorithme
- Anomalies
- Améliorations
- Résultats et expériences

Contexte général

- Problème de **partitionnement de graphe** et **d'ordonnancement** (*réf :* [8,10,16])

- Les **solveurs numériques parallèles** utilisent l'arbre d'élimination qui n'est définie que pour une matrice symétrique (*réf :* [17])

→ complétion $A+A^T$

→ Augmentation des cout de calcul

→On cherche à augmenter la symétrie dans une matrice

Problématique

Décider si une matrice est symétrisable

NP-complet

Matrice
- **Carré**
- **Non symétrique**
- **A valeur dans {0,1}**

Trouver une permutation des colonnes
- **Maximiser la symétrie**
- **Diagonale non nulle**

NP-hard

Matrice
- **Carré**
- **Plus symétrique**
- **A valeur dans {0,1}**
- **Diagonale non nulle**

On s'intéresse à **optimiser** le problème de symétrisation d'une matrice carré non symétrique à valeur dans {0,1}

PLAN

- **Contexte Générale et problématique**
- **Principe de l'algorithme**
- **Anomalies**
- **Améliorations**
- **Résultats et expériences**

Principe de l'algorithme

Exemple de matrice à symétriser

$$
A = \begin{array}{c|cccccc}
 & C1 & C2 & C3 & C4 & C5 & C6 \\
\hline
R1 & 1 & 1 & 0 & 1 & 0 & 0 \\
R2 & 1 & 0 & 1 & 0 & 1 & 1 \\
R3 & 1 & 0 & 0 & 1 & 1 & 0 \\
R4 & 1 & 1 & 0 & 1 & 1 & 1 \\
R5 & 0 & 0 & 0 & 1 & 1 & 0 \\
R6 & 1 & 0 & 0 & 1 & 0 & 0
\end{array}
$$

$$S(A) = \sum_{a_{ij} \neq 0} a_{ij} a_{ji} = 11$$

Principe de l'algorithme

Construction du graphe biparti

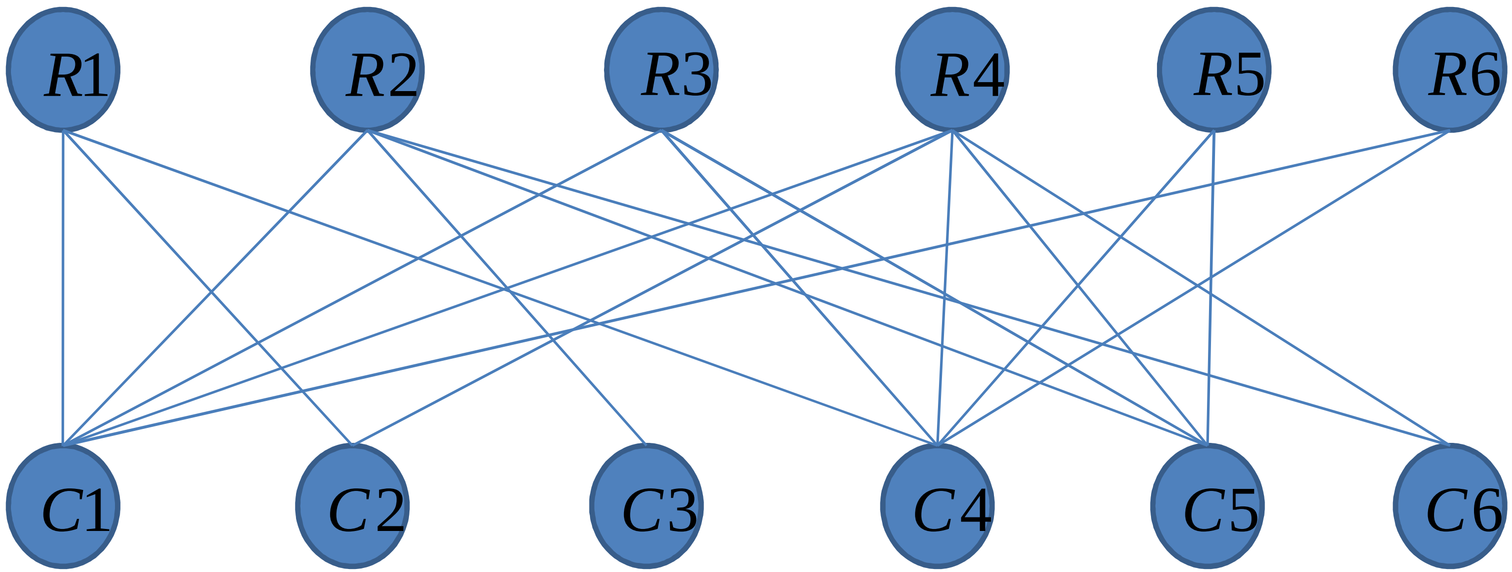
$$A = \begin{array}{c|cccccc} & C1 & C2 & C3 & C4 & C5 & C6 \\ \hline R1 & 1 & 1 & 0 & 1 & 0 & 0 \\ R2 & 1 & 0 & 1 & 0 & 1 & 1 \\ R3 & 1 & 0 & 0 & 1 & 1 & 0 \\ R4 & 1 & 1 & 0 & 1 & 1 & 1 \\ R5 & 0 & 0 & 0 & 1 & 1 & 0 \\ R6 & 1 & 0 & 0 & 1 & 0 & 0 \end{array}$$

Graphe (V,E) biparti

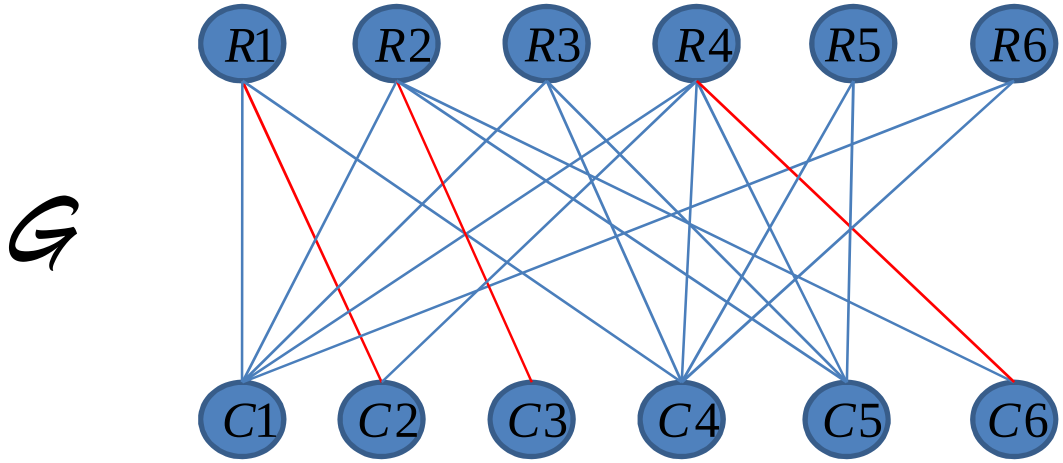
Il existe une partition de V en deux sous-ensembles R et C

telle que chaque arête ait une extrémité dans R et l'autre dans C.

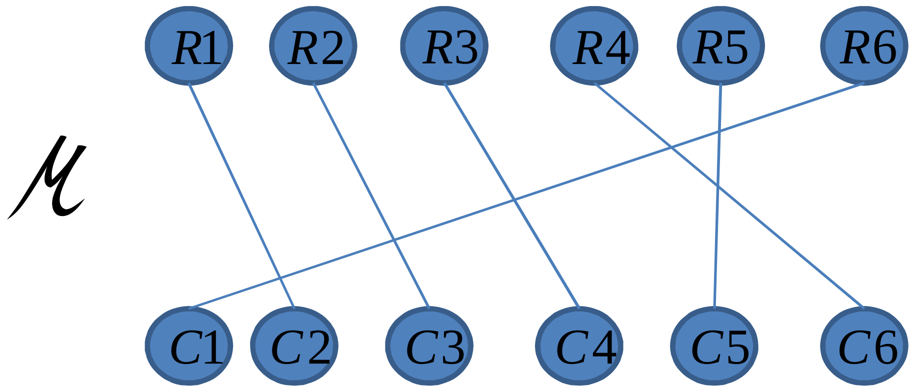
$\mathcal{G}$



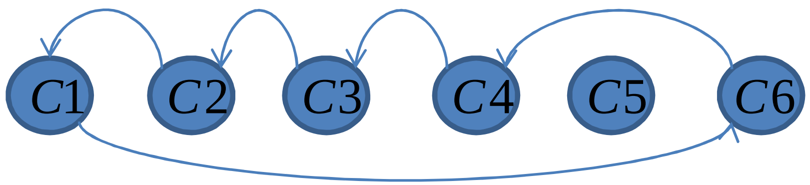
Principe de l'algorithme



Générer un couplage
à partir du graphe

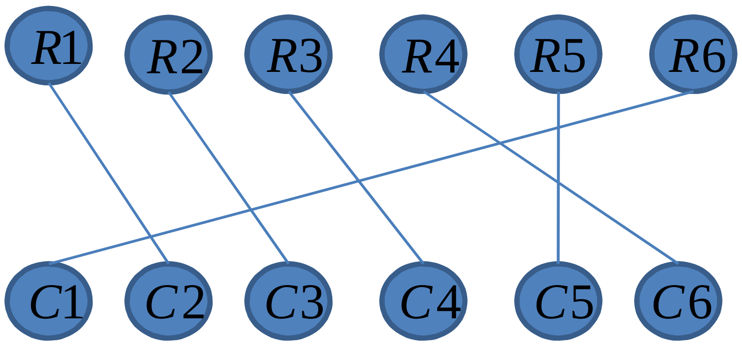


$$[(R_i, C_j) \in M] \Rightarrow [C_j \rightarrow i^{ème}\, position]$$



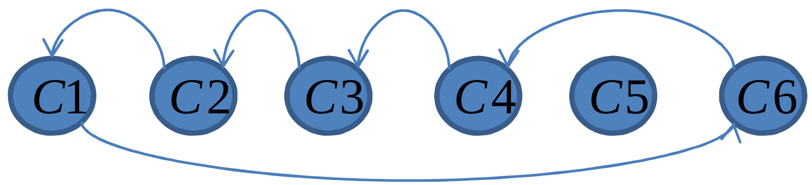
Principe de l'algorithme

$\mathcal{M}$



$$M = \begin{array}{c|cccccc} & C1 & C2 & C3 & C4 & C5 & C6 \\ \hline R1 & 0 & 1 & 0 & 0 & 0 & 0 \\ R2 & 0 & 0 & 1 & 0 & 0 & 0 \\ R3 & 0 & 0 & 0 & 1 & 0 & 0 \\ R4 & 0 & 0 & 0 & 0 & 0 & 1 \\ R5 & 0 & 0 & 0 & 0 & 1 & 0 \\ R6 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}$$

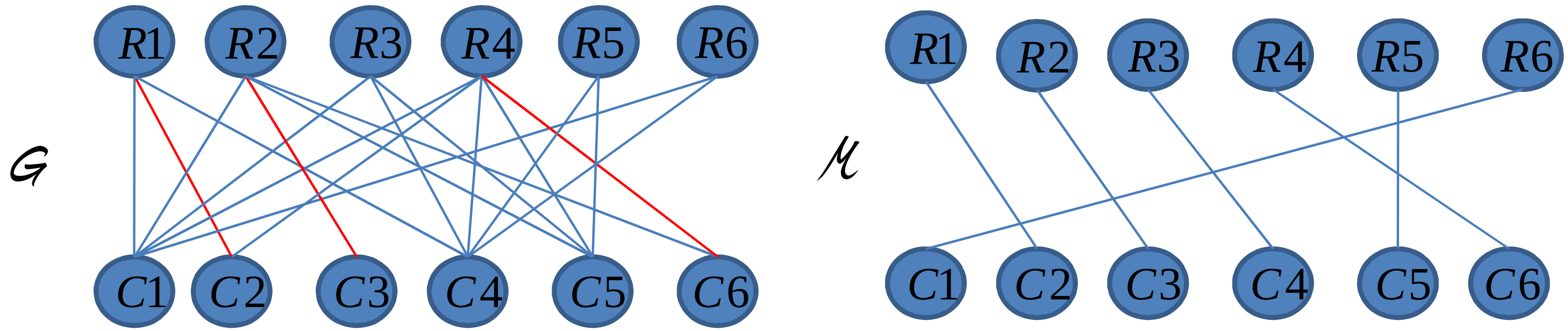
Matrice de permutation correspondant au couplage $\mathcal{M}$



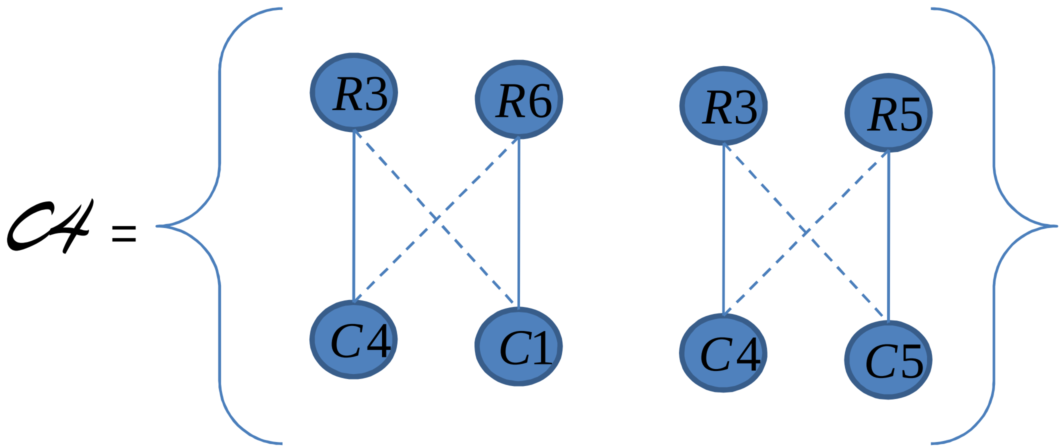
$$A \times^T M = \begin{array}{c|cccccc} & C1 & C2 & C3 & C4 & C5 & C6 \\ \hline R1 & 1 & 0 & 1 & 0 & 0 & 1 \\ R2 & 0 & 1 & 0 & 1 & 1 & 1 \\ R3 & 0 & 0 & 1 & 0 & 1 & 1 \\ R4 & 1 & 0 & 1 & 1 & 1 & 1 \\ R5 & 0 & 0 & 1 & 0 & 1 & 0 \\ R6 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}$$

$$S(AM) = 10$$

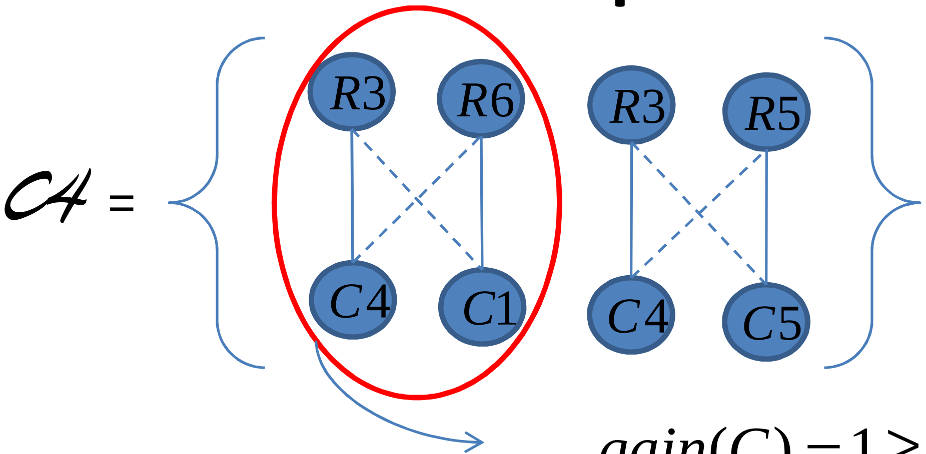
Principe de l'algorithme



Construction de ***C4***
ensemble des cycles
alternés

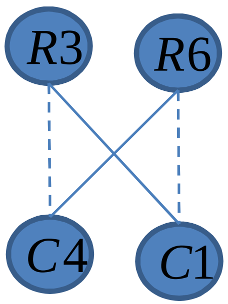


Principe de l'algorithme




```
while C4 ≠ ∅ do
    pick a cycle 𝒞 = (r1, c1, r2, c2) ∈ C4
    if isReversible(𝒞) and gain(𝒞) ≥ 0 then
        ℳ(1) = ℳ(1) ⊕ 𝒞
    remove the cycle 𝒞 from C4
```

$$gain(C) = 1 \geq 0$$



$$M^{(1)} = \begin{array}{c|cccccc} & C1 & C2 & C3 & C4 & C5 & C6 \\ \hline R1 & 0 & 1 & 0 & 0 & 0 & 0 \\ R2 & 0 & 0 & 1 & 0 & 0 & 0 \\ R3 & 1 & 0 & 0 & 0 & 0 & 0 \\ R4 & 0 & 0 & 0 & 0 & 0 & 1 \\ R5 & 0 & 0 & 0 & 0 & 1 & 0 \\ R6 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}$$

while $C4 \neq \emptyset$

Calcul du gain

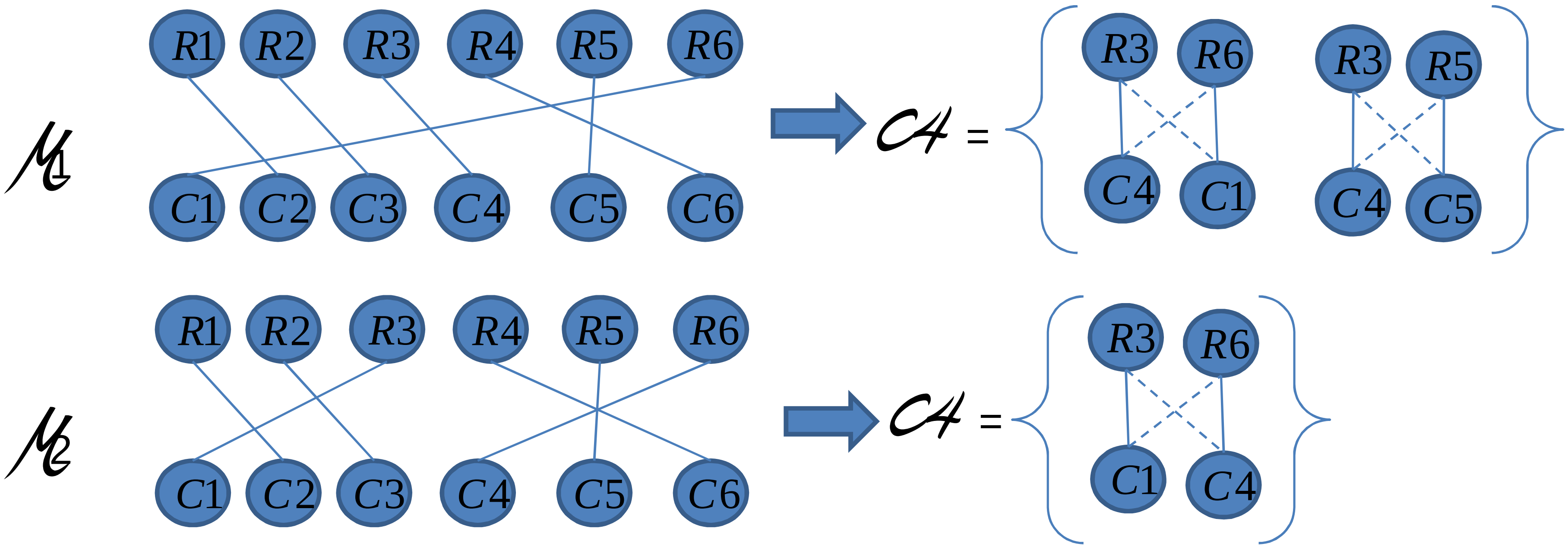
```
for each (ri, cj) ∈ ℳ do
    for each c ∈ N(ri) do
        mark(mate(c)) ← j
    for each r ∈ N(cj) do
        if mark(r) = j then
            score ← score + 1
```

PLAN

- **Contexte Générale et problématique**
- **Principe de l'algorithme**
- **Anomalies**
- **Améliorations**
- **Résultats et expériences**

Anomalies

L’ensemble C4 dépend du premier couplage



Chaque cycle n’est considéré qu’une seule fois ➔ problème d’ordre

PLAN

- **Contexte Générale et problématique**
- **Principe de l’algorithme**
- **Anomalies**
- **Améliorations**
- **Résultats et expériences**

Améliorations

- Trier C4 selon le gain des cycles.
- Parcours aléatoire des sommets et choisir le meilleur cycle contenant ce sommet.
- Borne supérieure et couplage initiale.

Améliorations

Exemple bornes supérieures UB1

A vertex v can be in at most d(v) − 1 alternating cycles of length four

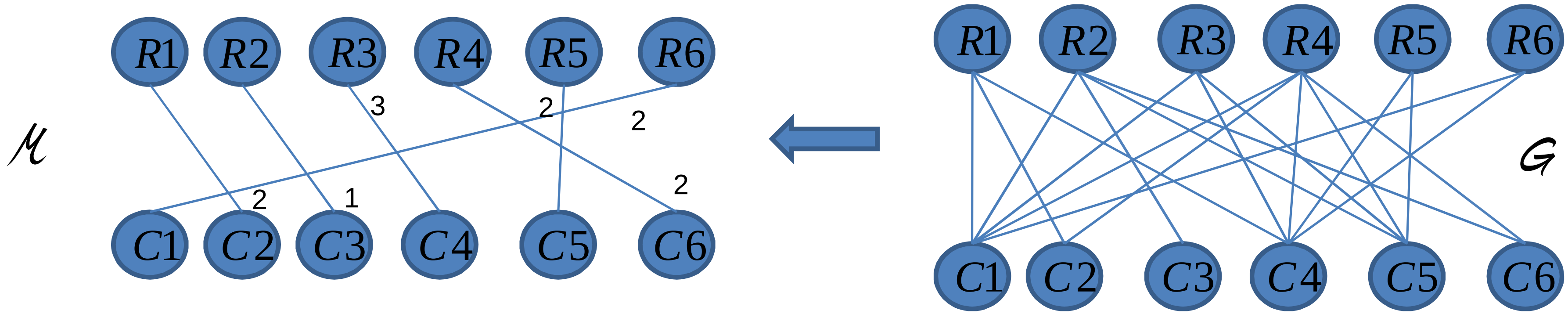
$$(R_i, C_j) \in M \Rightarrow w_{ij} = \min(d(R_i), d(C_j))$$

Améliorations

Exemple bornes supérieures UB1

A vertex v can be in at most d(v) − 1 alternating cycles of length four

$$(R_i, C_j) \in M \Rightarrow w_{ij} = \min(d(R_i), d(C_j))$$

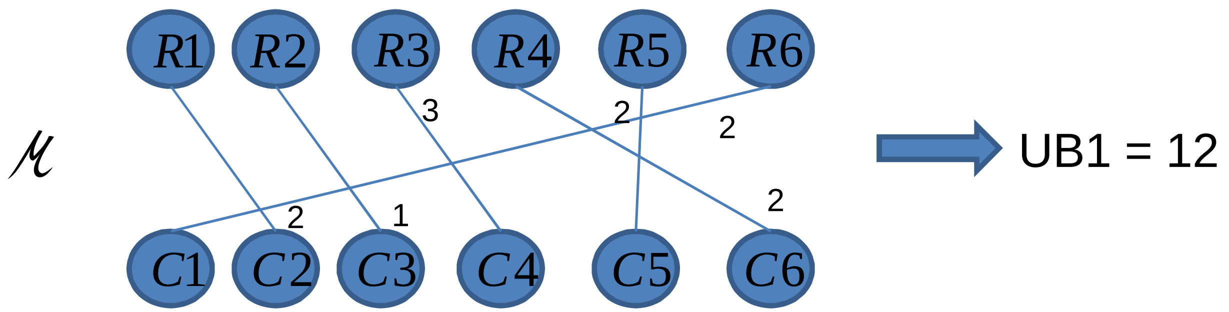


Améliorations

Exemple bornes supérieures UB1

A vertex v can be in at most d(v) − 1 alternating cycles of length four

$$(R_i, C_j) \in M \Rightarrow w_{ij} = \min(d(R_i), d(C_j))$$

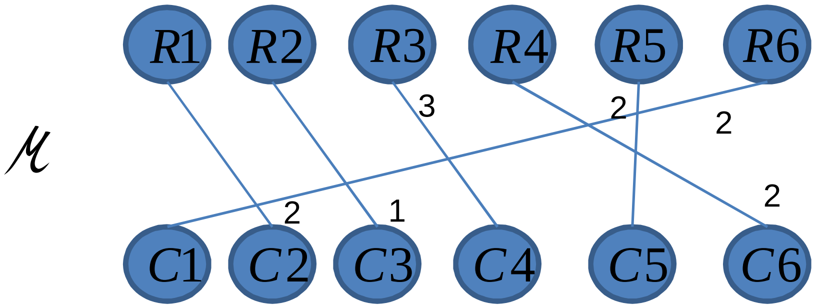


Améliorations

Exemple bornes supérieures UB1

A vertex v can be in at most d(v) − 1 alternating cycles of length four

$$(R_i, C_j) \in M \Rightarrow w_{ij} = \min(d(R_i), d(C_j))$$



UB1 = 12

$$S(AM) \leq 12$$

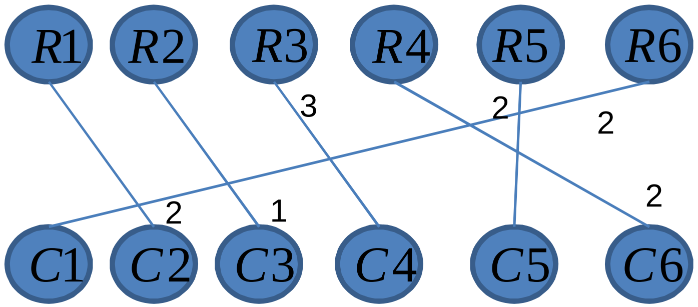
Améliorations

Exemple bornes supérieures UB1

A vertex v can be in at most d(v) − 1 alternating cycles of length four

$$(R_i, C_j) \in M \Rightarrow w_{ij} = \min(d(R_i), d(C_j))$$

$\mathcal{M}$



UB1 = 12

$$S(AM) \leq 12$$

On choisit le couplage initiale ayant le meilleur score

PLAN

- **Contexte Générale et problématique**
- **Principe de l’algorithme**
- **Anomalies**
- **Améliorations**
- **Résultats et expériences**

Résultats et expériences

Avec des matrices initialement symétriques + permutation aléatoire

$$A = P(B+I)Q$$

➔Le meilleur score de symétrie est connue

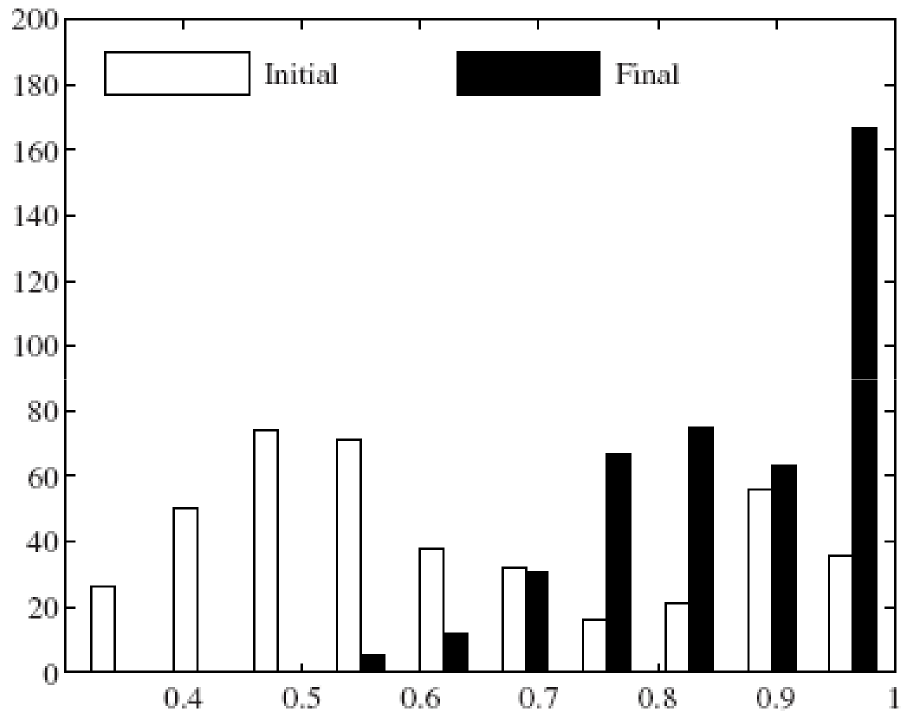
Avec des matrices non symétriques

➔Le meilleur score de symétrie n'est **pas** connue

	$S(A)$	$S(AM^{(0)})$	$S(AM^{(1)})$	$S(AM^{(3)})$	$S(AM^{(5)})$	$S(AM^{(7)})$
min	0.000	0.307	0.407	0.500	0.556	0.556
avg	0.005	0.628	0.754	0.834	0.872	0.891
max	0.125	1.000	1.000	1.000	1.000	1.000

Statistics regarding the symmetry scores normalized by the optimal score on 420 matrices. $S(A)$ is the symmetry score without any column permutation.

Résultats et expériences



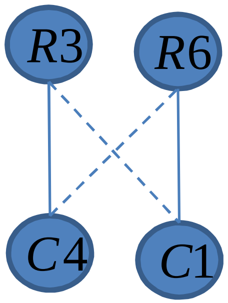
Histogram of the normalized symmetry scores for the initial matching $\mathcal{M}^{(0)}$ (white bars) and the final matching $\mathcal{M}^{(5)}$ (black bars)

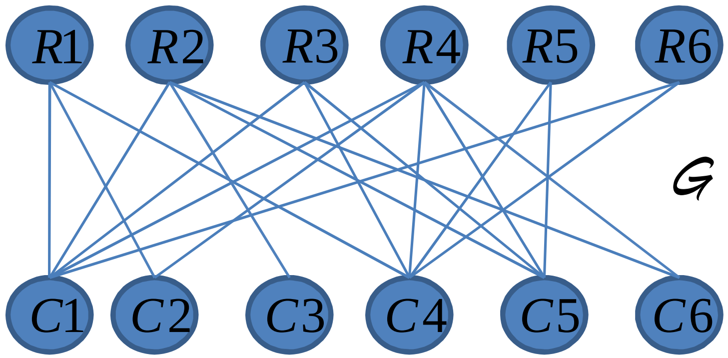
Références

- [5] Davis, T.: University of Florida sparse matrix collection. NA Digest 92/96/97 (1994/1996/1997), http://www.cise.ufl.edu/research/sparse/matrices
- [7] Duff, I.S., Koster, J.: On algorithms for permuting large entries to the diagonal of a sparse matrix. SIAM Journal on Matrix Analysis and Applications 22, 973–996 (2001)
- [8] Hendrickson, B., Kolda, T.G.: Partitioning rectangular and structurally unsymmetric sparse matrices for parallel processing. SIAM Journal on Scientific Compu
- [10] Hu, Y.F., Scott, J.A.: Ordering techniques for singly bordered block diagonal forms for unsymmetric parallel sparse direct solvers. Numerical Linear Algebra with Applications 12, 877–894 (2005)ting 21, 2048–2072 (2000)
- [12] Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. The Bell System Technical Journal 49, 291–307 (1970)
- [13] Lawler, E.: Combinatorial Optimization: Networks and Matroids. Dover, Mineola, New York (unabridged reprint of Combinatorial Optimization: Networks and Matroids, originally published by New York: Holt, Rinehart, and Wilson, c1976) (2001)
- [16] Reid, J.K., Scott, J.A.: Reducing the total bandwidth of a sparse unsymmetric matrix. SIAM Journal on Matrix Analysis and Applications 28, 805–821 (2006)
- [17] P. Hénon, P. Ramet, and J. Roman. PaStiX: A High-Performance Parallel Direct Solver for Sparse Symmetric Definite Systems. Parallel Computing, 28(2):301-321, January 2002. http://www.labri.fr/perso/ramet/
- [18] http://fr.wikipedia.org …

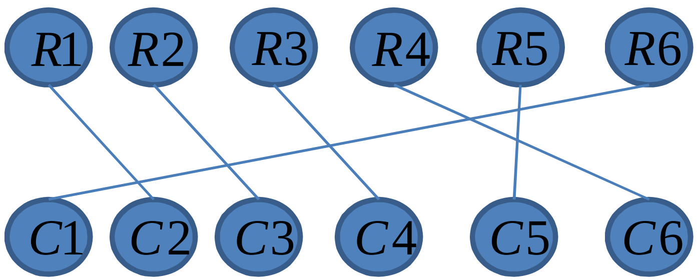
Questions ?

Annexe : Calcul du gain d'un cycle






```
for each (r_i, c_j) ∈ M do
    for each c ∈ N(r_i) do
        mark(mate(c)) ← j
    for each r ∈ N(c_j) do
        if mark(r) = j then
            score ← score + 1
```



$$gain(C) = 5 - 4 = 1 \geq 0$$