Redistribution de données à travers un réseau à haut débit

Wagner Frédéric

MOAIS
ID - IMAG

28 septembre 2006

Redistribution

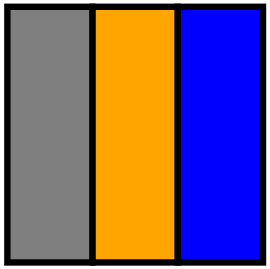
- opération de communication parallèle
- repositionnement des données au sein d'un calculateur
- utilisée entre 2 calculateurs dans le cadre de couplages de code
- transfert et réorganisation de la distribution de données

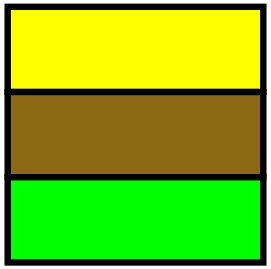
Redistribution

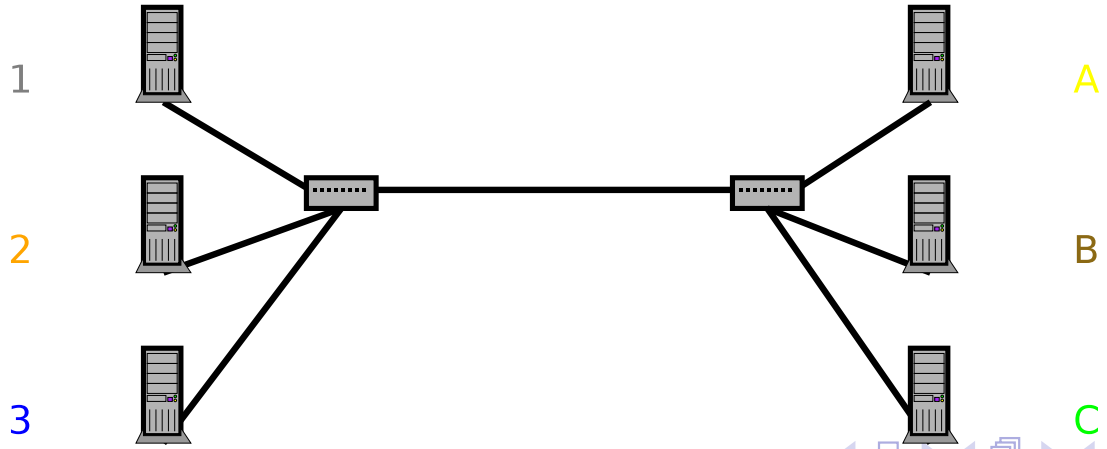
- opération de communication parallèle
- repositionnement des données au sein d'un calculateur
- utilisée entre 2 calculateurs dans le cadre de couplages de code
- transfert et réorganisation de la distribution de données

Redistribution

- opération de communication parallèle
- repositionnement des données au sein d'un calculateur
- utilisée entre 2 calculateurs dans le cadre de couplages de code
- transfert et réorganisation de la distribution de données

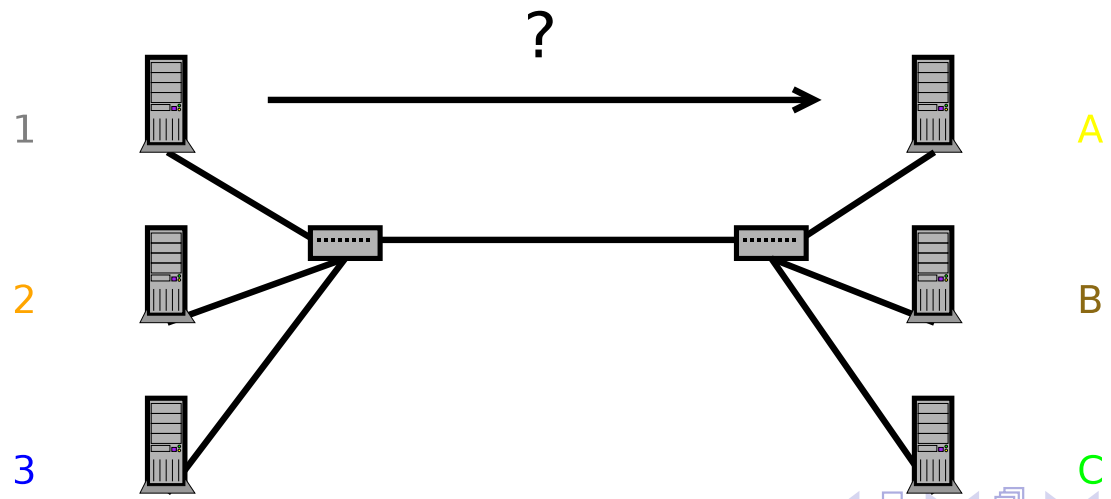






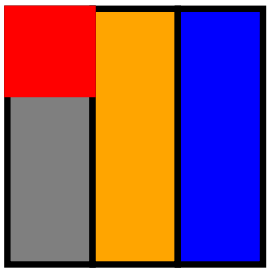
Redistribution

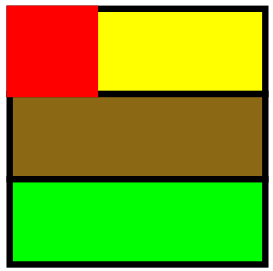
- opération de communication parallèle
- repositionnement des données au sein d'un calculateur
- utilisée entre 2 calculateurs dans le cadre de couplages de code
- transfert et réorganisation de la distribution de données

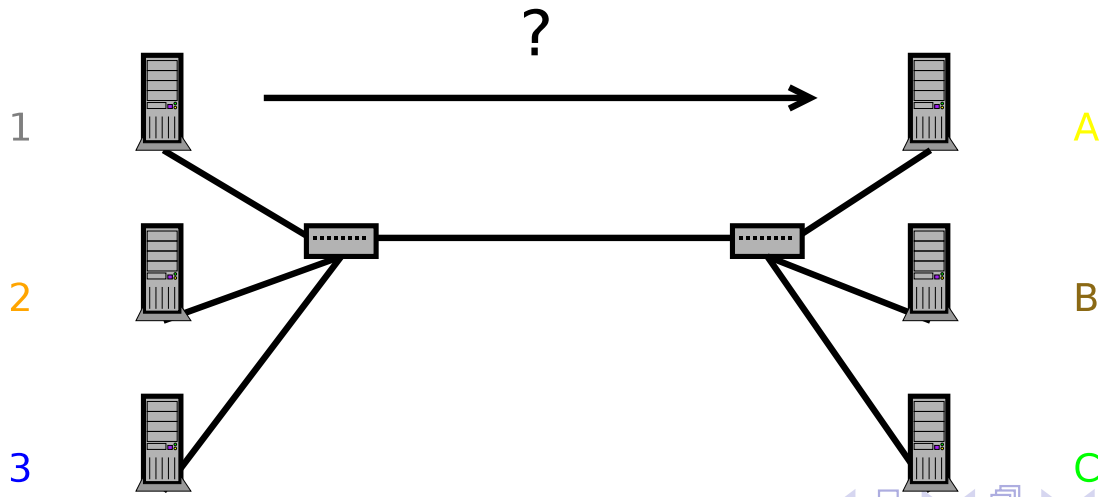


Redistribution

- opération de communication parallèle
- repositionnement des données au sein d'un calculateur
- utilisée entre 2 calculateurs dans le cadre de couplages de code
- transfert et réorganisation de la distribution de données

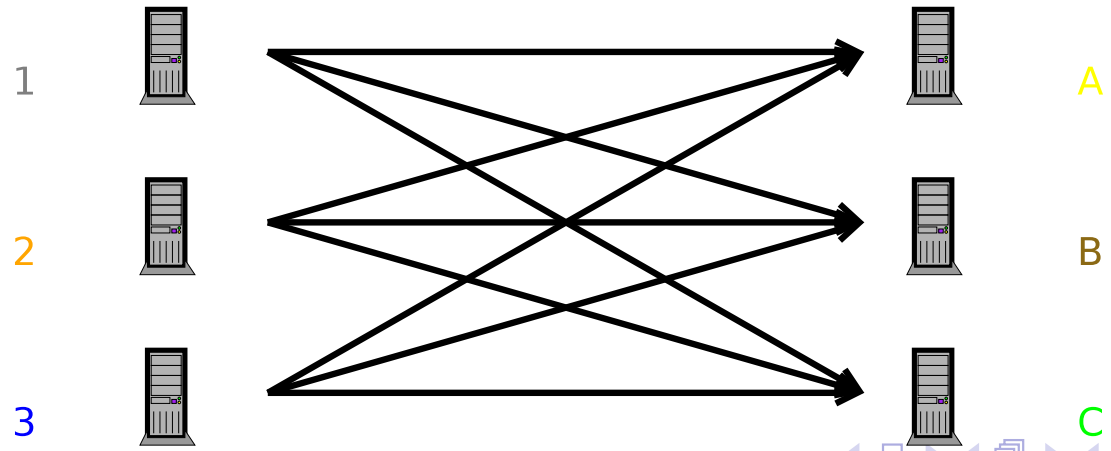






Redistribution

- opération de communication parallèle
- repositionnement des données au sein d'un calculateur
- utilisée entre 2 calculateurs dans le cadre de couplages de code
- transfert et réorganisation de la distribution de données



Redistribution

État de l'art

- de nombreux travaux sur des redistributions locales
  - placement des données [Hsu01]
  - ordonnancement des communications
    - algorithme de la chenille [Prylli96]
    - redistributions blocs-cyclique [Desprez97]
    - ordonnancement, graphes bipartis [Crescenzi01]

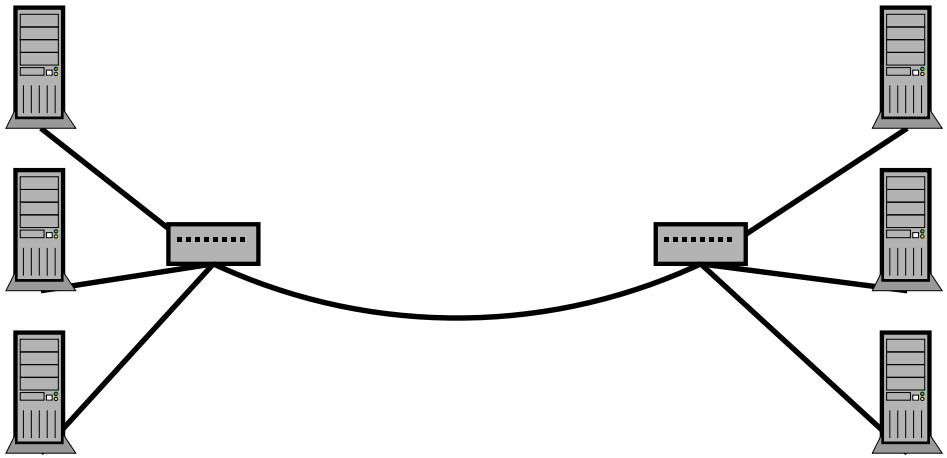
Plan de l'exposé

1. Redistributions
2. Le problème KPBS
   - Présentation
   - Modélisation
   - Bornes inférieures
3. Algorithme pseudo-polynomial
4. Algorithme polynomial (GGP)
5. Conclusion

Plan de l'exposé

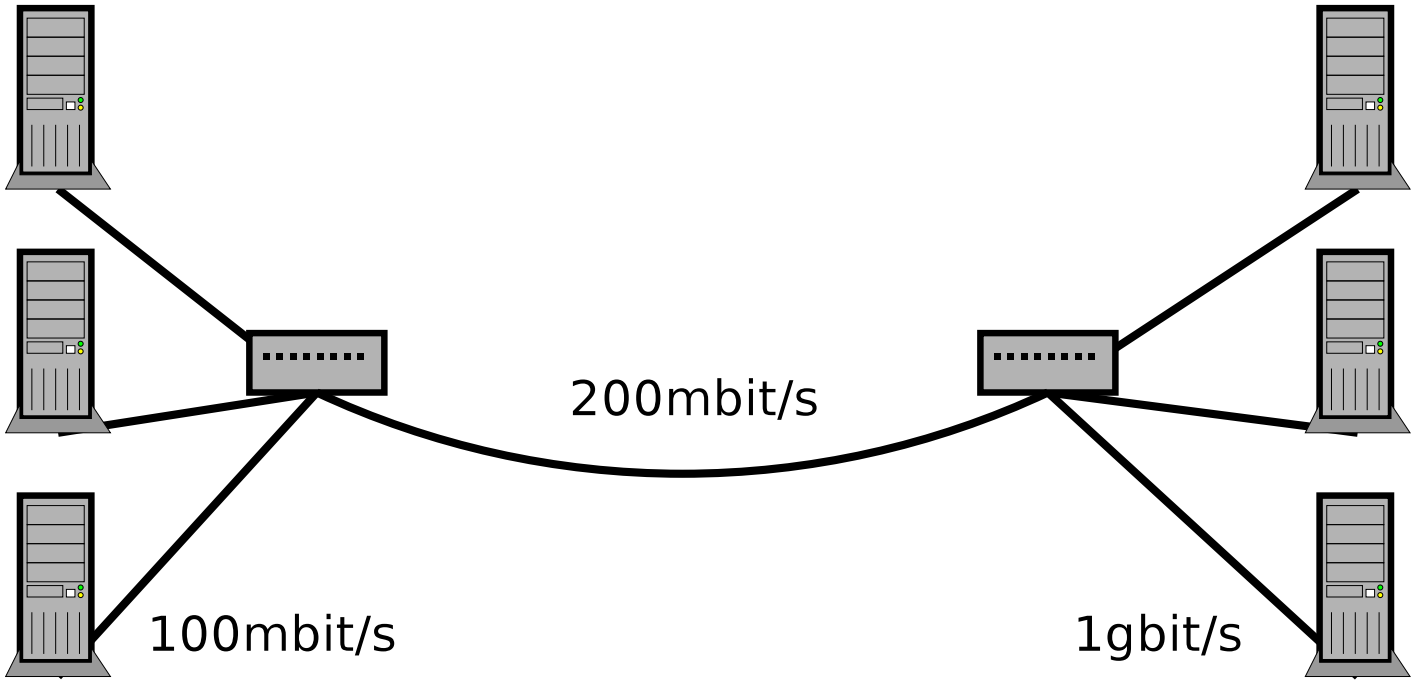
1. Redistributions
2. Le problème KPBS
   - Présentation
   - Modélisation
   - Bornes inférieures
3. Algorithme pseudo-polynomial
4. Algorithme polynomial (GGP)
5. Conclusion

Contexte

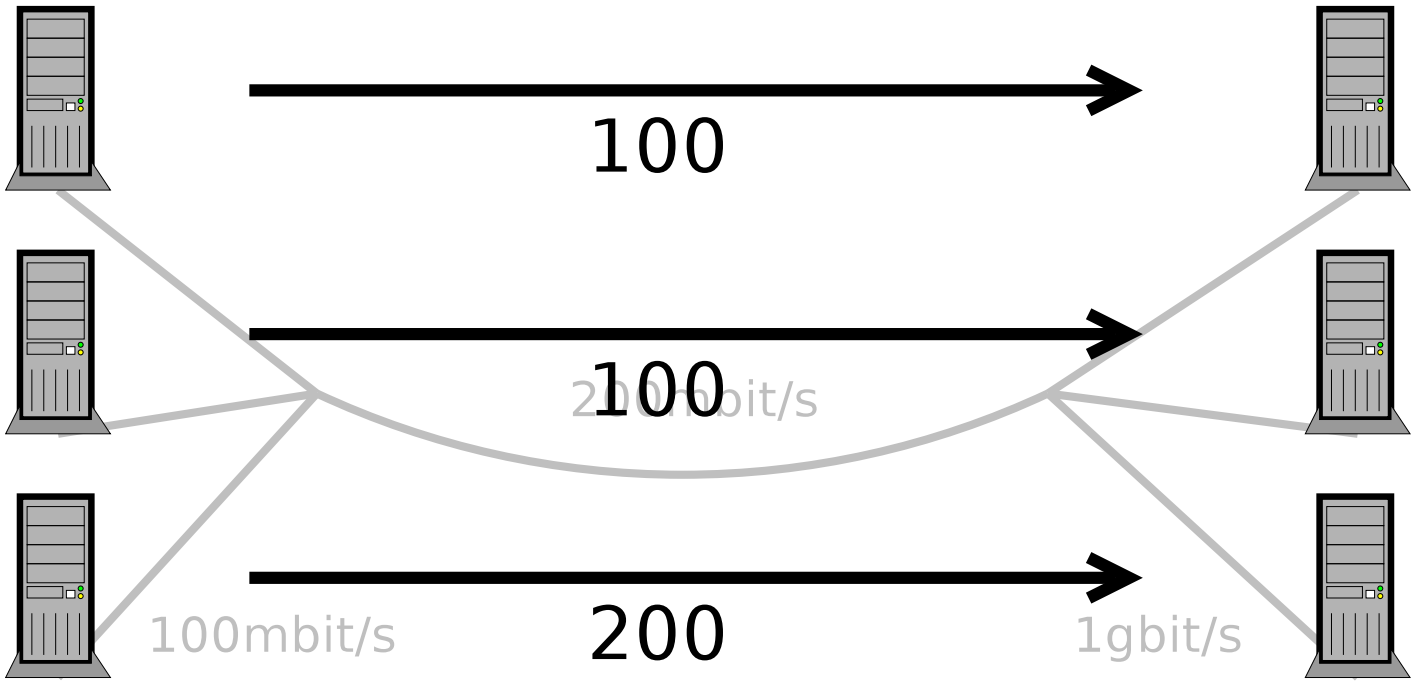


- redistribution de données entre deux grappes homogènes
- bandes passantes connues
- latence connue

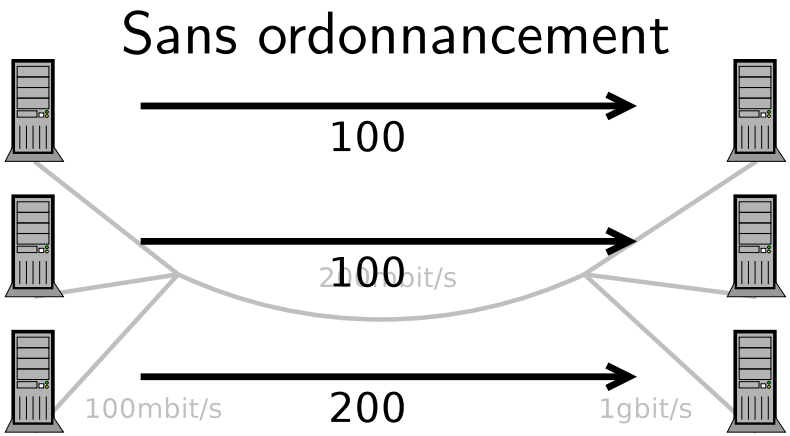
Exemple

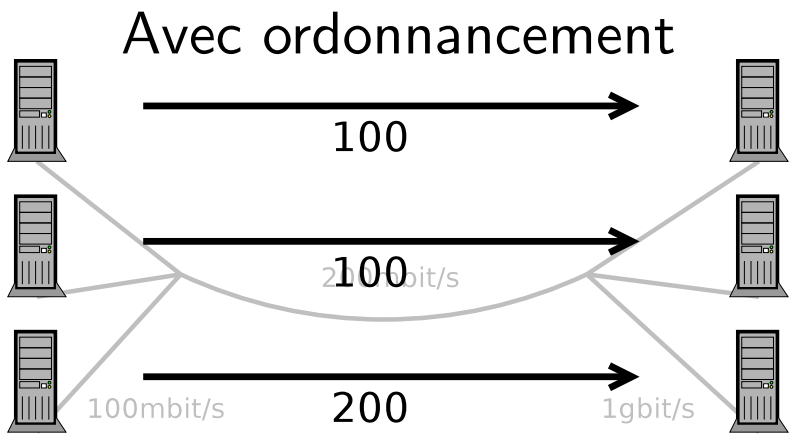


Exemple

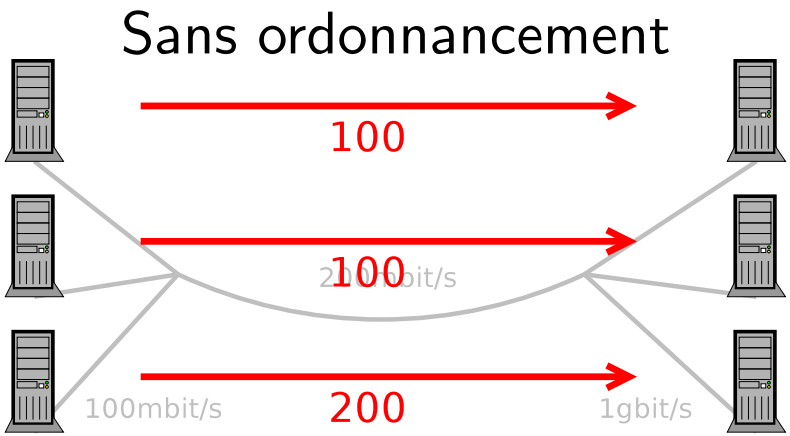


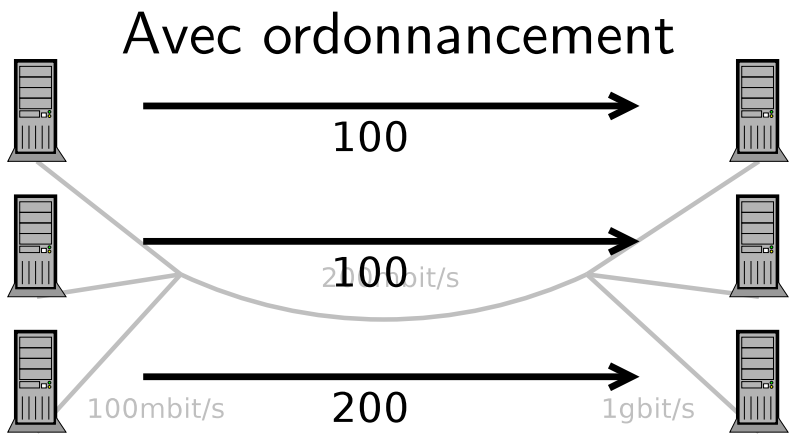
Exemple


Sans ordonnancement
100
100
100mbit/s
200
1gbit/s


Avec ordonnancement
100
100
100mbit/s
200
1gbit/s

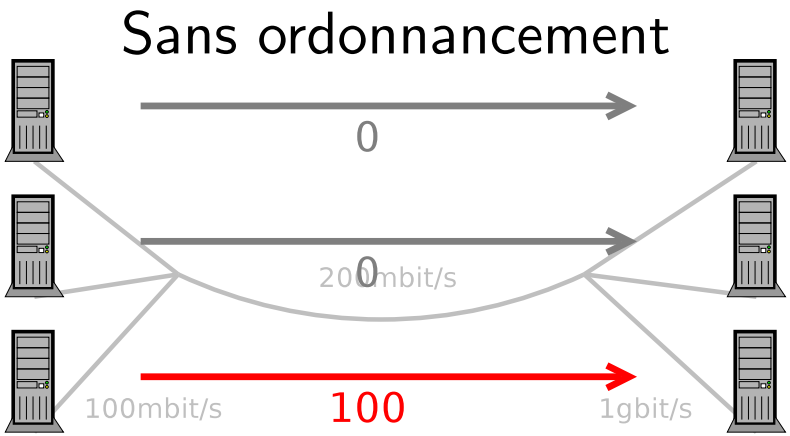
Exemple

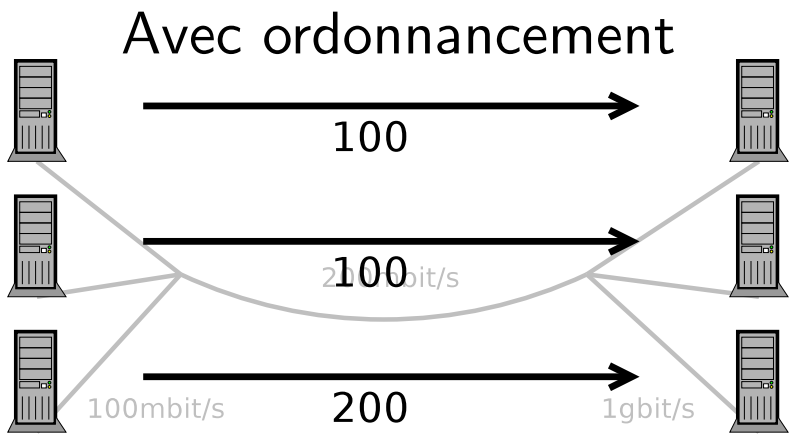




Temps : 1,5 seconde

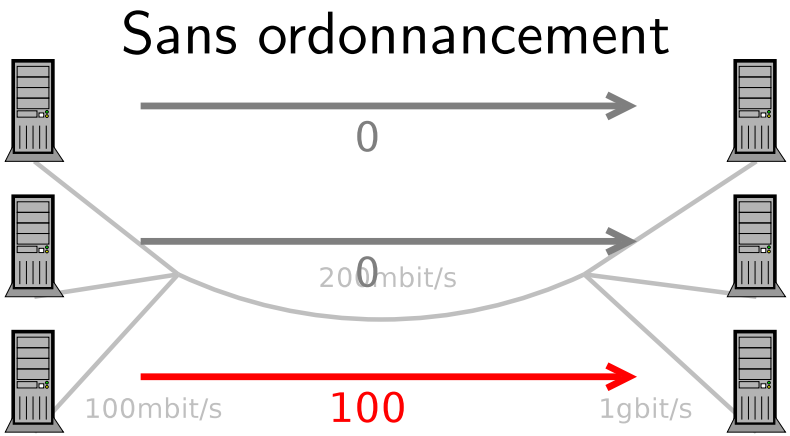
Exemple

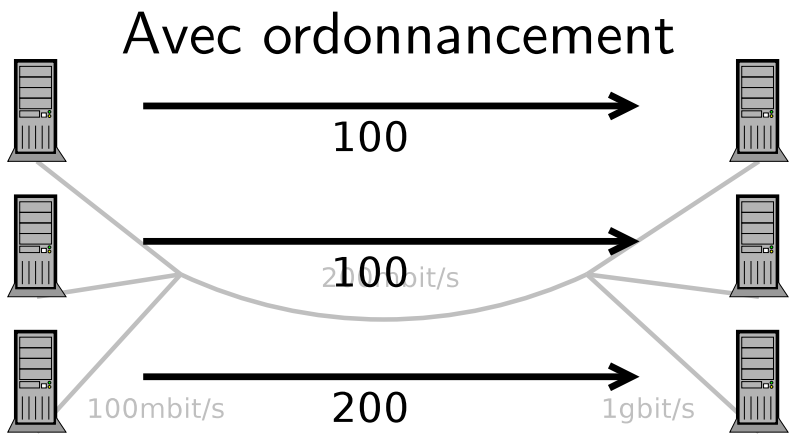




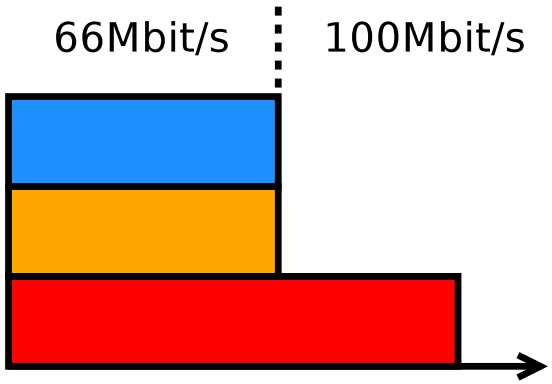
Temps : 1,5 + 1 secondes

Exemple

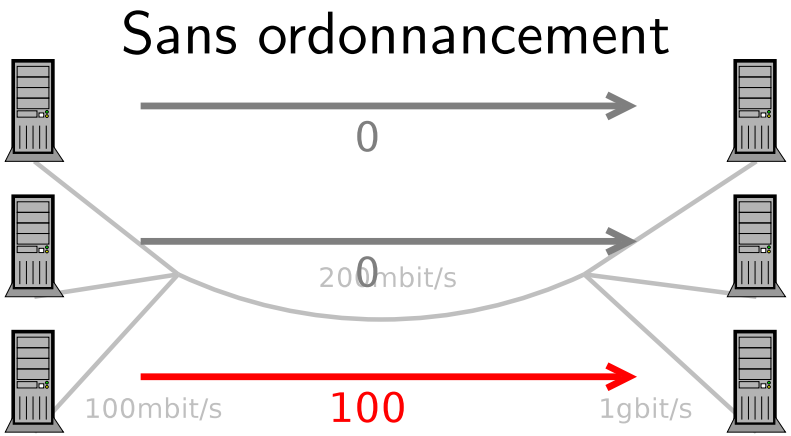


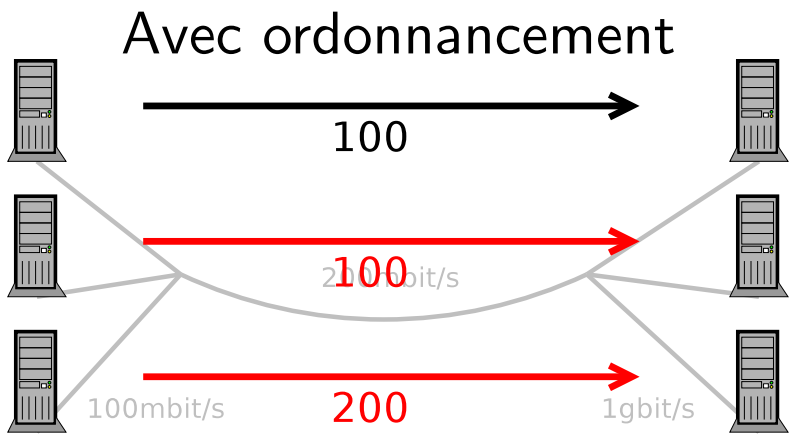


Temps : 2,5 secondes



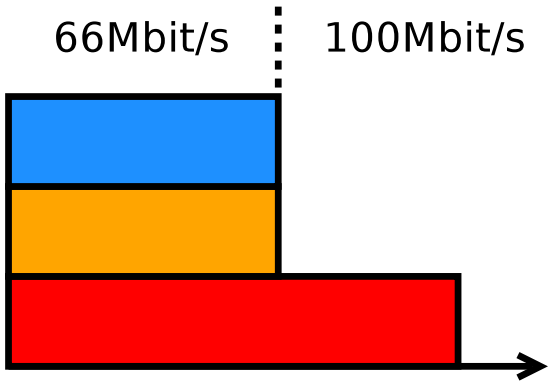
Exemple



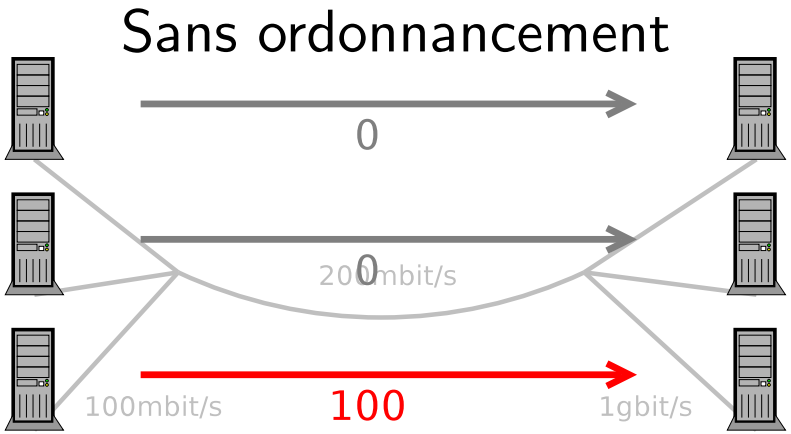


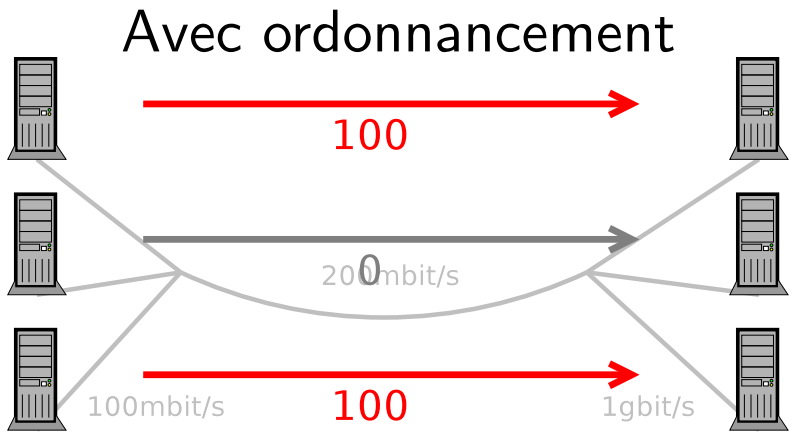
Temps : 2,5 secondes

Temps : 1 seconde



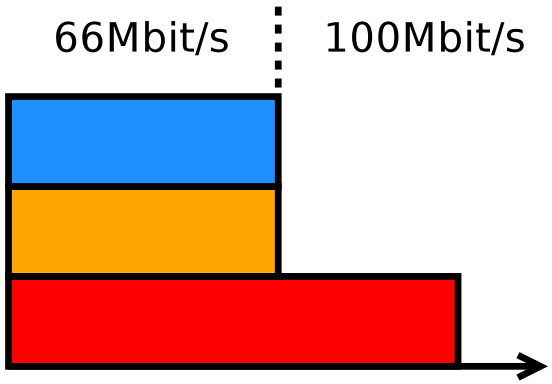
Exemple

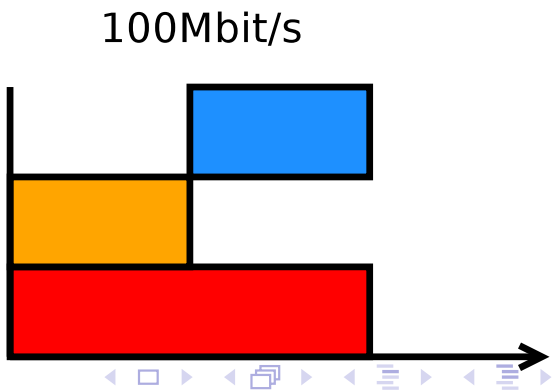




Temps : 2,5 secondes

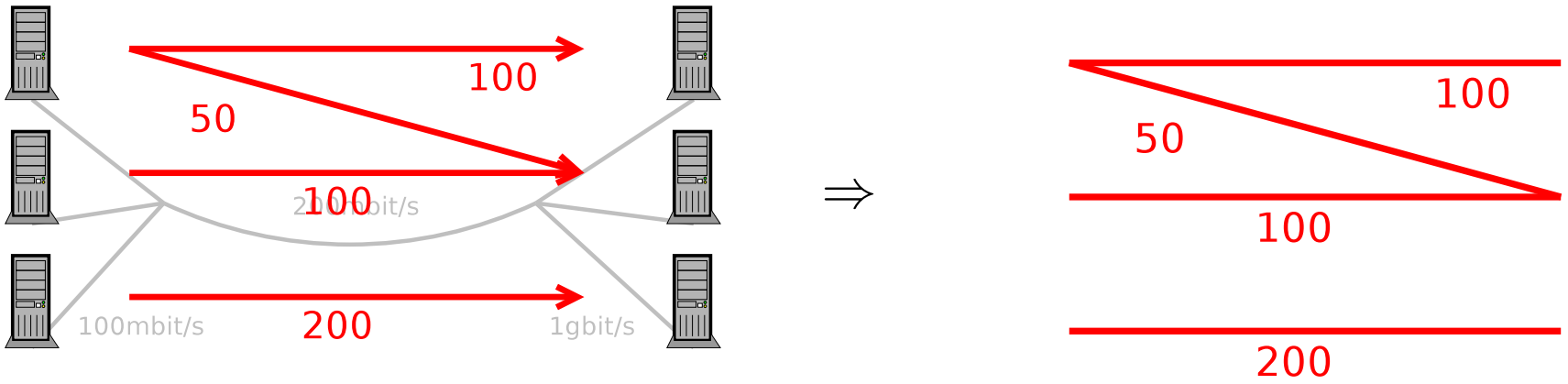
Temps : 2 secondes





Modélisation

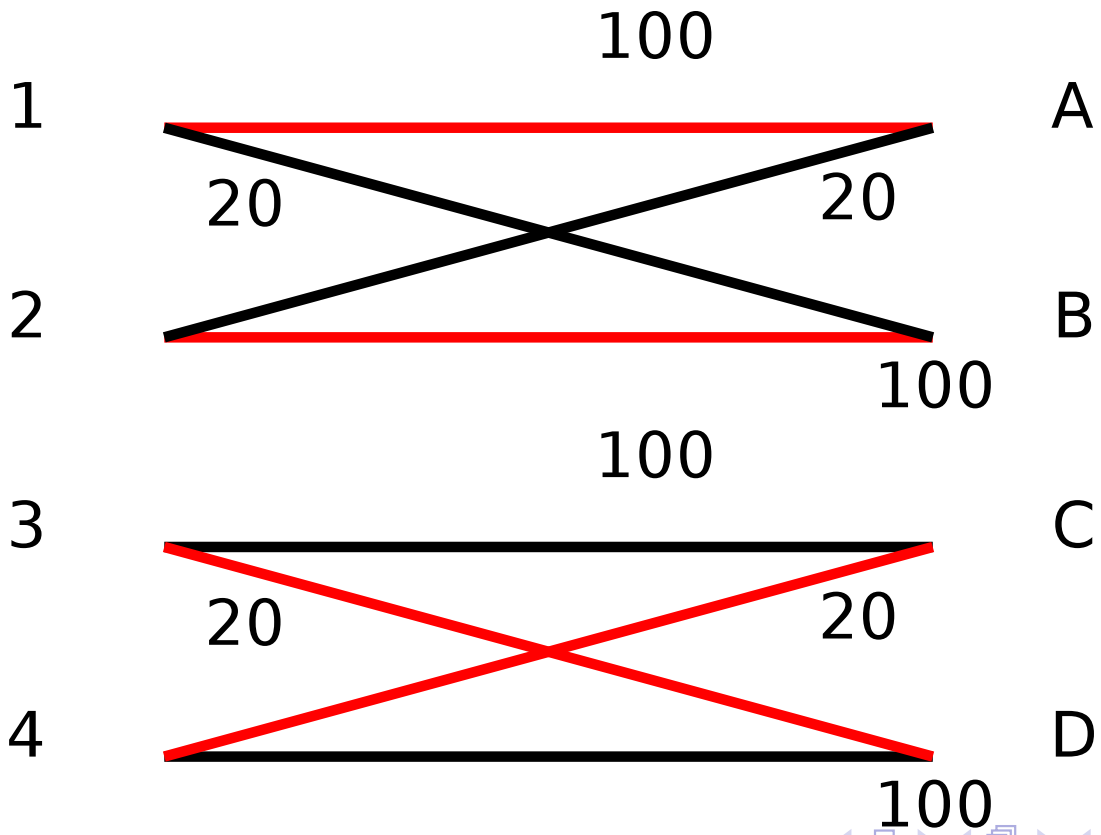
- modélisation par un graphe biparti valué
  - redistribution $\Leftrightarrow$ graphe biparti
  - nœud de calcul $\Leftrightarrow$ nœud du graphe
  - communication $\Leftrightarrow$ arête
  - durée minimale $\Leftrightarrow$ poids
- paramètres réseau
  - nombre maximal de communications simultanées autorisées : k
  - latence : β



Modélisation

Contraintes

- éviter de surcharger un lien
  - local : modèle 1-port
  - distant : k communications simultanées
- découpage en étapes de communications
  - 1 étape $\Leftrightarrow$ 1 couplage



Le problème KPBS [Cohen Jeannot Padoy]

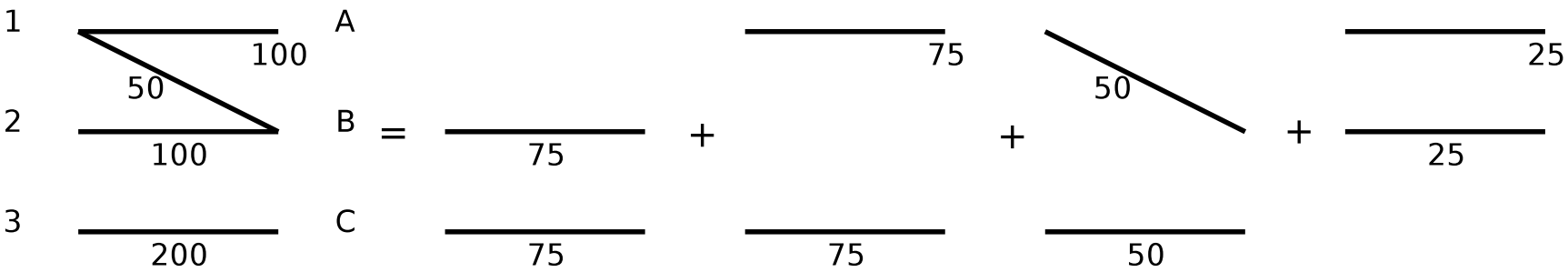
- K-Preemptive Bipartite Scheduling
- étant donné :
  - un graphe biparti correspondant au motif de données à redistribuer
  - les paramètres réseau : k, β
- décomposer le graphe
  - en un ensemble de couplages (1-port)
  - d'au plus k arêtes
  - correspondant à un temps minimal $(\sum_i D_i + \beta e)$
  - autorisant la préemption des communications
- problème NP-difficile

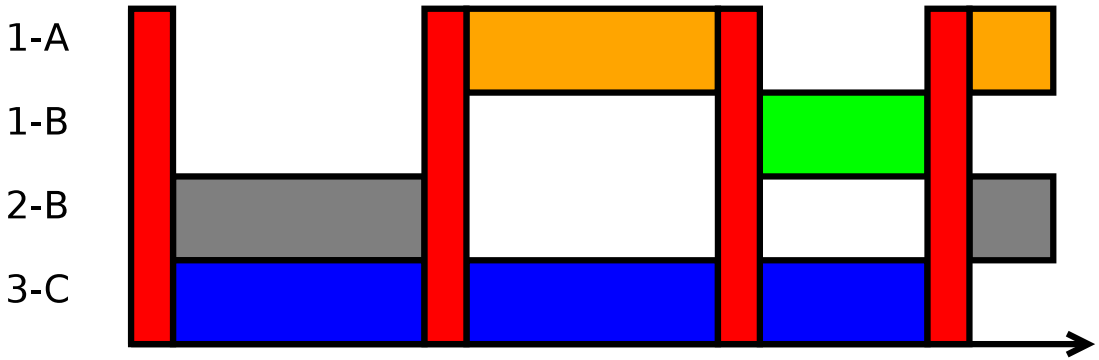
Le problème KPBS [Cohen Jeannot Padoy]

- K-Preemptive Bipartite Scheduling
- étant donné :
  - un graphe biparti correspondant au motif de données à redistribuer
  - les paramètres réseau : k, β
- décomposer le graphe
  - en un ensemble de couplages (1-port)
  - d'au plus k arêtes
  - correspondant à un temps minimal $(\sum_i D_i + \beta e)$
  - autorisant la préemption des communications
- problème NP-difficile

Le problème KPBS

Exemple ($k = 2$)



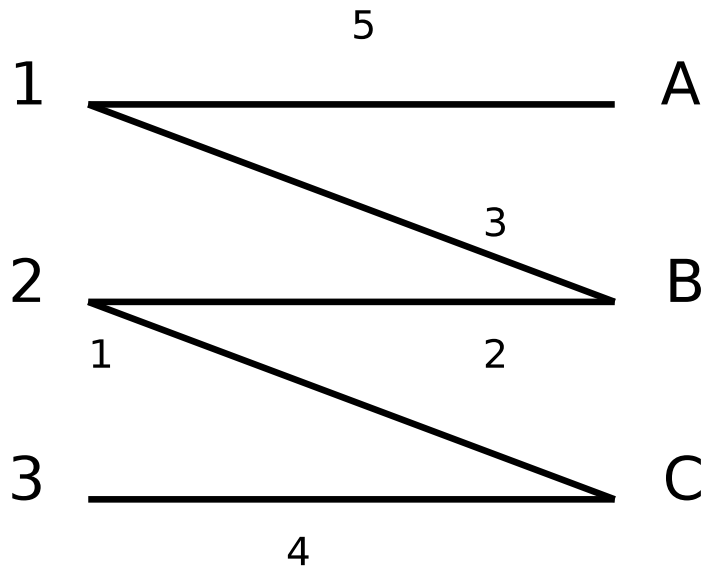


Coût total : $75 + 75 + 50 + 25 + 4 \times \beta$

Bornes sur le temps de redistribution

Notations

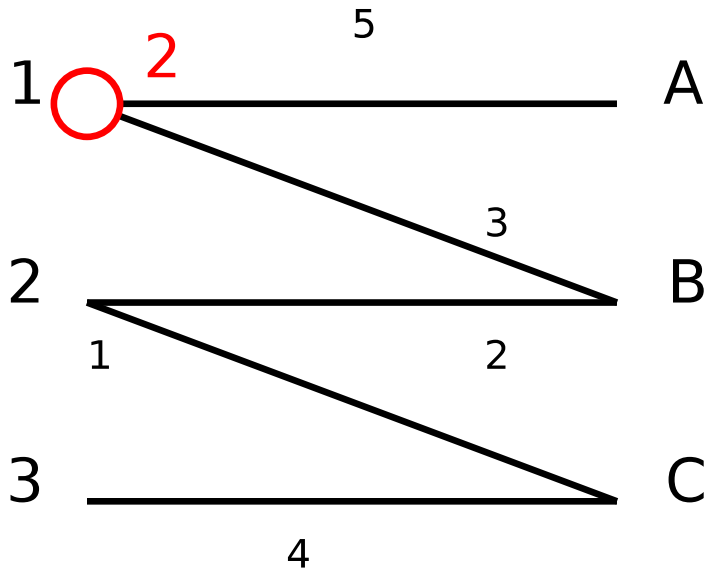
- Δ : degré du graphe
- m : nombre d'arêtes
- $w(s)$: poids d'un sommet s
- $W = \max_{s \in V_1 \cup V_2}(w(s))$
- $P = \sum_{e \in E}(w(e))$



Bornes sur le temps de redistribution

Notations

- Δ : degré du graphe
- m : nombre d'arêtes
- $w(s)$: poids d'un sommet s
- $W = \max_{s \in V_1 \cup V_2}(w(s))$
- $P = \sum_{e \in E}(w(e))$

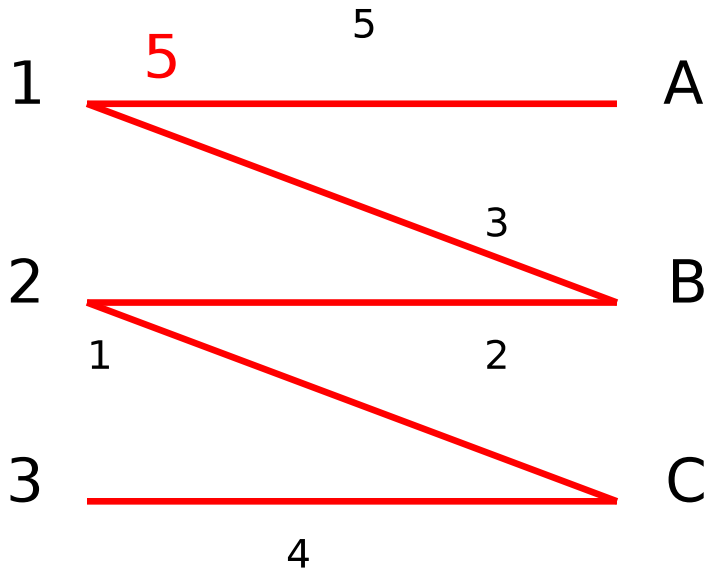


$\Delta = 2$

Bornes sur le temps de redistribution

Notations

- Δ : degré du graphe
- m : nombre d'arêtes
- $w(s)$: poids d'un sommet s
- $W = \max_{s \in V_1 \cup V_2}(w(s))$
- $P = \sum_{e \in E}(w(e))$

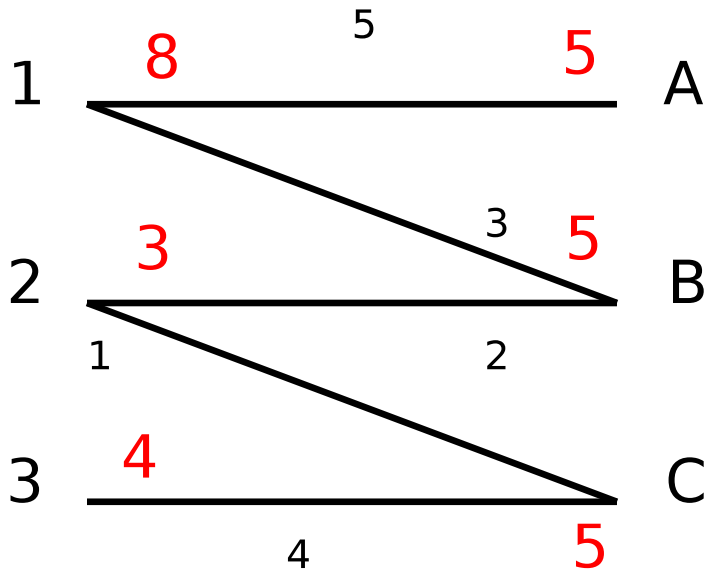


$m = 5$

Bornes sur le temps de redistribution

Notations

- Δ : degré du graphe
- m : nombre d'arêtes
- $w(s)$: poids d'un sommet s
- $W = \max_{s \in V_1 \cup V_2}(w(s))$
- $P = \sum_{e \in E}(w(e))$

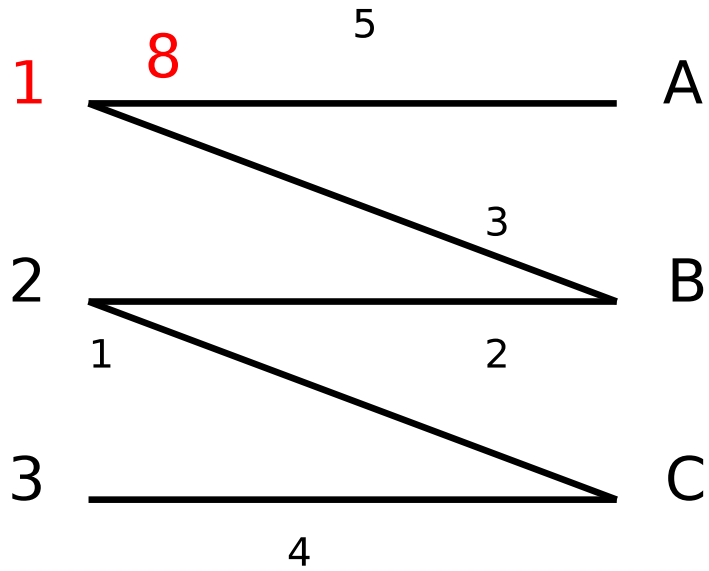


$$w(B) = 3 + 2$$

Bornes sur le temps de redistribution

Notations

- Δ : degré du graphe
- m : nombre d'arêtes
- $w(s)$: poids d'un sommet s
- $W = \max_{s \in V_1 \cup V_2}(w(s))$
- $P = \sum_{e \in E}(w(e))$

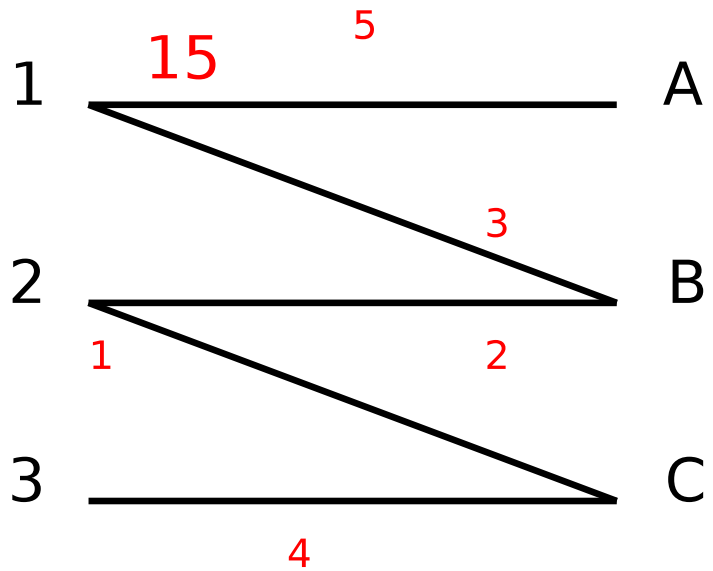


$W = 8$

Bornes sur le temps de redistribution

Notations

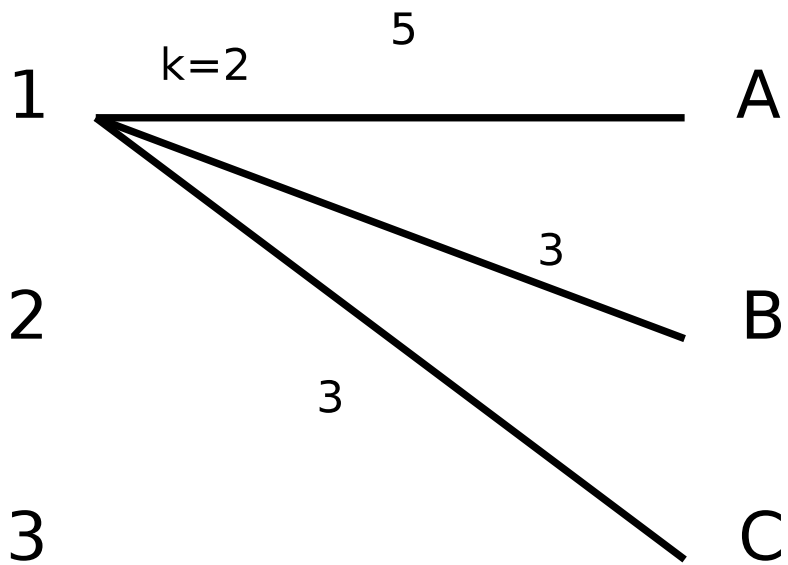
- Δ : degré du graphe
- m : nombre d'arêtes
- $w(s)$: poids d'un sommet s
- $W = \max_{s \in V_1 \cup V_2}(w(s))$
- $P = \sum_{e \in E}(w(e))$



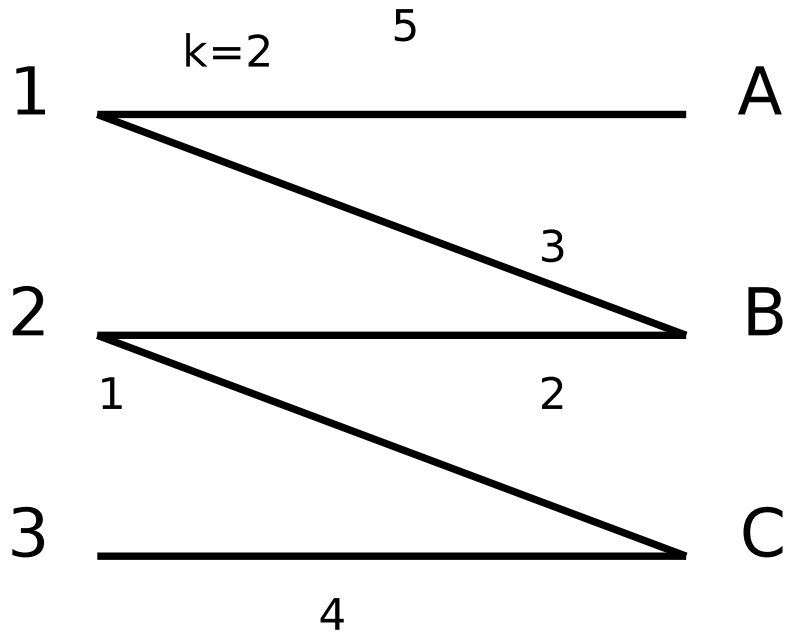
$P = 15$

Borne sur le nombre d'étapes

- au moins Δ étapes (1-port)
- au moins $\lceil \frac{m}{k} \rceil$ étapes (contrainte de bande passsante)
- borne inf : $\eta_e = \max(\Delta, \lceil \frac{m}{k} \rceil)$



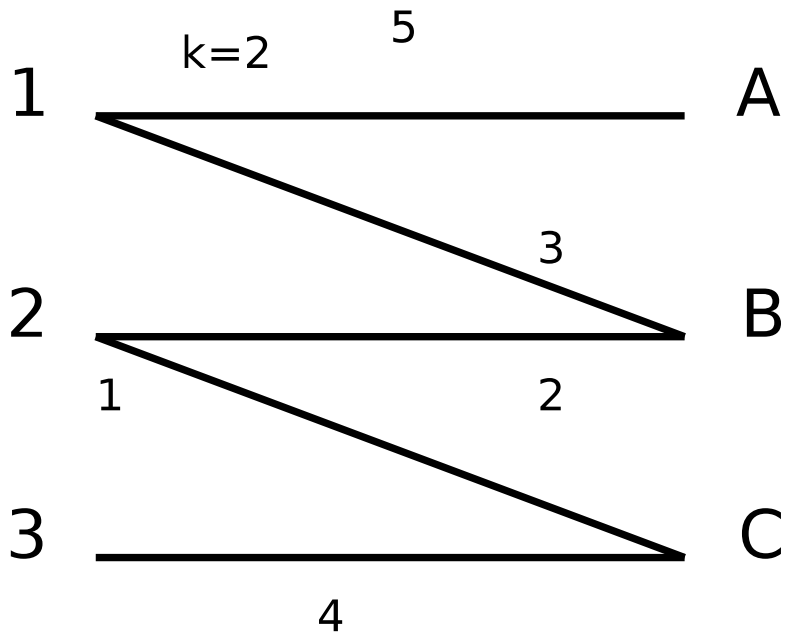
$\max(3, \lceil \frac{3}{2} \rceil) = 3$



$\max(2, \lceil \frac{5}{2} \rceil) = 3$

Borne sur les temps de transfert

- au moins W
- au moins $\frac{P}{k}$ (contrainte de bande passante)
- borne inf : $\eta_d = \max(W, \frac{P}{k})$

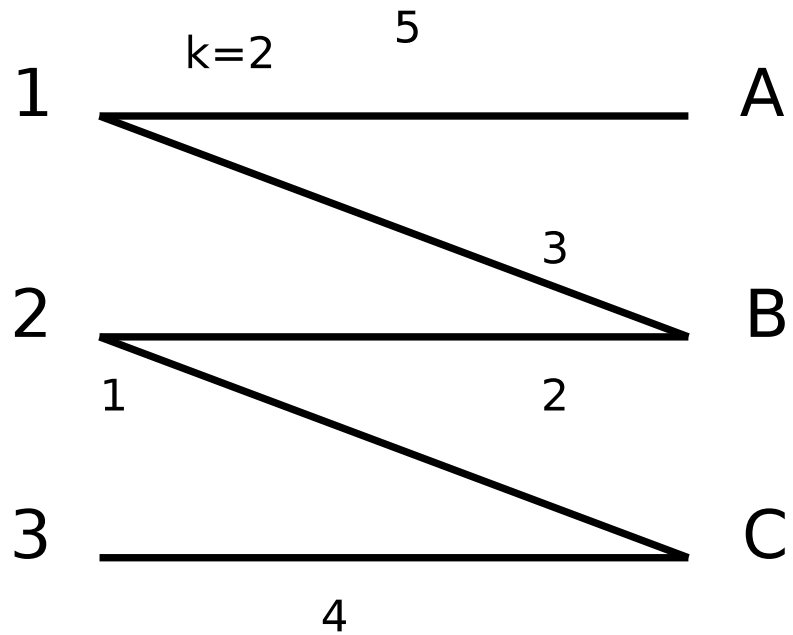


$\max(8, \frac{15}{2}) = 8$

$\max(5, \frac{11}{1}) = 11$

Bornes inférieures

- borne inf : $\eta = \eta_d + \eta_e \times \beta$
- η_e atteignable
- η_d atteignable
- η n'est pas toujours atteignable



$$\max(8, \tfrac{15}{2}) + \max(2, \lceil \tfrac{5}{2} \rceil) \times \beta = 8 + 3\beta$$

Plan de l'exposé

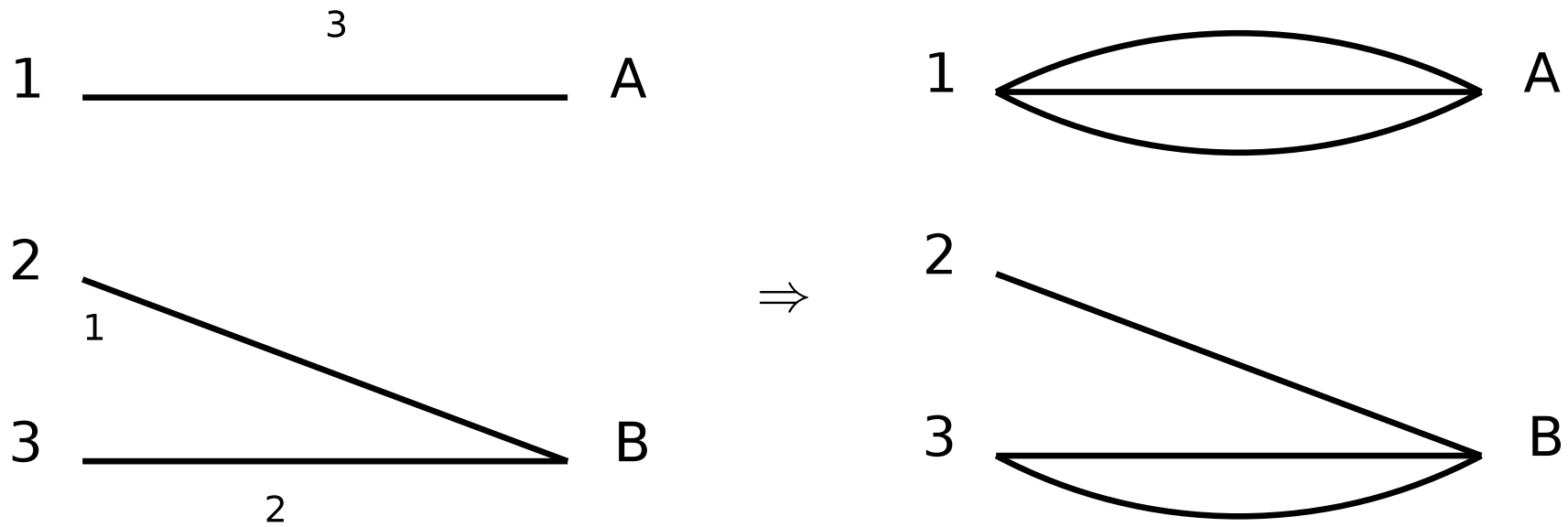
1. Redistributions
2. Le problème KPBS
   - Présentation
   - Modélisation
   - Bornes inférieures
3. Algorithme pseudo-polynomial
4. Algorithme polynomial (GGP)
5. Conclusion

Vers un problème plus simple

- objectif : toutes les communications de taille identique
- principe : plus de préemption à utiliser
- pour une étape donnée, toutes les communications de même taille
- transformation à partir du problème initial

Vers un problème plus simple

- objectif : toutes les communications de taille identique
- principe : plus de préemption à utiliser
- pour une étape donnée, toutes les communications de même taille
- transformation à partir du problème initial

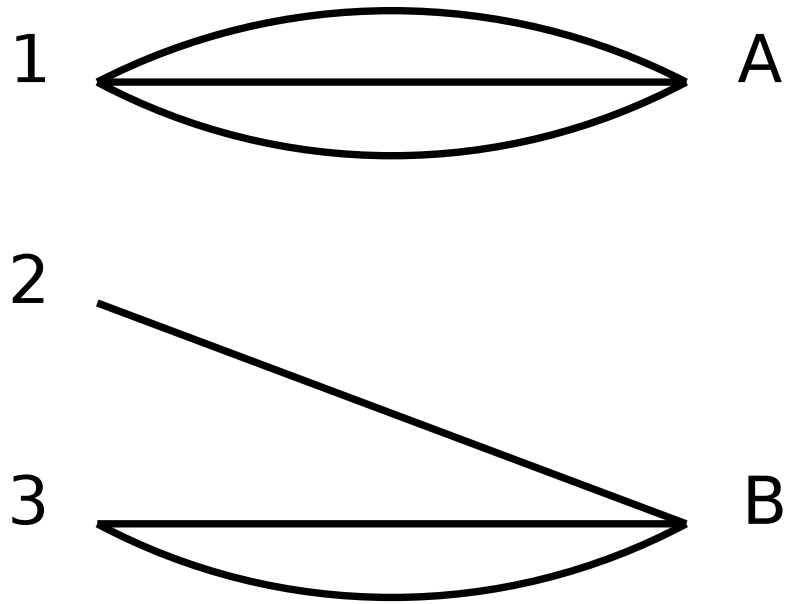


Algorithme pseudo-polynomial

principe

- rendre le multigraphe régulier
- choisir un couplage parfait comme première étape
- enlever le couplage du graphe
- choisir un couplage parfait

- enlever le couplage
- choisir un couplage

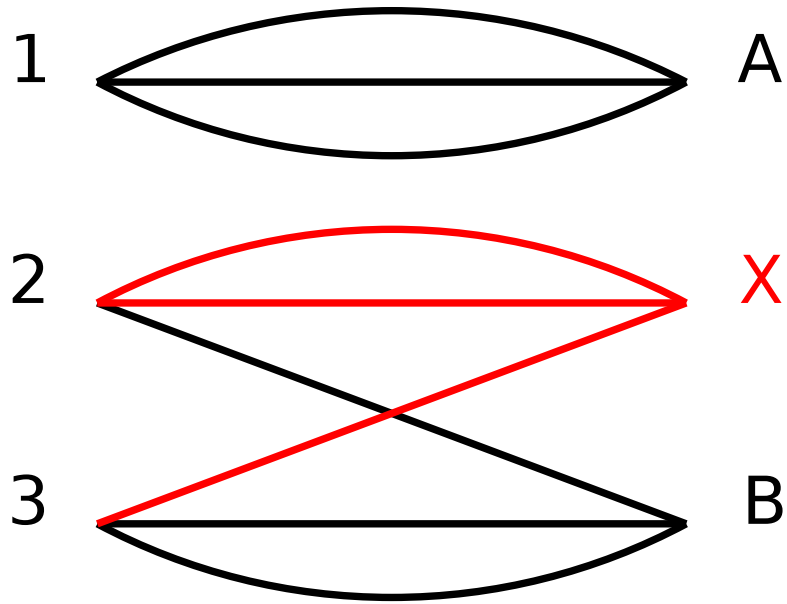


Algorithme pseudo-polynomial

principe

- rendre le multigraphe régulier
- choisir un couplage parfait comme première étape
- enlever le couplage du graphe
- choisir un couplage parfait

- enlever le couplage
- choisir un couplage

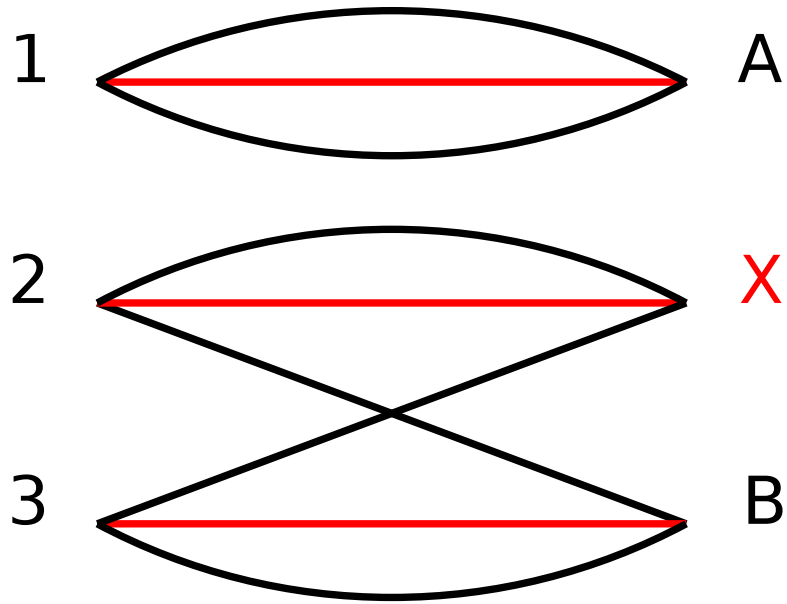


Algorithme pseudo-polynomial

principe

- rendre le multigraphe régulier
- choisir un couplage parfait comme première étape
- enlever le couplage du graphe
- choisir un couplage parfait

- enlever le couplage
- choisir un couplage

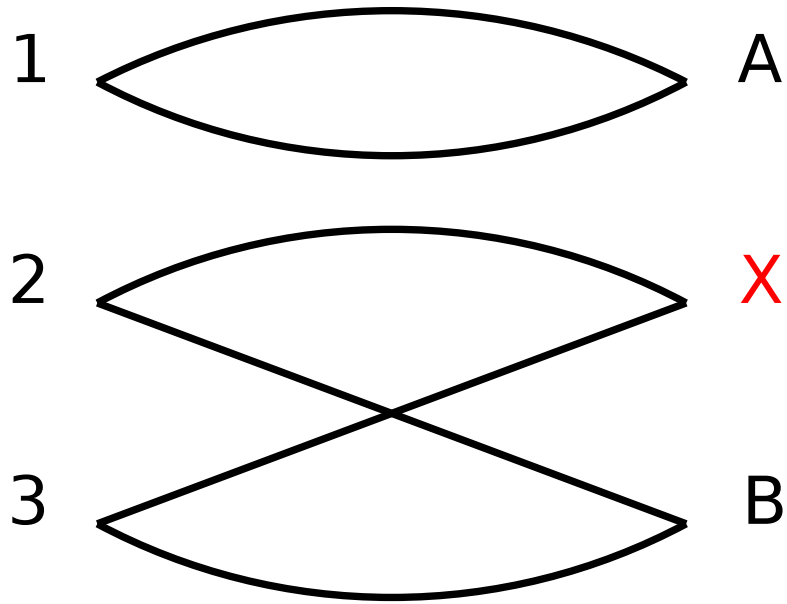


Algorithme pseudo-polynomial

principe

- rendre le multigraphe régulier
- choisir un couplage parfait comme première étape
- enlever le couplage du graphe
- choisir un couplage parfait

- enlever le couplage
- choisir un couplage

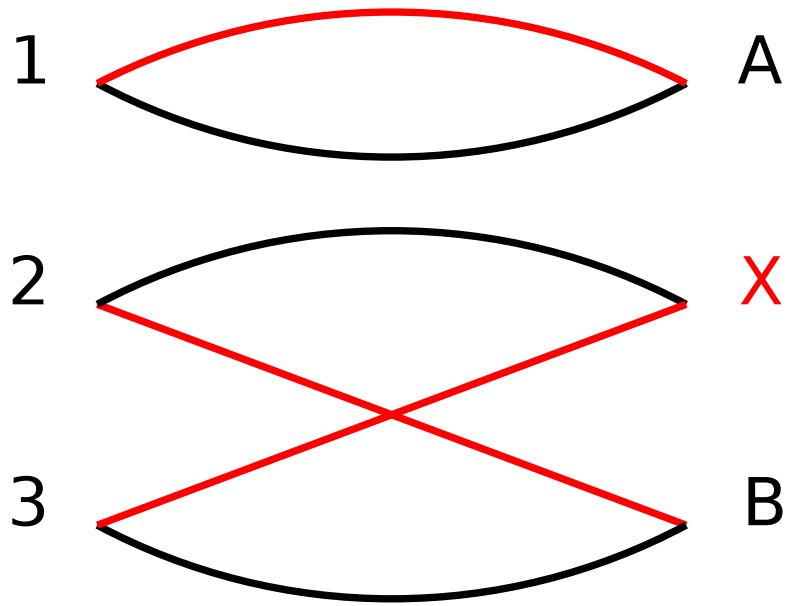


Algorithme pseudo-polynomial

principe

- rendre le multigraphe régulier
- choisir un couplage parfait comme première étape
- enlever le couplage du graphe
- choisir un couplage parfait

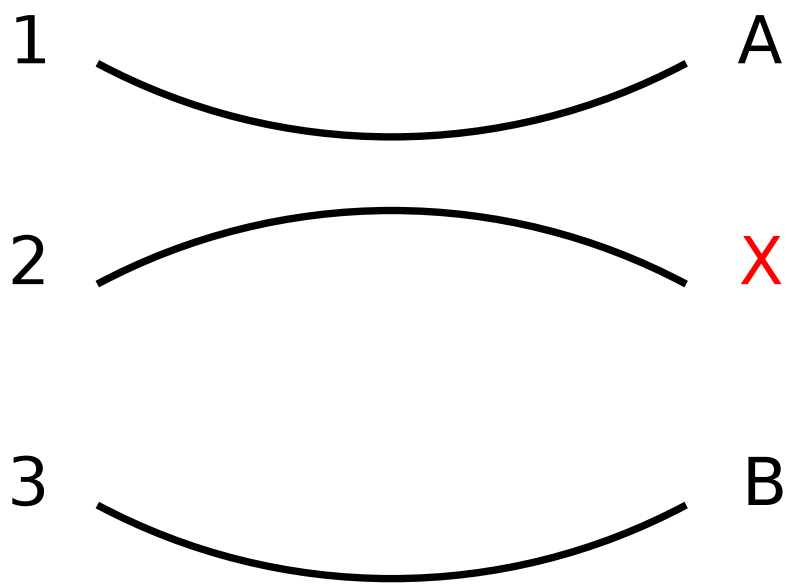
- enlever le couplage
- choisir un couplage



Algorithme pseudo-polynomial

principe

- rendre le multigraphe régulier
- choisir un couplage parfait comme première étape
- enlever le couplage du graphe
- choisir un couplage parfait
- enlever le couplage
- choisir un couplage

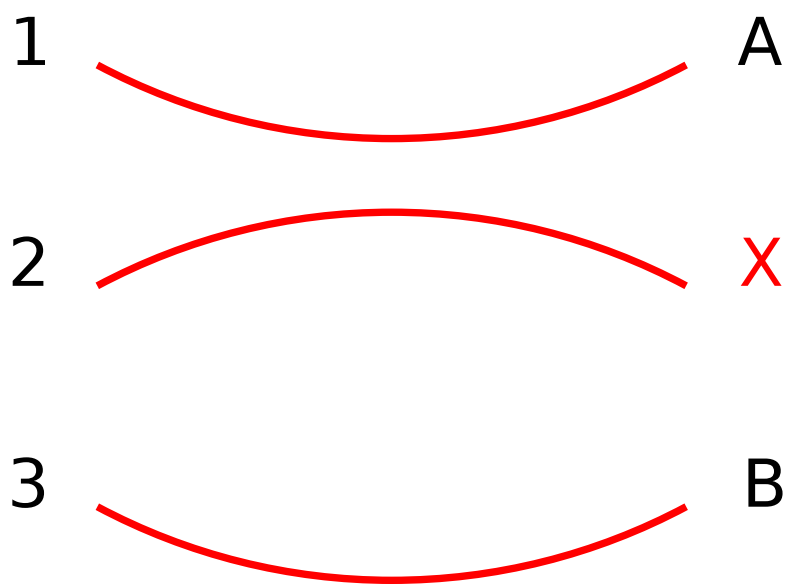


Algorithme pseudo-polynomial

principe

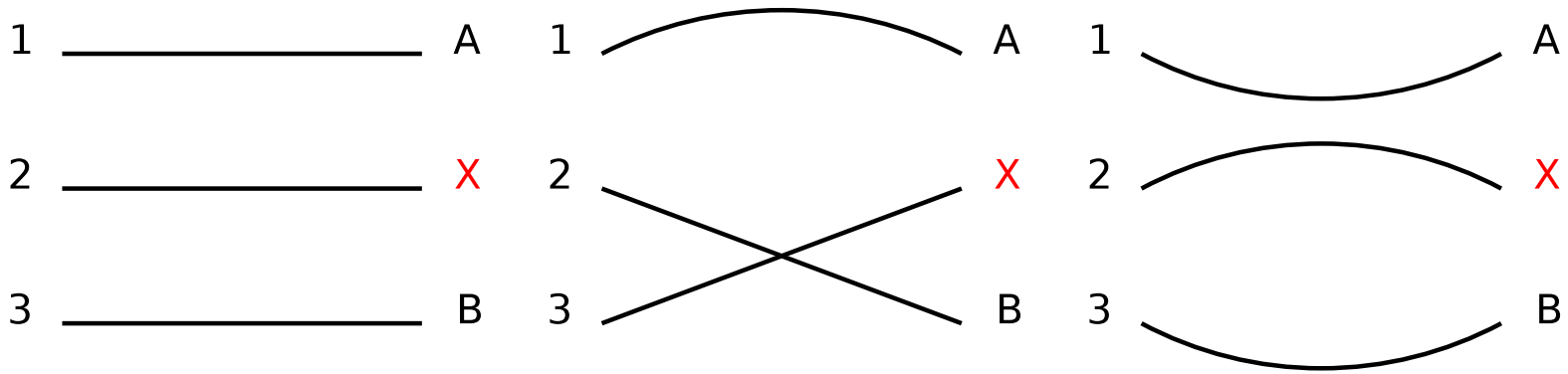
- rendre le multigraphe régulier
- choisir un couplage parfait comme première étape
- enlever le couplage du graphe
- choisir un couplage parfait

- enlever le couplage
- choisir un couplage



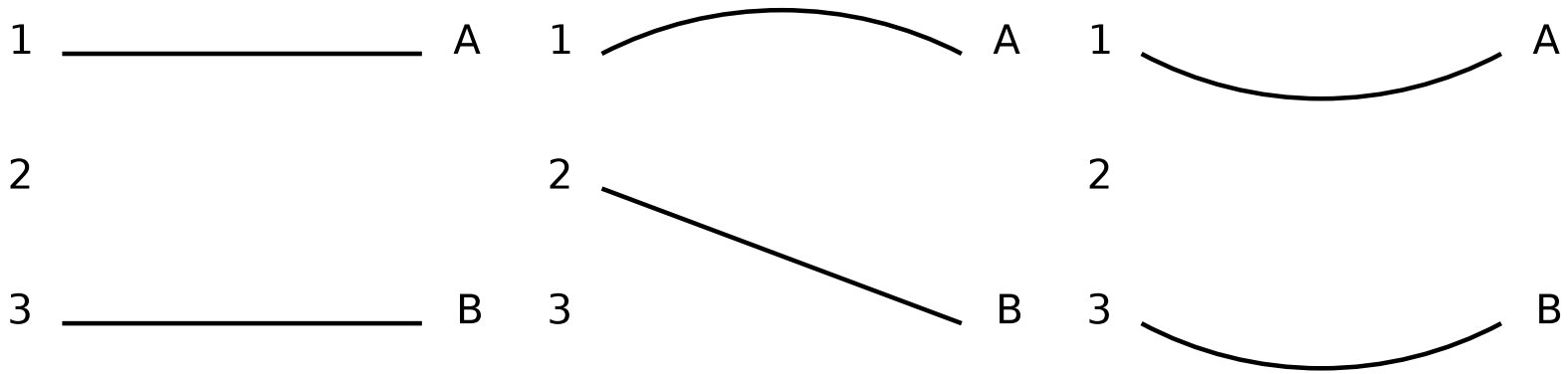
Résultats

- ensemble de couplages
- k arêtes par couplage ?
- éliminer les arêtes virtuelles



Résultats

- ensemble de couplages
- k arêtes par couplage ?
- éliminer les arêtes virtuelles

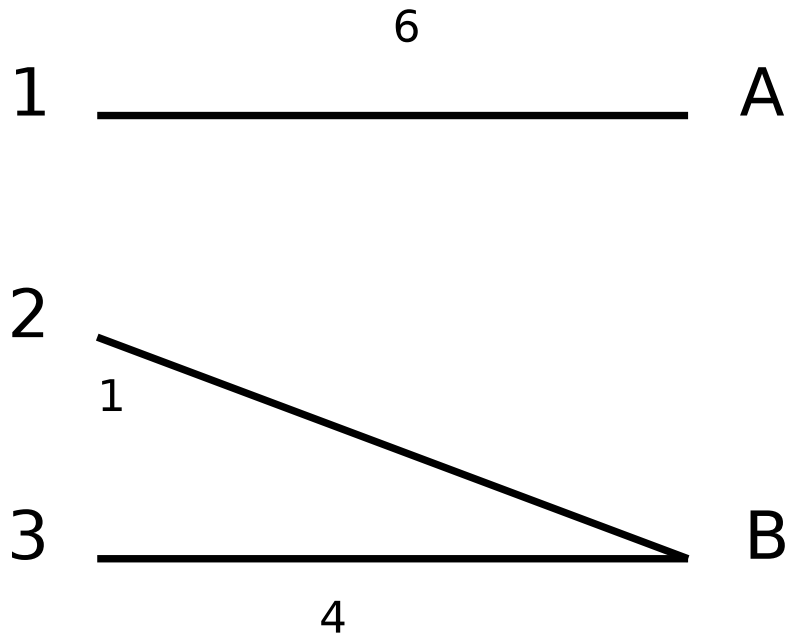


Propriétés

- soit G le graphe initial et G' le graphe régulier
- $\Delta(G')$ étapes de durée 1
- durée totale : $\beta\Delta(G') + \Delta(G') = 2\Delta(G')$ si $\beta = 1$
- ajout d'arêtes tel que $\Delta(G') = \max(\Delta(G), \left\lceil \frac{m(G)}{k} \right\rceil) = \eta_e(G)$
- durée totale : $2\eta_e(G)$

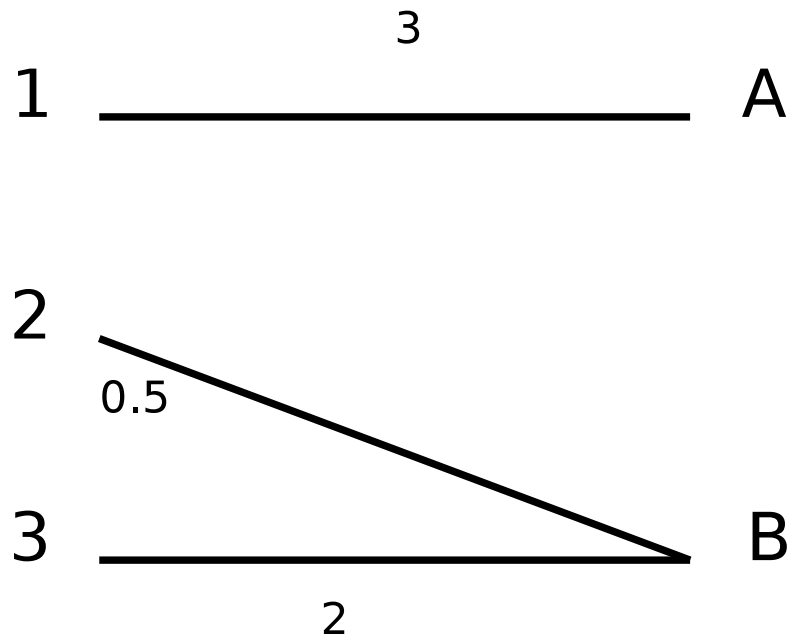
Algorithme complet

- durée d'une étape : β
- normalisation du problème par β
- arrondi
- construction du multigraphe
- calcul des étapes
- élimination de l'arrondi



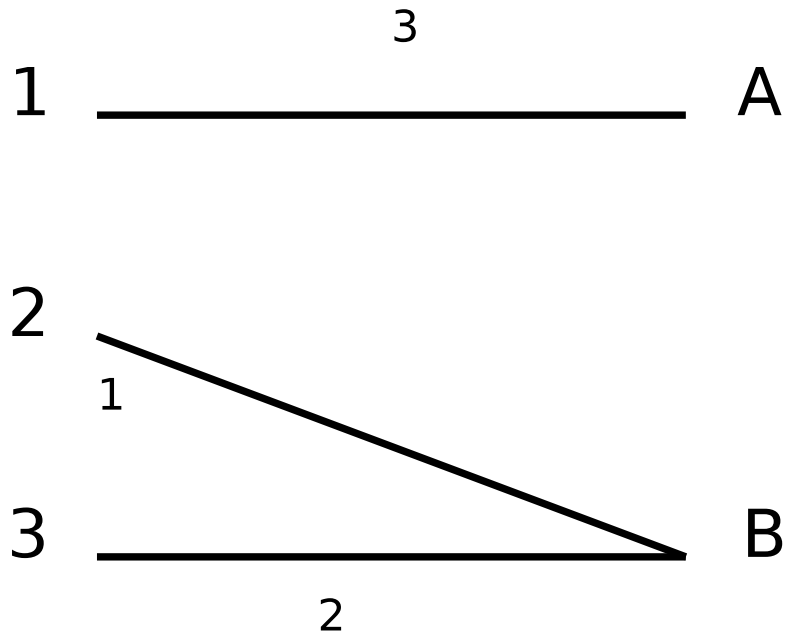
Algorithme complet

- durée d'une étape : β
- normalisation du problème par β
- arrondi
- construction du multigraphe
- calcul des étapes
- élimination de l'arrondi



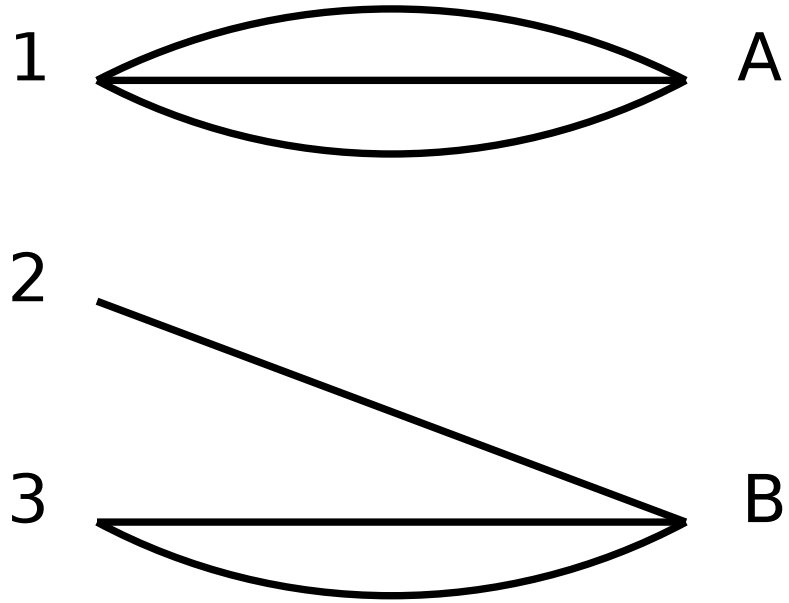
Algorithme complet

- durée d'une étape : β
- normalisation du problème par β
- arrondi
- construction du multigraphe
- calcul des étapes
- élimination de l'arrondi



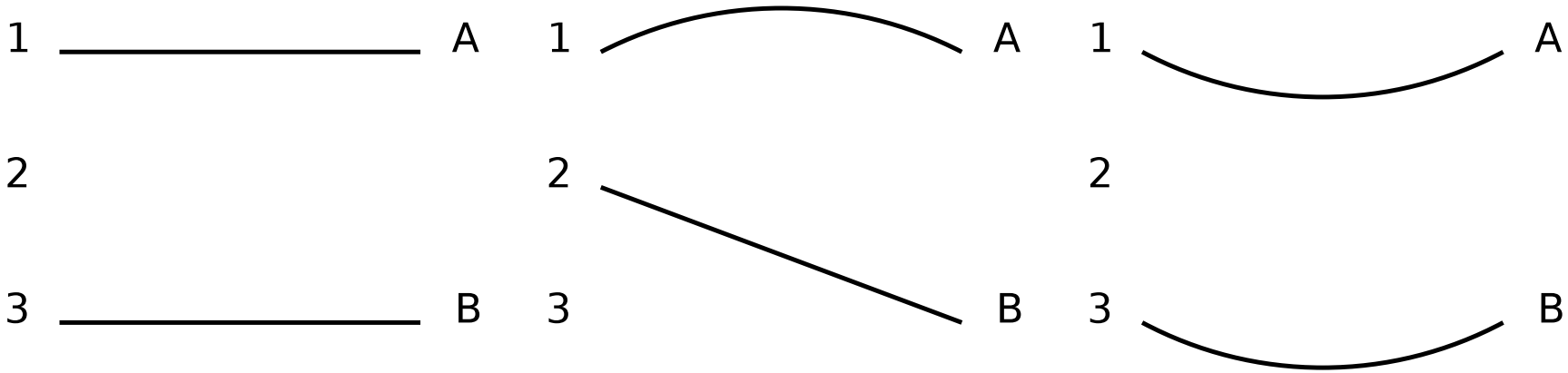
Algorithme complet

- durée d'une étape : β
- normalisation du problème par β
- arrondi
- construction du multigraphe
- calcul des étapes
- élimination de l'arrondi



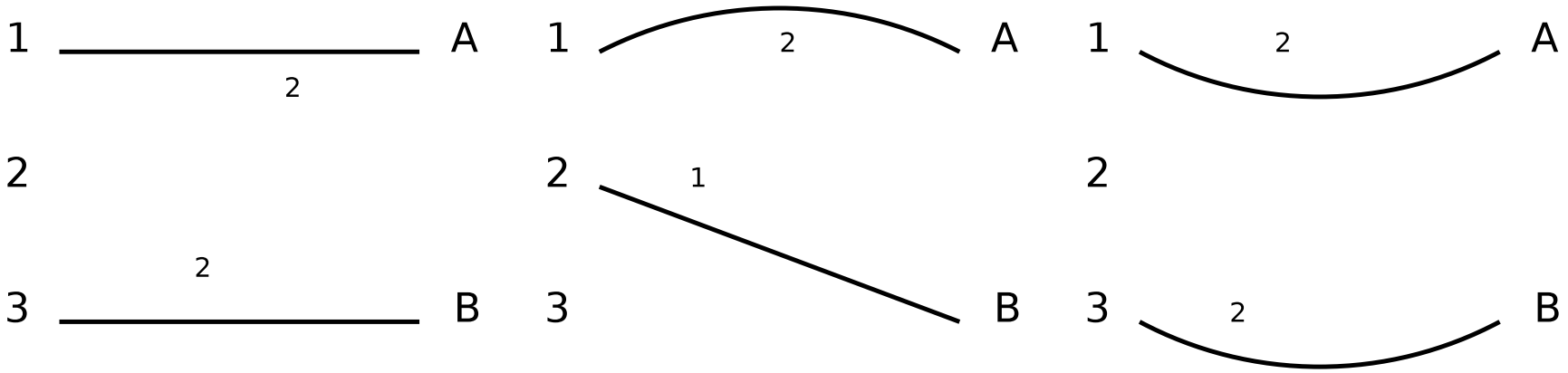
Algorithme complet

- durée d'une étape : β
- normalisation du problème par β
- arrondi
- construction du multigraphe
- calcul des étapes
- élimination de l'arrondi



Algorithme complet

- durée d'une étape : β
- normalisation du problème par β
- arrondi
- construction du multigraphe
- calcul des étapes
- élimination de l'arrondi



Propriétés

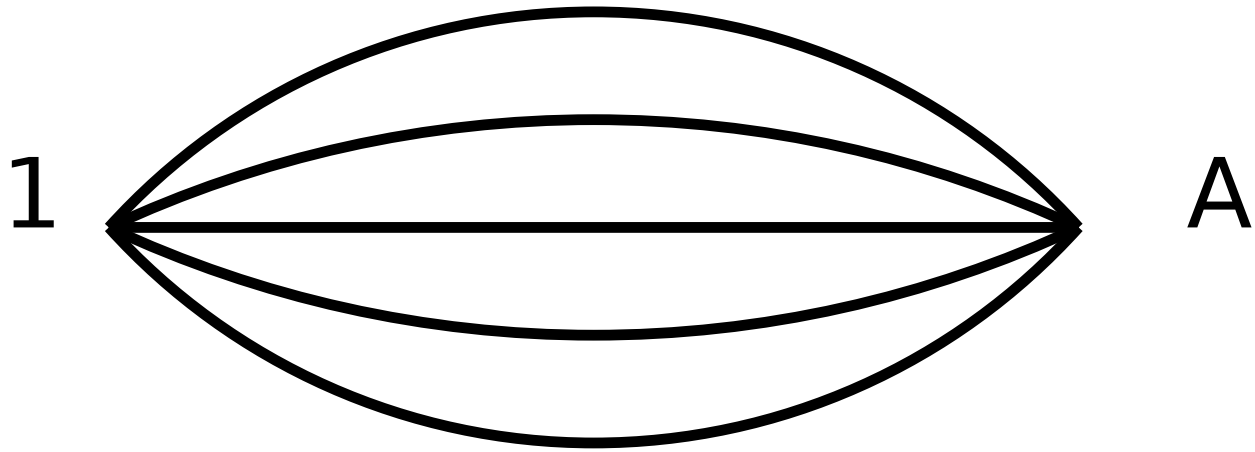
- ratio d'approximation de $\frac{8}{3}$
- pseudo-polynomial (poids des arêtes)
- grand nombre d'étapes si β faible devant les comms

Plan de l'exposé

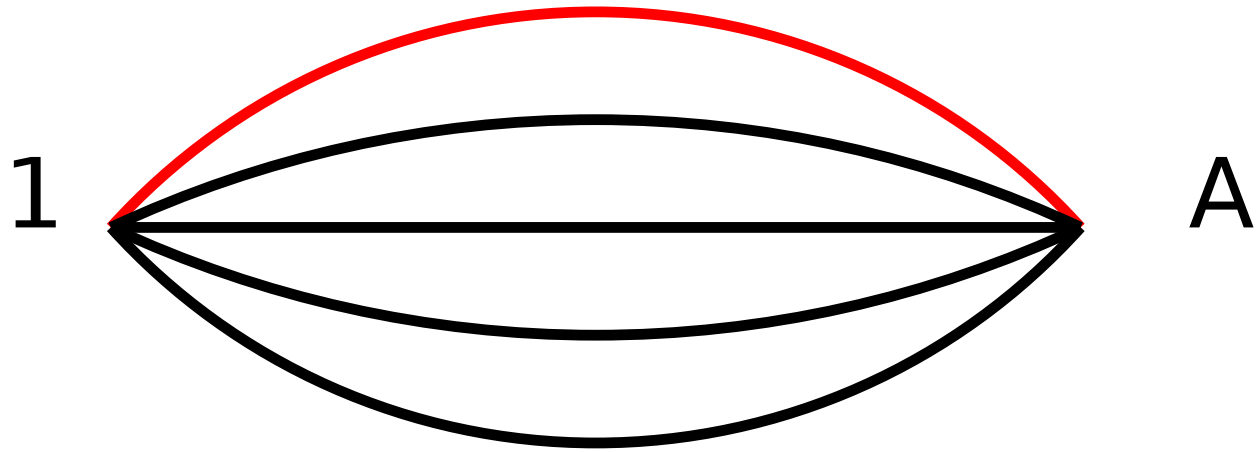
1. Redistributions
2. Le problème KPBS
   - Présentation
   - Modélisation
   - Bornes inférieures
3. Algorithme pseudo-polynomial
4. Algorithme polynomial (GGP)
5. Conclusion

Exemple

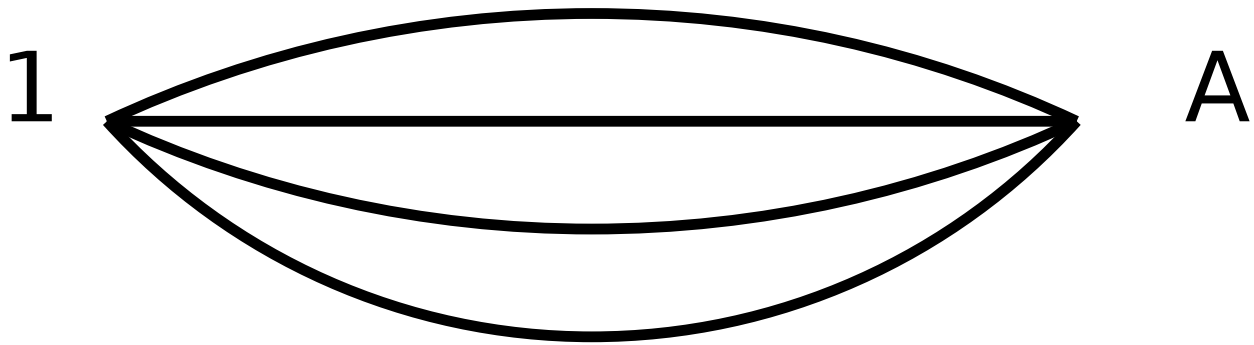
Exemple



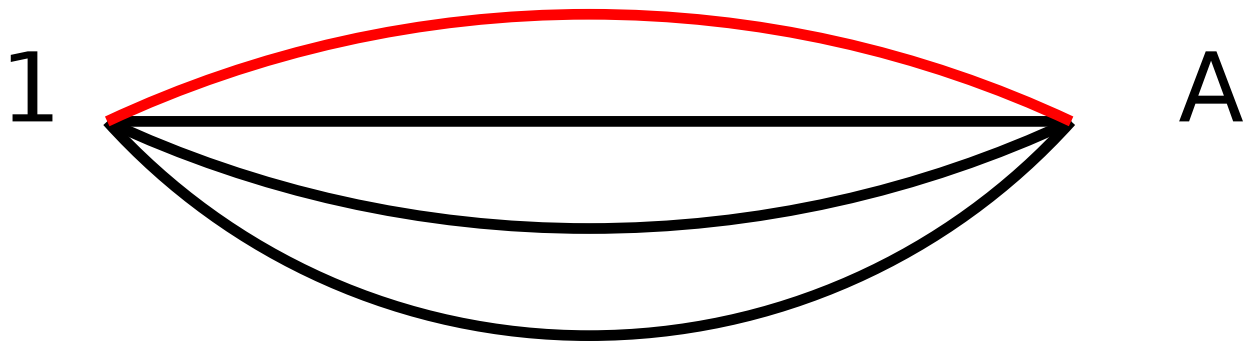
Exemple



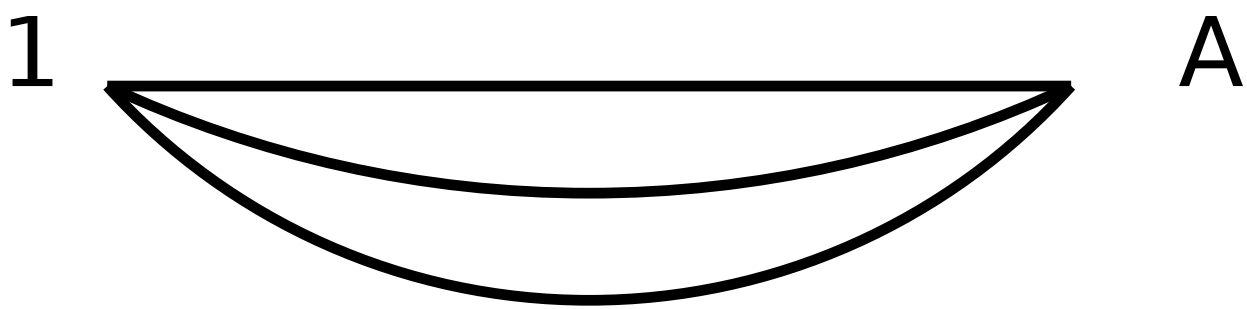
Exemple



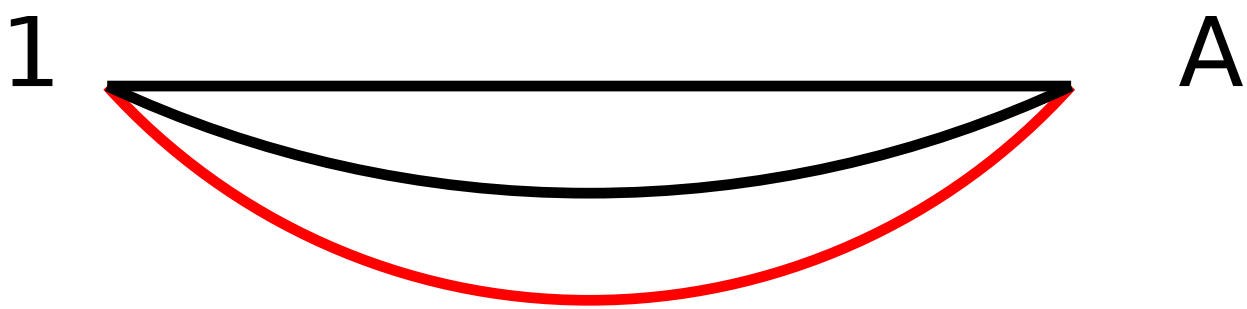
Exemple



Exemple



Exemple

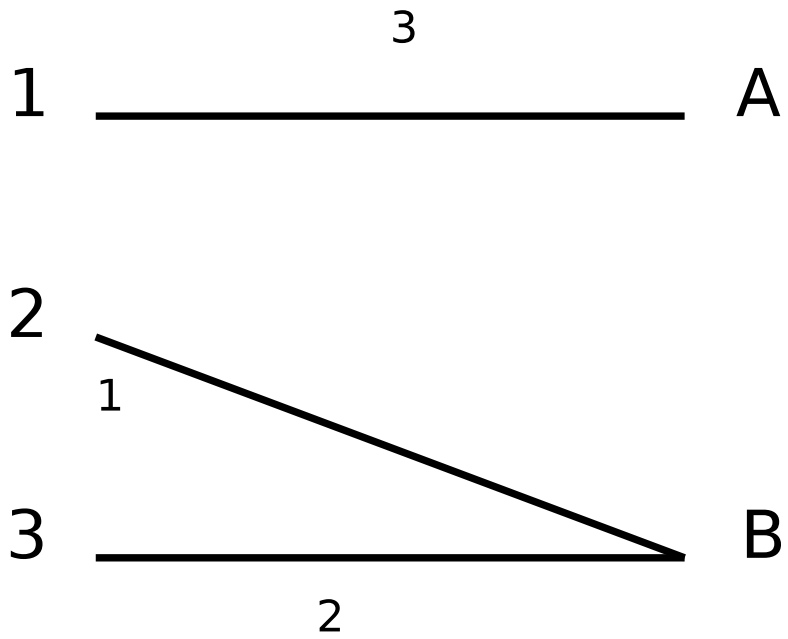


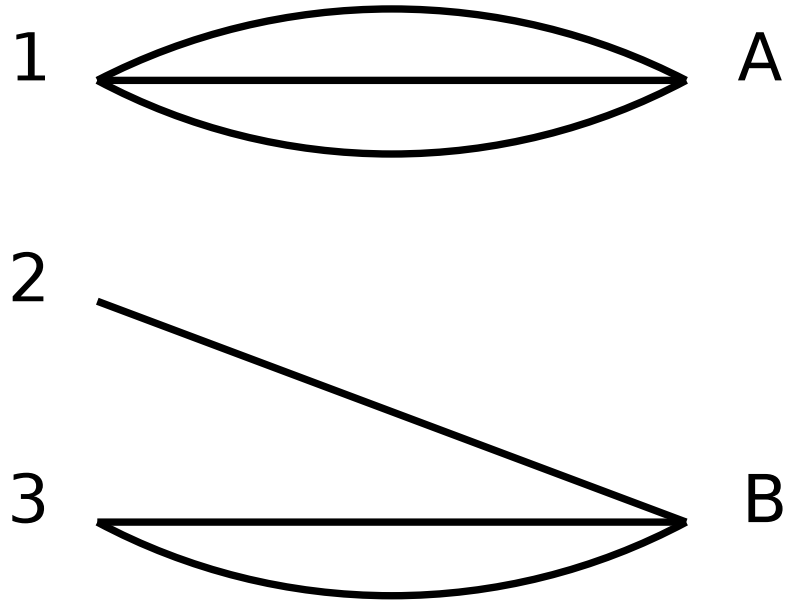
Principe de GGP

- factoriser les opérations répétitives de l'algo précédent
  - prendre n fois le même couplage de taille 1 sur un multigraphe == prendre 1 couplage de taille n sur un graphe
  - fusion des étapes identiques
- travailler sur des graphes au lieu de multigraphes
- garder l'algorithme identique (garder le ratio d'approximation)

Correspondances

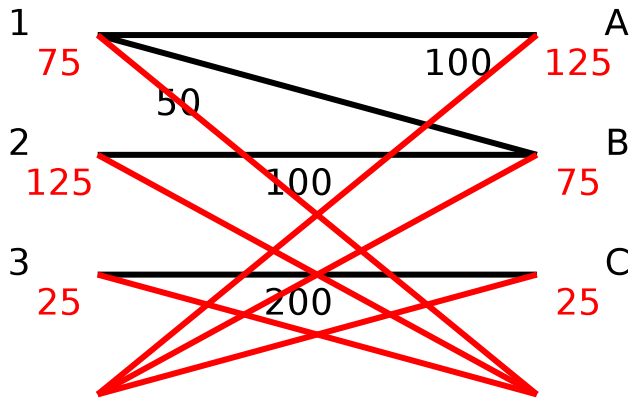
- à n'importe quelle étape, équivalence graphe $G \Leftrightarrow$ multigraphe G'
- arête de poids $n \Leftrightarrow n$ arêtes de poids 1
- formules :
  - $\Delta(G') = W(G)$
  - $m(G') = P(G)$
  - $\eta_e(G') = \eta(G)$
  - ...





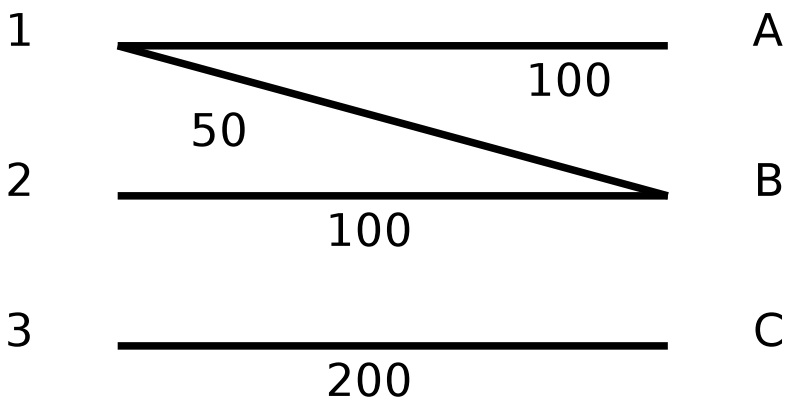
GGP [IPDPS2004][IEEE TPDS]

- Generic Graph Peeling
- algorithme en 3 étapes
  1. normalisation des poids par β
  2. extension du graphe en graphe régulier sur les poids
     - ★ prise en compte de k
  3. découpage du graphe en couplages
     - ★ choix d'un couplage parfait $\Leftrightarrow$ étape de la solution
     - ★ calcul du poids le plus faible du couplage, mise-à-jour des poids
     - ★ enlever le couplage obtenu du graphe
     - ★ boucler sur cette étape jusqu'à ce que le graphe soit vide



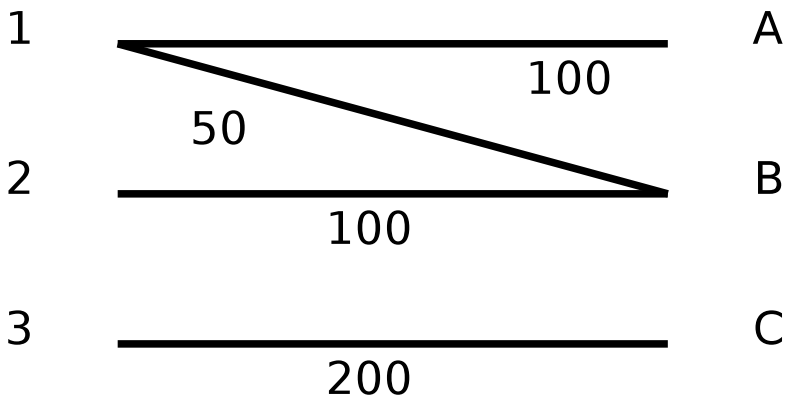
GGP

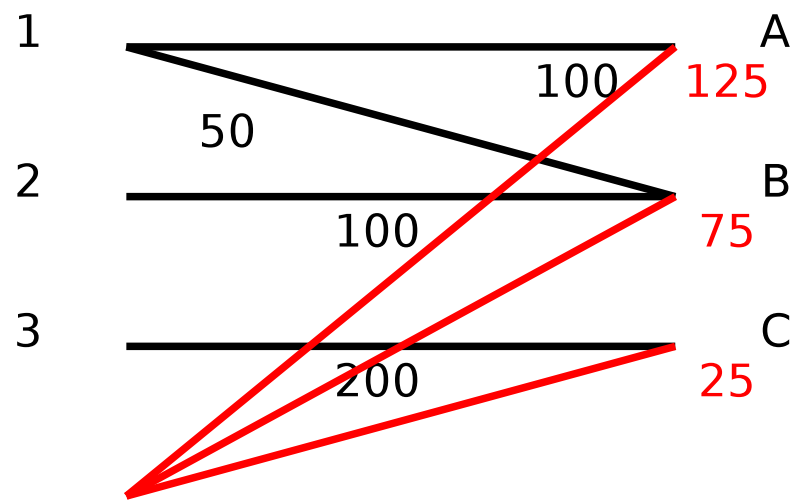
Exemple ($k = 2$)



GGP

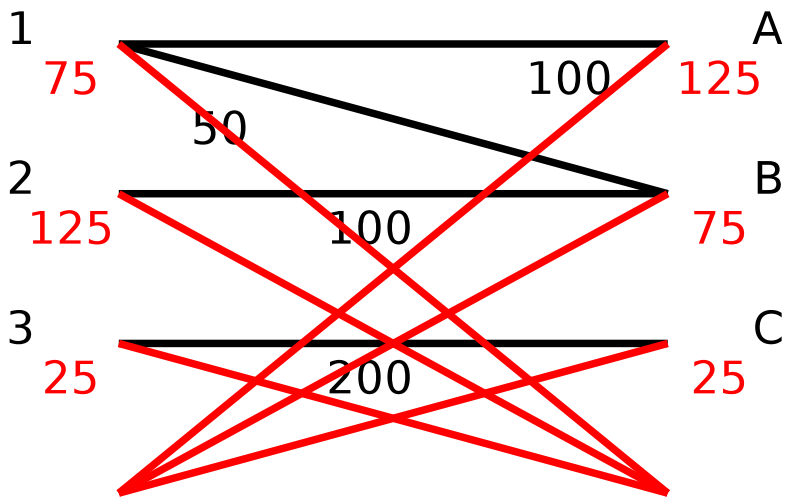
Exemple ($k = 2$)





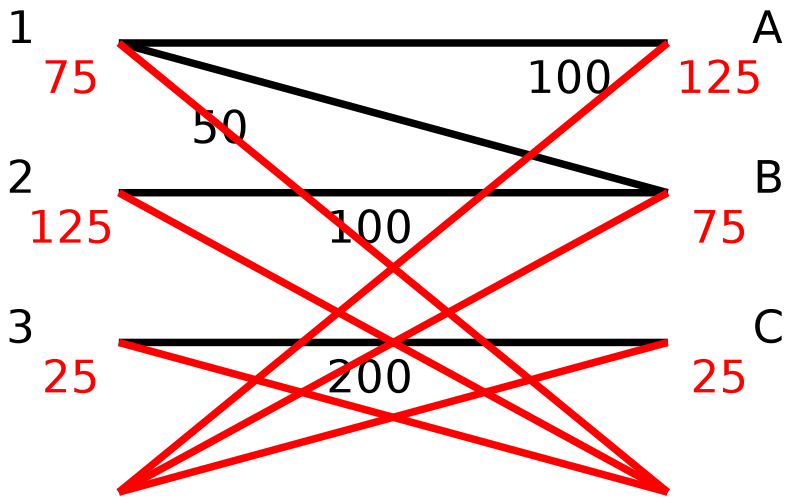
GGP

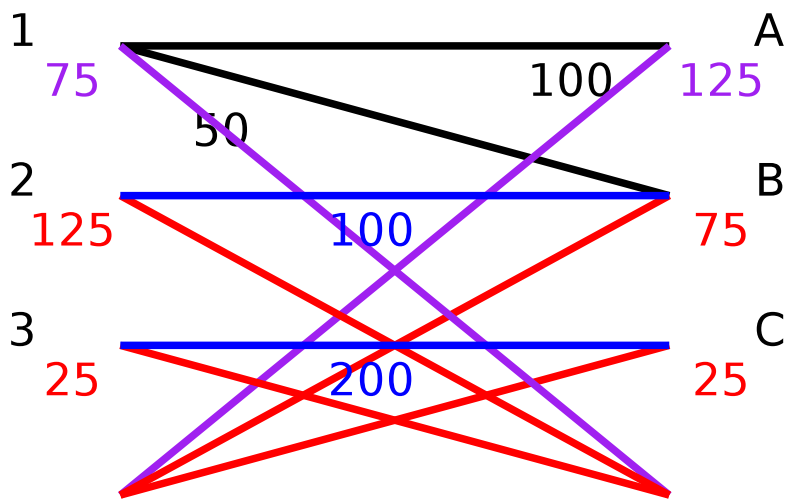
Exemple ($k = 2$)



GGP

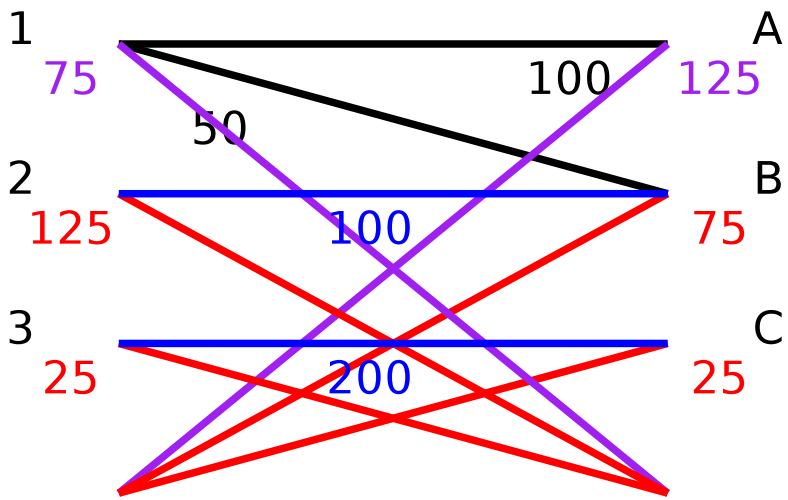
Exemple ($k = 2$)





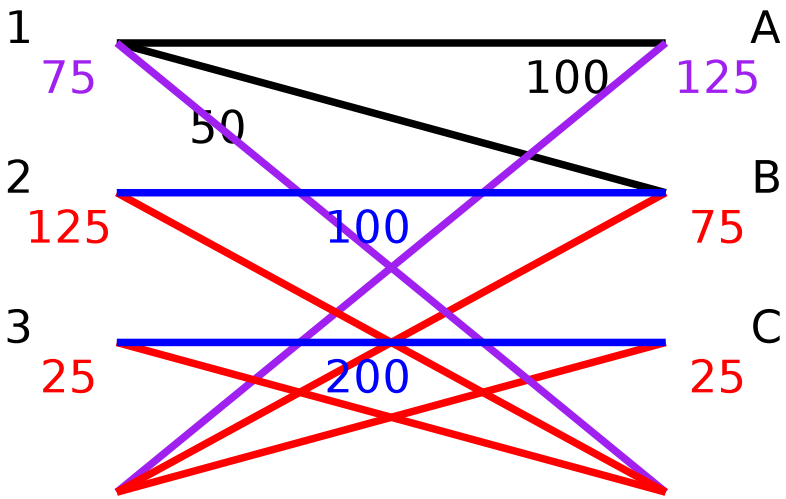
GGP

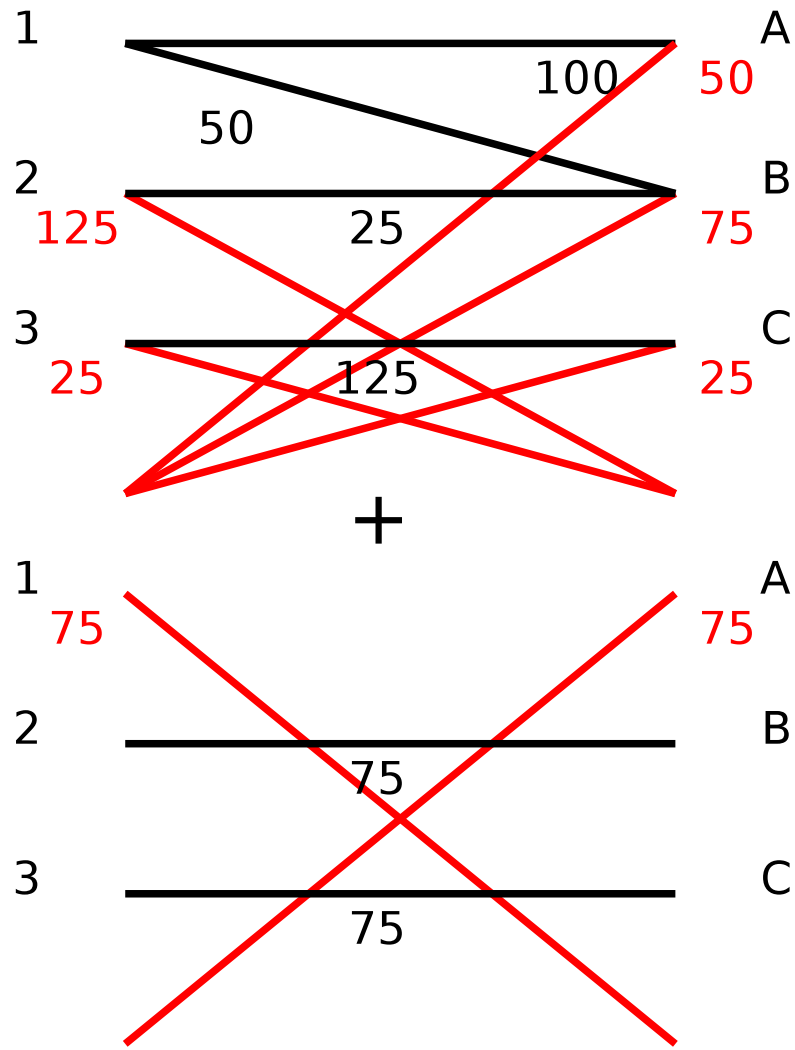
Exemple ($k = 2$)



GGP

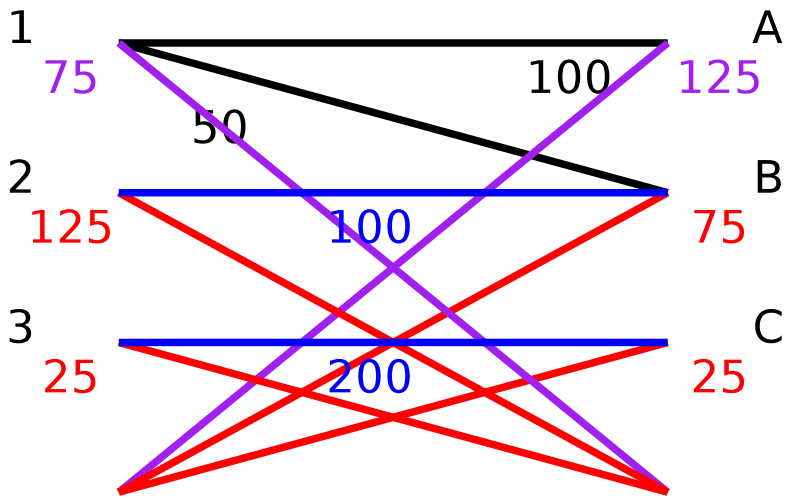
Exemple ($k = 2$)

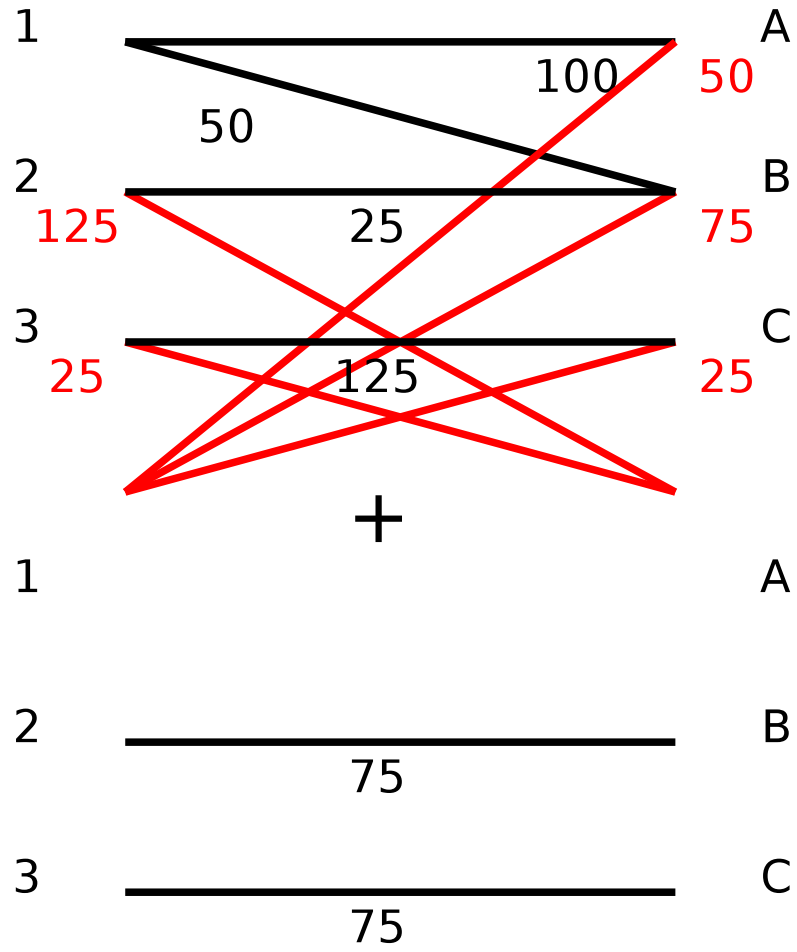




GGP

Exemple ($k = 2$)





GGP

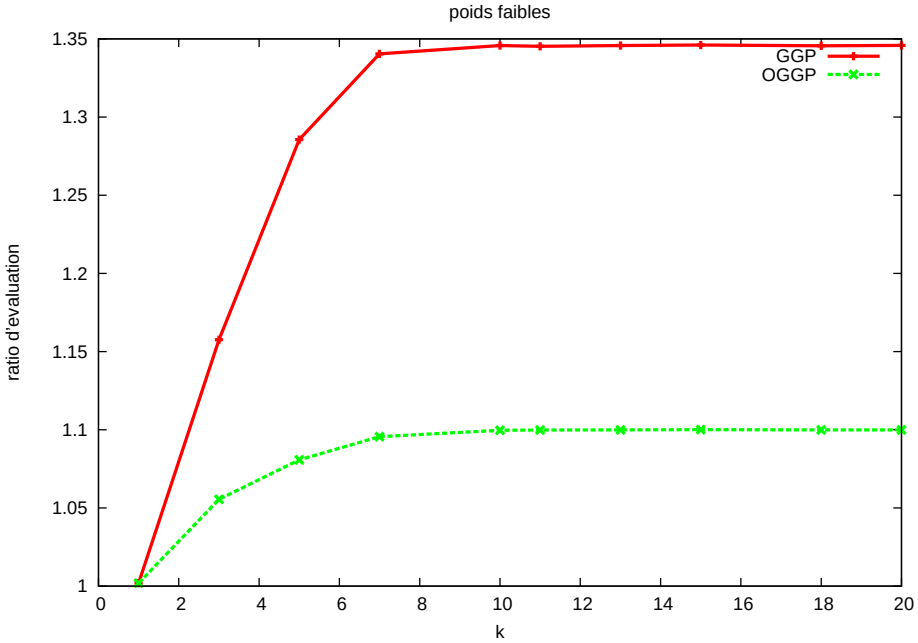
Propriétés

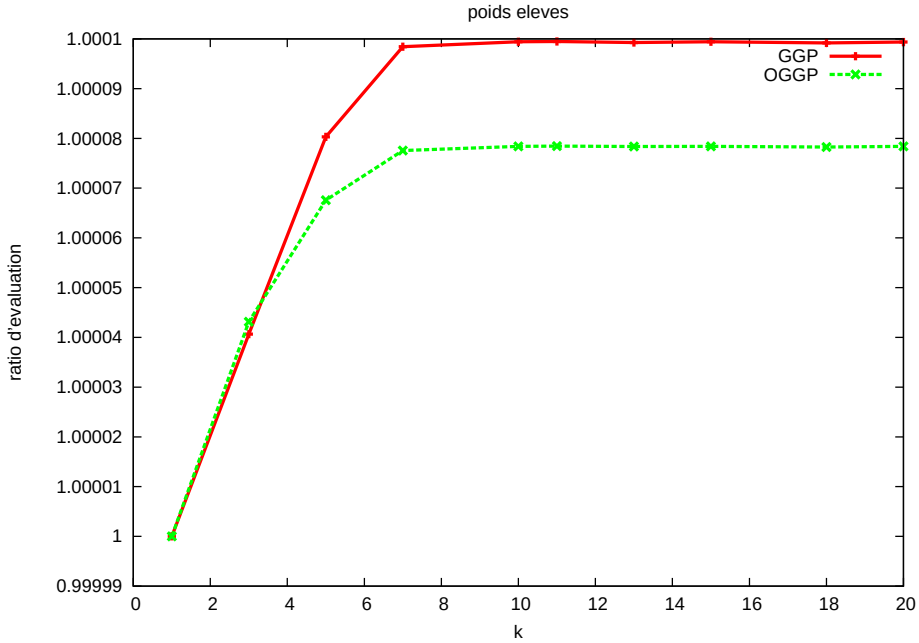
- algorithme d'approximation d'un ratio d'approximation de $\frac{8}{3}$
- ratio de 2 atteignable sur certains exemples
- complexité en $O(\sqrt{n}(n+m)^2)$ où :
  - n : nombre de sommets
  - m : nombre d'arêtes

OGGP

- "Optimized" GGP
- extension de directe GGP
- modification du choix des couplages
- conservation du ratio d'approximation
- amélioration du traitement des exemples problématiques pour GGP
- complexité en $O(\sqrt{n}(n+m)^3)$

Simulations





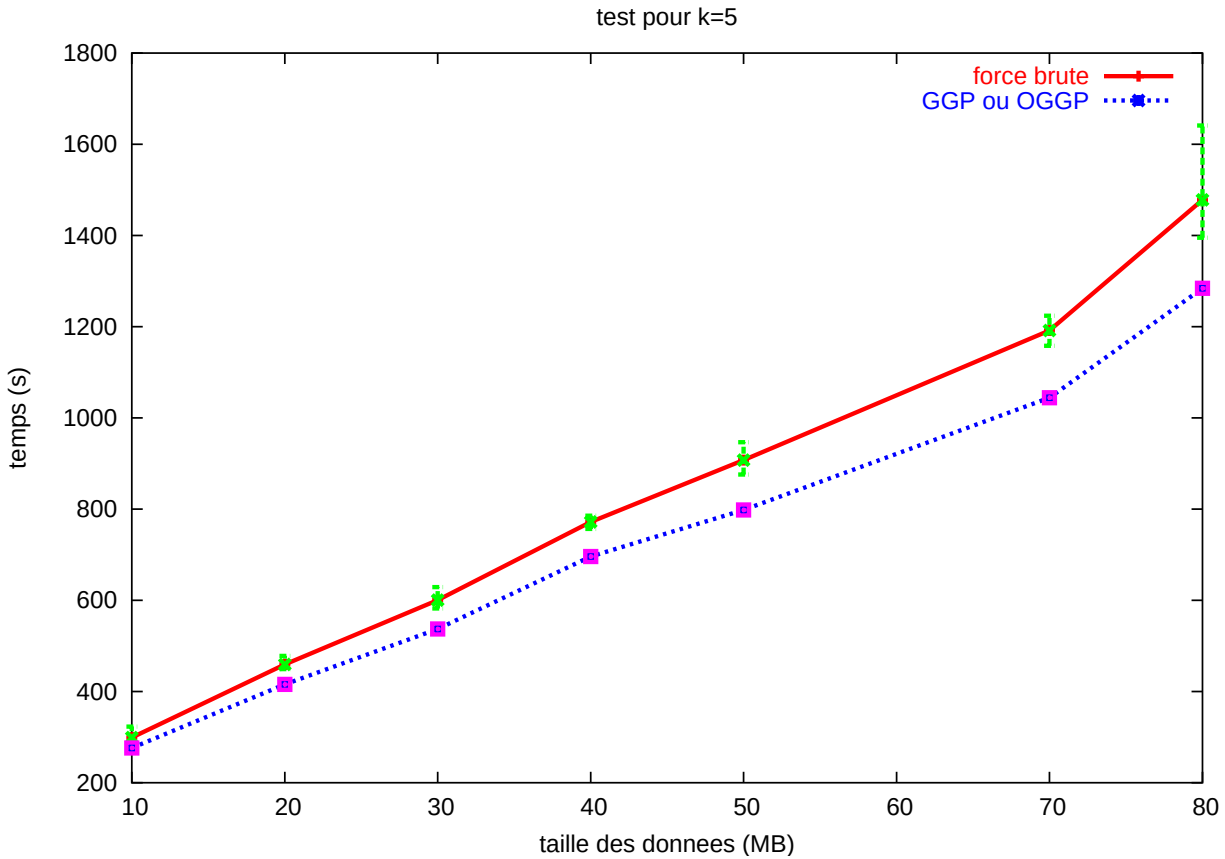
- tests des algorithmes sur des graphes aléatoires
- comparaisons à une borne inférieure sur le temps de communication
- algorithmes quasi-optimaux pour des poids grands devant β
- OGGP meilleur que GGP

Expériences

- tests sur réseau local (2 grappes de 10 machines chacune)
- simulation de différents réseaux à l'aide de qualité de service
- redistributions générées aléatoirement ou choisies
- communications réalisées à l'aide de MPICH / sockets POSIX
- pour chaque redistribution
  - communications ordonnancées (GGP/OGGP)
  - communications simultanées

Expériences

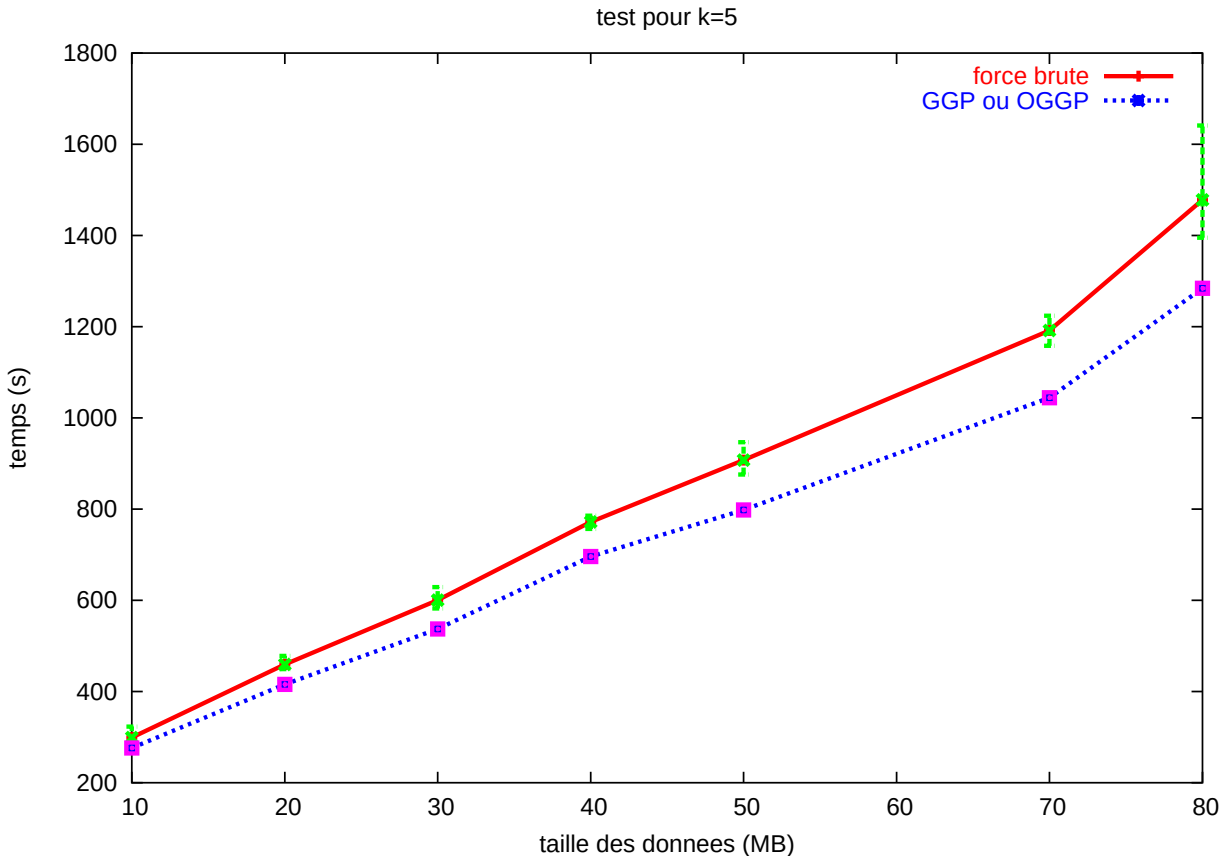
MPI



- entre 5% et 20% d'améliorations
- GGP et OGGP se comportent de manière identique

Expériences

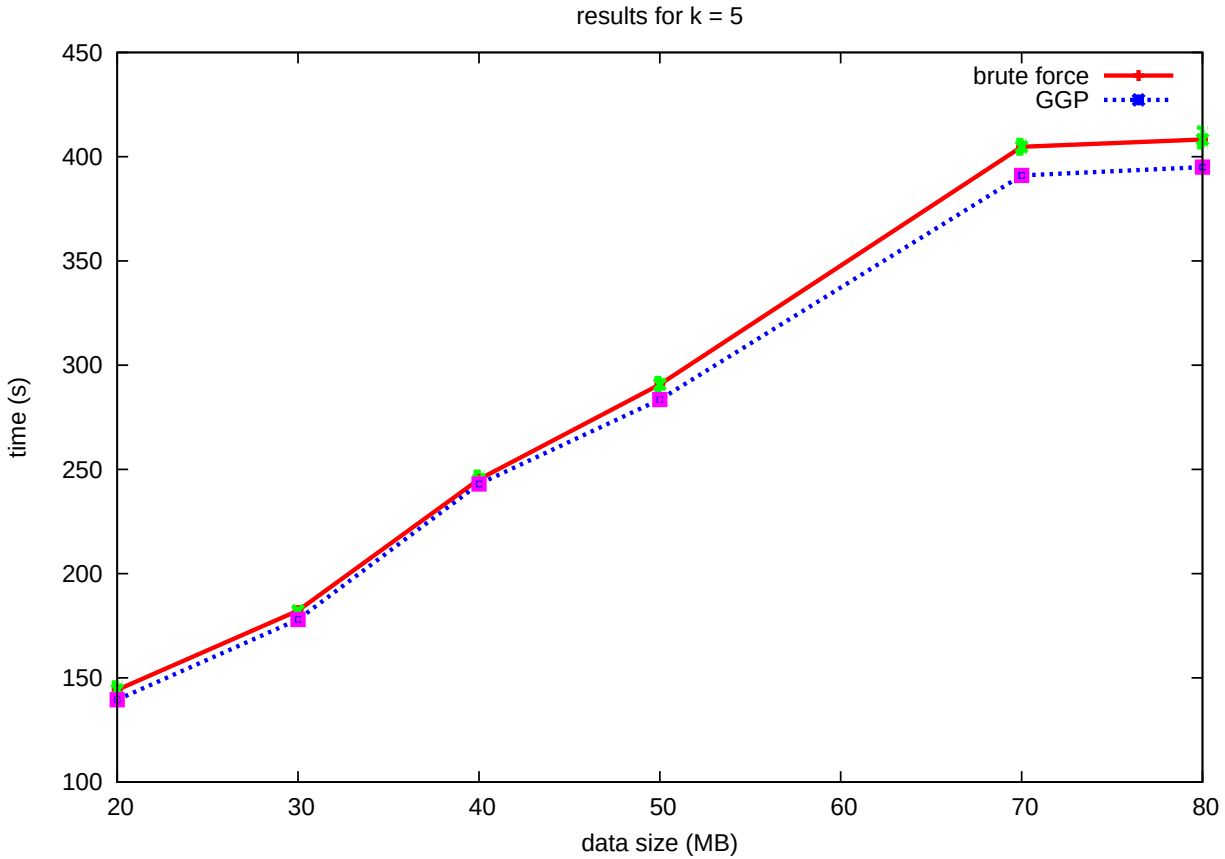
MPI



- entre 5% et 20% d'améliorations
- GGP et OGGP se comportent de manière identique
- gain lié à MPICH ?

Expériences

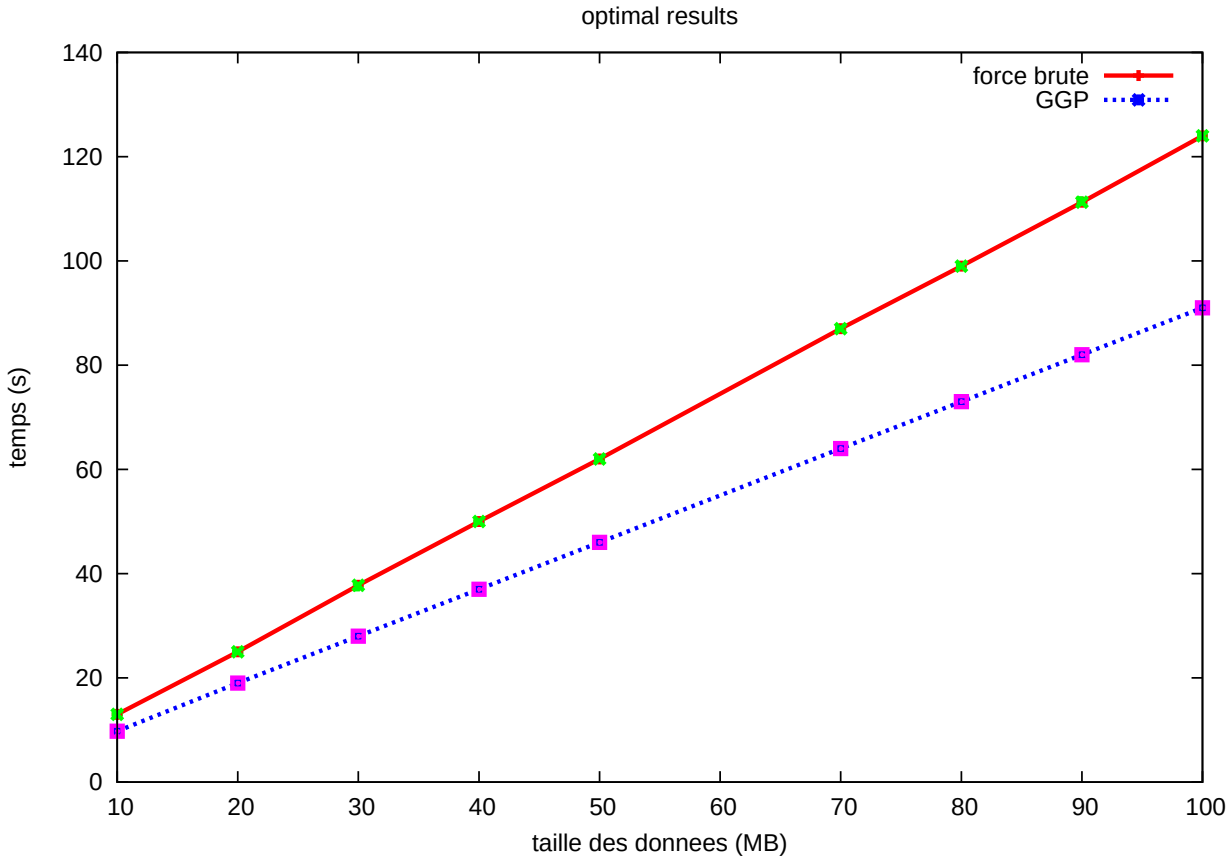
Sockets : redistributions aléatoires



- gains plus faibles compris entre 0 et 20%
- gains dépendants du motif de données

Expériences

Sockets : redistributions fixes



- gain régulier, correspondant à la théorie
- gain pouvant aller jusqu'à 33%

Expériences

Conclusion

- peu de pertes sur le réseau local
- gains possibles même sur un réseau performant
- bonnes performances de l'approche par force brute
- gains en temps de transfert dépendant fortement du motif de données à transférer

Plan de l'exposé

1. Redistributions
2. Le problème KPBS
   - Présentation
   - Modélisation
   - Bornes inférieures
3. Algorithme pseudo-polynomial
4. Algorithme polynomial (GGP)
5. Conclusion

Travaux réalisés

- différentes configurations :
  - redistribution simple
  - grappes hétérogènes
  - communications locales
- dans chaque cas :
  - algorithmes approchés
  - benchmarks (simulations, expériences, ...)
  - prévision de performance
- programmation :
  - programmation d'une bibliothèque de calcul d'ordonnancements pour le problème KPBS
  - début d'intégration à PACO++ (collaboration avec Christian Perez, Irisa)