

Redistribution de données entre deux grappes d'ordinateurs

Travail tiré de

l'article *On Approximating a Scheduling Problem* ,
de P. Crescenzi, D. Xiaotie, C. Papadimitriou
dans la conférence Approx98.

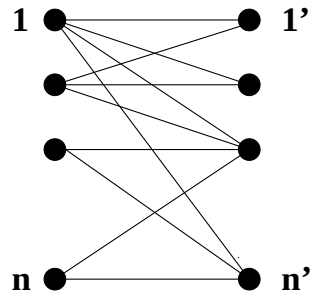
Plan

1. Présentation du problème
2. Complexité
3. Algorithmes d'approximations
4. Heuristiques

1. PRÉSENTATION DU PROBLÈME.

Le problème initial : configuration

Réseau :



Matrice du trafic Q :

	destination			
source	A	B	C	D
A'	3	18	15	3
B'	14	0	10	1
C'	4	0	0	0

Le problème initial : contraintes

- Modèle 1-port: un noeud participe à 1 communication.
- Prémption des communications.
- *Étape* : ensemble de communications fixées simultanées.

Définition du coût

$$\begin{aligned}
 \text{coût d'une étape} &= \overbrace{\max(\text{durée des comm.})}^{\text{durée effective}} + \underbrace{\text{coût d'initialisation d'une communication.}}_{\beta} \\
 \text{coût d'un schéma} &= \sum_{i=1..t} \text{coût de étape } i \\
 &= \underbrace{\sum_{i=1..t} \text{durée eff. de étape } i}_{\text{durée effective}} + t \times \beta
 \end{aligned}$$

Objectif : Minimiser le coût d'un schéma de communications.

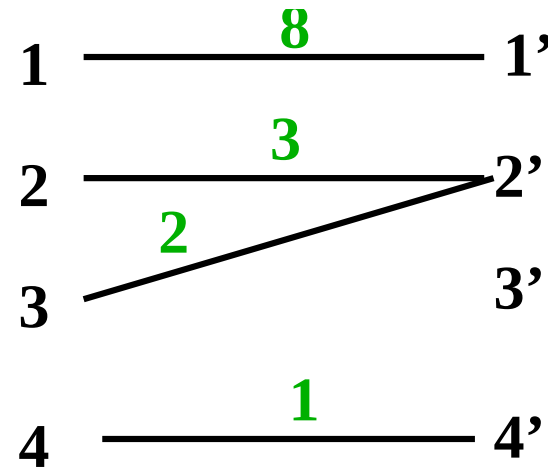
Simplification du problème

Le Problème.

- Données du problème: $\{source\}, \{destination\}$
 $\beta, Q = (q_{i,j} \in N)_{1 \leq i \leq n, 1 \leq j \leq n}$
- Minimiser le coût d'un schéma de communications (ensemble d'étapes).

Formulation en termes de couplages dans un graphe biparti

sources	destinations			
	1'	2'	3'	4'
1	8	0	0	0
2	0	3	0	0
3	0	2	0	0
4	0	0	0	1



Formulation en termes de graphe.

PRÉEMPTIVE BIPARTITE SCHEDULING **Problème:**

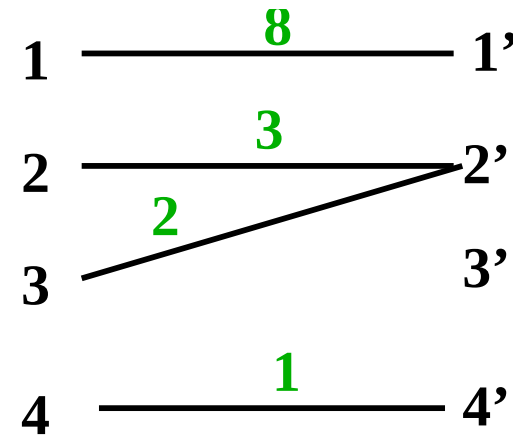
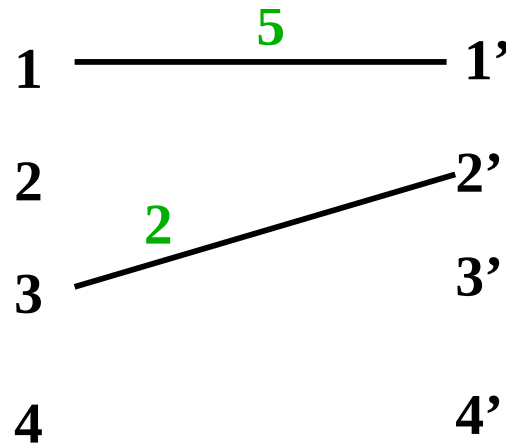
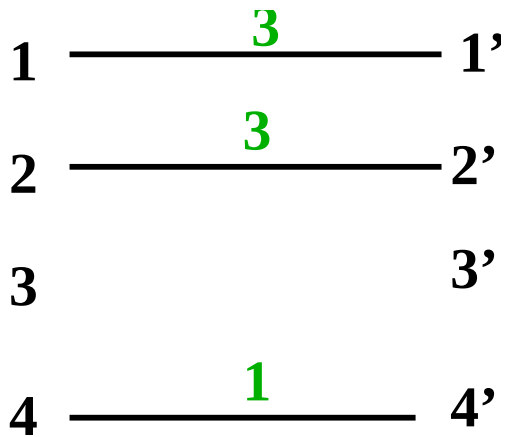
Entrée: Un graphe biparti $G = (X, Y, E)$, une fonction poids des aretes w .

Objectif : Trouver t couplages C_i pondérés valides de somme égale à G tel que

- $\sum_{i=1}^t w(C_i) + \beta \times t$ soit minimal.
- pour tout $e \in E$, $w(e) \leq \sum_{e \in C_i} |w(C_i)|$

Exemple

Configuration :



$$\text{coût} = 3 + 5 + 2 \times \beta$$

Simplification du problème: $\beta = 1$.

Entrée: Un graphe biparti $G = (X, Y, E)$, une fonction poids des aretes w .

Objectif : Trouver t couplages C_i pondérés valides de somme égale à G tel que

- $\sum_{i=1}^t w(C_i) + t$ soit minimal.
- pour tout $e \in E$, $w(e) \leq \sum_{e \in C_i} |w(C_i)|$

Notations

$m(G)$ = le nombre d'arêtes de G

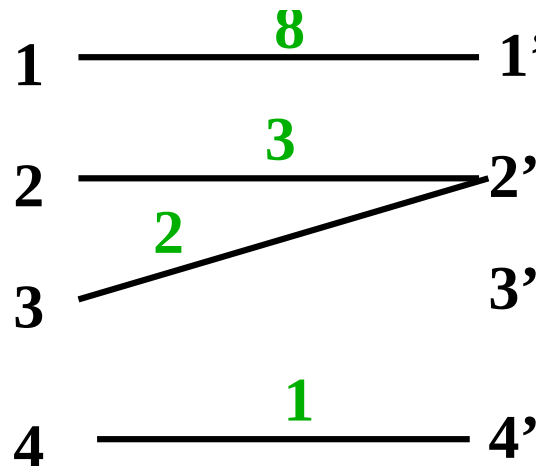
$\Delta(G)$ = le maximum des degrés du graphe G

$W(G)$ = le maximum de la somme des poids incidents à un sommet

$P(G)$ = la somme des poids de toutes les arêtes

Exemple :

- $m(G) = 4$
- $\Delta(G) = 2$
- $W(G) = 8$
- $P(G) = 14$



2. COMPLEXITÉ DU PROBLÈME

NP-complétude du problème BPS

Théorème 1: Le problème *BPS*

Instance: Un entier n , un graphe biparti G pondéré, un entier B .

Question: Trouver une suite de couplages pondérés valides $C_1, C_2, \dots, C_t$ tel que $G = \sum_{i=1}^t C_i$ et $\sum_{i=1}^t w(C_i) + t \leq B$

est NP-complet.

Définition le problème Restricted TimeTable Design

Problème Restricted TimeTable Design

Instance: Un ensemble $T = \{t_1, \dots, t_n\}$ d'enseignant, Un ensemble $C = \{c_1, \dots, c_m\}$ de classes, une fonction de disponibilité $A : T \times \{1, 2, 3\} \rightarrow \{0, 1\}$, et une fonction de demande $R : T \times C \rightarrow \{0, 1\}$

Question: Existe-t'il une fonction d'affectation $f : T \times C \times \{1, 2, 3\} \rightarrow \{0, 1\}$ tel que

1. $f(t_i, c_j, k) = 1$ implique $A(t_i, k) = 1$
2. $\forall c_j \in C, \forall k \in \{1, 2, 3\}, \sum_{i=1}^n f(t_i, c_j, k) \leq 1.$
3. $\forall t_i \in T, \forall k \in \{1, 2, 3\}, \sum_{j=1}^m f(t_i, c_j, k) \leq 1.$
4. $\forall c_j \in C, \forall t_i \in T, \sum_{k=1}^3 f(t_i, c_j, k) = R(t_i, c_j).$

Exemple.

$$A =$$

0	1	1
1	0	0
1	1	0

$$\text{et } R =$$

1	0	1	0
0	1	0	0
0	0	1	1

Une fonction d'affectation:

$$f(t_1, c_1, 2) = f(t_1, c_3, 3) = f(t_2, c_2, 1) = f(t_3, c_3, 1) = f(t_3, c_4, 2) = 1$$

1. $A(t_1, 2) = A(t_1, 3) = A(t_2, 1) = A(t_3, 1) = A(t_3, 2) = 1$
2. $\forall c_j \in C, \forall k \in \{1, 2, 3\}, \sum_{i=1}^n f(t_i, c_j, k) \leq 1$: Okay.
3. $\forall t_i \in T, \forall k \in \{1, 2, 3\}, \sum_{j=1}^m f(t_i, c_j, k) \leq 1$: Okay.
4. $\forall c_j \in C, \forall t_i \in T, \sum_{k=1}^3 f(t_i, c_j, k) = R(t_i, c_j)$: Okay.

Transformation de Restricted TimeTable Design. vers BPS.

- Soit une instance $\langle T, C, A, R \rangle$ de Restricted TimeTable Design.
- Voici l'instance correspondante $\langle G = (V_1, V_2, E), w \rangle$ de BPS.
 1. ϵ un rationnel positif et arbitraire.
 2. $|V_1| = 2|T| = 2n$, $|V_2| = |T| + |C| = n + m$
 3. $\forall i, j$, tel que $1 \leq i \leq n$ et $1 \leq j \leq m$, $(v_1^i, v_2^j) \in E \Leftrightarrow R(t_i, c_j) = 1$.
Et, dans ce cas $w(v_1^i, v_2^j) = \epsilon$
 4. $\forall i, j$, tel que $n + 1 \leq i \leq 2n$ et $1 \leq j \leq m$, $(v_1^i, v_2^j) \notin E$
 5. $\forall i, j$, tel que $1 \leq i \leq n$ et $m + 1 \leq j \leq m + n$, $\forall k \in \{1, 2, 3\}$,
 $(v_1^i, v_2^j) \in E \Leftrightarrow j = m + [(i + k - 2) \bmod n + 1] \wedge A(t_i, k) = 0$.
Et, dans ce cas $w(v_1^i, v_2^j) = k - 1 + \epsilon$
 6. $\forall i, j$, tel que $n + 1 \leq i \leq 2n$ et $m + 1 \leq j \leq m + n$, $\forall k \in \{1, 2, 3\}$,
 $(v_1^i, v_2^j) \in E \Leftrightarrow j = m + [(i - n + k - 2) \bmod n + 1] \wedge A(t_i, k) = 1$.
Et, dans ce cas $w(v_1^i, v_2^j) = k - 1 + \epsilon$

Exemple: $|T| = 3$ et $|C| = 4$

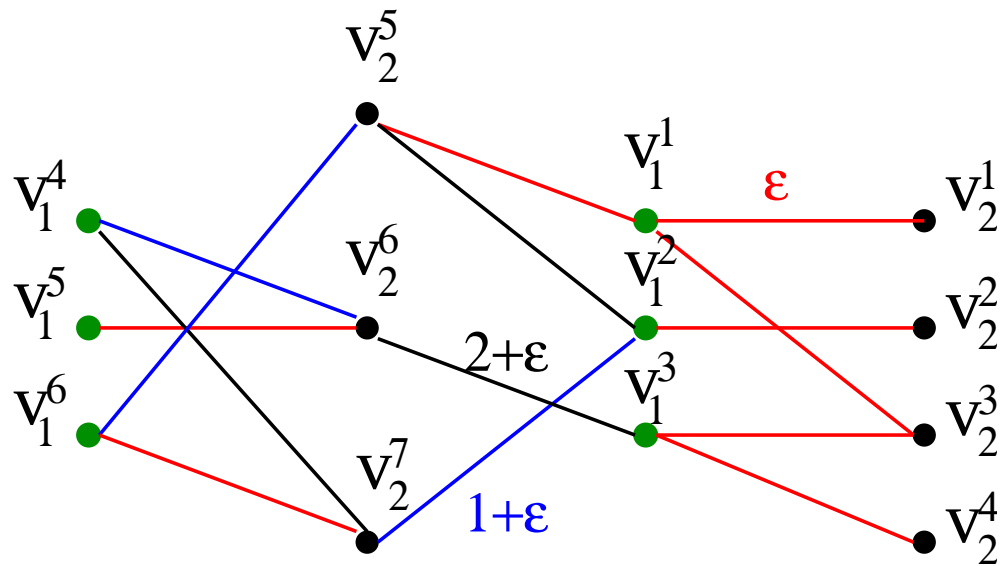
$$A = \begin{array}{|c|c|c|} \hline 0 & 1 & 1 \\ \hline 1 & 0 & 0 \\ \hline 1 & 1 & 0 \\ \hline \end{array}$$

et

R

$$= \begin{array}{|c|c|c|c|} \hline 1 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 \\ \hline \end{array}$$

L'instance correspondante de PBS :



Remarque

Lemme 2: Si $\langle T, C, A, R \rangle$ admet une réponse positive alors
 $opt(\langle G = (V_1, V_2, E), w \rangle) = 6 + 3\epsilon$

Proof: Soit f une fonction d'affectation pour $\langle T, C, A, R \rangle$,

Pour $h = 1, 2, 3$,

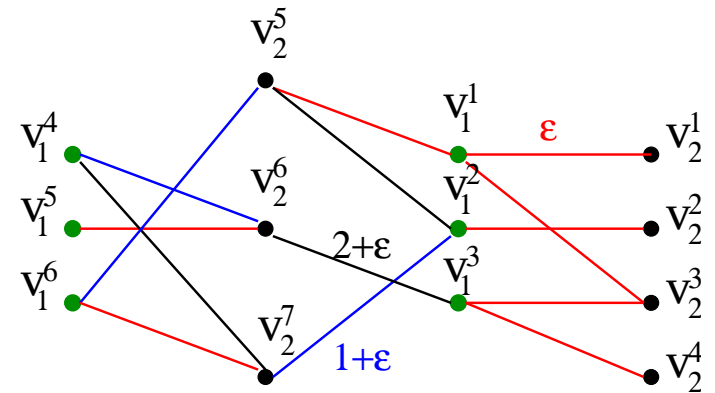
$$\begin{aligned} M_k &= \{(v_1^i, v_2^j) : 1 \leq i \leq n, 1 \leq j \leq m, f(t_i, c_j, k) = 1\} \\ &\cup \{(v_1^i, v_2^j) : 1 \leq i \leq 2n, m+1 \leq j \leq m+1, w(v_1^i, v_2^j) = k-1+\epsilon\} \end{aligned}$$

est une solution de PBS dont son coût est $6 + 3\epsilon$.

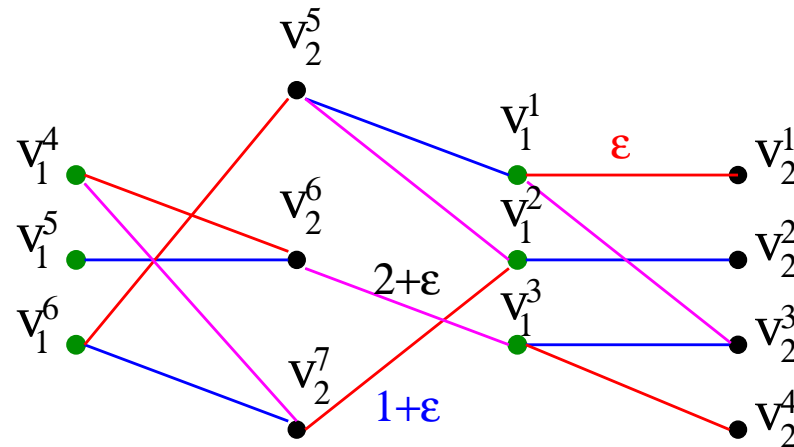
□

Retour à l'exemple: $|T| = 3$ et $|C| = 4$

$$A = \begin{array}{|c|c|c|} \hline 0 & 1 & 1 \\ \hline 1 & 0 & 0 \\ \hline 1 & 1 & 0 \\ \hline \end{array}$$

$$R = \begin{array}{|c|c|c|c|} \hline 1 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 \\ \hline \end{array}$$


$$\begin{aligned} f(t_1, c_1, 2) &= 1 \\ f(t_1, c_3, 3) &= 1 \\ f(t_2, c_2, 1) &= 1 \\ f(t_3, c_3, 1) &= 1 \\ f(t_3, c_4, 2) &= 1 \end{aligned}$$



— couplage M_1
 — couplage M_2
 — couplage M_3

Remarque

Lemme 3: Si $\langle T, C, A, R \rangle$ admet une réponse négative alors $opt(\langle G = (V_1, V_2, E), w \rangle) \geq 7 + 3\epsilon$

Proof: Supposons, que $\langle T, C, A, R \rangle$ admet une réponse négative et qu'il existe une collection $M_1, \dots, M_s$ tel que leur coût $\leq 7 + 3\epsilon$

- Degrée minimum de G est 3 $\Rightarrow s \geq 3$.
- somme des couplages $\leq 4 + 3\epsilon$.
- toutes les arêtes de poids ϵ (resp. $1 + \epsilon$, $2 + \epsilon$) sont dans le même couplage.
- Donc $\langle T, C, A, R \rangle$ admet une réponse Positive (d'où contradiction). $\square$

Inapproximabilité.

Théorème 4: Approximer PBS avec un rapport inférieure à $7/6$ est NP-dur

Preuve par contradiction. Supposons qu'il existe un **POLYNOMIAL** algo. $\mathcal{A}$ avec un rapport d'approx. ρ avec $\rho \leq 7/6$.

1. Construire $\langle G, w \rangle$ à partir de $\langle T, C, A, R \rangle$ et de $\mathcal{R}$

Un algo. $\mathcal{B}$: 2. Appliquer $\mathcal{A}$ sur l'instance $\langle G, w \rangle$

3. Si le coût $cost$ de la solution $\leq 7 + 3\epsilon$, retourner **VRAI** sinon **FAUX**.

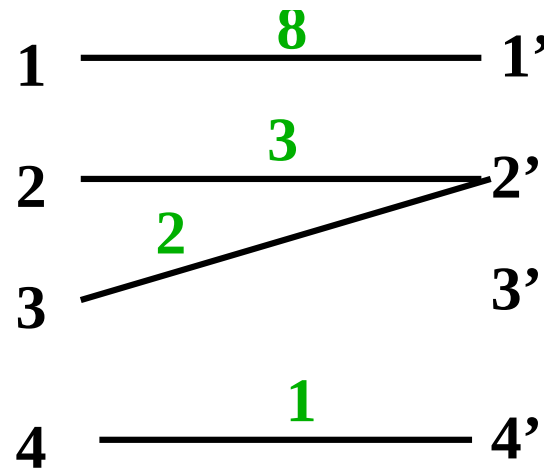
$\mathcal{B}$ résout le problème RTD en temps polynomial: car

- Si $\langle T, C, A, R \rangle$ admet une réponse positive, alors $opt(G, w) = (6 + 3\epsilon)$
- $6 + 3\epsilon \leq cost \leq \rho(6 + 3\epsilon)$ avec $\rho \leq 7/6 \Rightarrow 6 + 3\epsilon \leq cost < 7 + 3\epsilon$

3. ALGORITHMES D'APPROXIMATIONS.

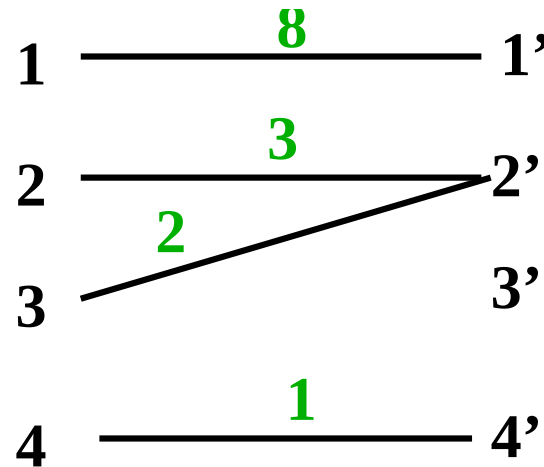
Borne inférieure de la solution

- Borne inférieure de la durée effective: $W(G)$
- Rappel: $W(G)$ le maximum de la somme des poids incidents à un sommet.



Borne inférieure du coût

- Borne inférieure du nombre d'étapes: $\Delta(G)$
- Rappel: $\Delta(G)$ = le maximum des degrés du graphe G



Borne inférieure de la solution

Théorème 2: Le coût de la solution optimale est minimisé par

$$\eta(G) = W(G) + \Delta(G)$$

Car le coût = durée effective + nbre_étapes $\times 1$

Algorithme

Algorithme 1 :

- **Entrée:** un graphe biparti G .
 - **Sortie:** une décomposition en couplages valides.
1. Décomposer les arêtes e en $\lceil w(e) \rceil$ arêtes de poids 1.
ce qui donne un multigraphe H
 2. Décomposer $\Delta(H)$ couplages dans H .
 3. Les couplages précédents forment une solution pour G .

1 couplage = 1 étape de durée 1

Complexité :dépend du poids des arêtes:

Algorithme pseudo-polynomial.

Évaluation de la qualité de la solution

- Nombre d'étapes = nombre de couplages = $\Delta(H)$.
- Durée effective de chaque étape = 1
- $Cost(A1) = 2\Delta(H)$
- Chaque arête de G représente $\lceil w_G(e) \rceil$ dans H .

$$\Rightarrow \Delta(H) \leq \Delta(G) + W(G)$$

$$\Delta(G) + W(G) \leq 2\Delta(H) \leq 2(\Delta(G) + W(G))$$

Algorithme Polynomiale

Théorème 5: Il existe un algorithme pour le problème PBS avec un rapport d'approximation 2.

Preuve: Algorithme:

- Soit une nouvelle fonction poids w' tel que $\forall e \in E, w'(e) = \lceil w_G(e) \rceil$.
- $W(G, w') = \max\{w'(v) : v \in G\} \Rightarrow \Delta(H) = W(G, w')$
- Transformer G en un graphe poids-régulier (G', w') : $\forall v \in V(G'), w'(v) = W(G, w')$.
- Tant qu'il existe une arête dans G' ,
 - Trouver un couplage parfait M dans G' .
 - Soit $w'(M) = \min\{w'(e) : e \in M\}$.
 - Pour chaque arête $e \in M$, $w'(e) = w'(e) - w'(M) + \text{suppression}$.

Heuristique de Gopal et Wong

- Trouver une solution avec le nombre minimum de couplage.
- Solution: 1 couplage $M = 1$ étape de durée $\max\{w(e) : e \in M\}$

A=

M						
	1	M				
		1	1	M		
			1	1	1	M
				1	1	1
					1	1
						1

Évaluation de l'heuristique de Gopal et Wong

Proposition 6: Cette heuristique a un rapport d'approximation non borné.

Proof:

- Soit une $(2n + 1) \times (2n + 1)$ matrice A tel que
 - $a_{i,(2i-1)} = M$ et $a_{(i+j),(2i-1+j)} = 1$ avec $i = 1, \dots, n + 1$, $j = 1, \dots, 2(n - i) + 2$. sinon $a_{i,j} = 0$.
- Solution de cet algorithme
 - $M_i = \{a_{(i+j),(2i-1+j)} : j = 0, \dots, 2(n - i) + 2\}$ pour $i = 1, \dots, n + 1$
 - n couplages pour les autres arêtes.
- Coût $:= \sum_{i=1}^{n+1} \text{cout}(M_i) + n = M \times (n + 1) + n$.
- rapport $:= \frac{(n+1)+n}{M+n} = \infty +$

□