Présentation autour d'un problème d'ordonnancement avec tâches-couplées

G. SIMONIN R. GIROUDEAU J.C. KÖNIG
LIRMM - UM2 - CNRS
Montpellier France

3 avril 2008

MAO - Lyon

Présentation autour d'un problème d'ordonnancement avec tâches-couplées

G. SIMONIN R. GIROUDEAU J.C. KÖNIG
LIRMM - UM2 - CNRS
Montpellier France

3 avril 2008

MAO - Lyon

Table des matières

Table des matières

Introduction - présentation du problème

La torpille TAIPAN

Introduction - présentation du problème

La torpille TAIPAN

Vérification des pipelines

Mesure de la salinité et température de l'eau

Détecter et analyser les sources d'eau douce

Images acoustiques de l'eau

Mesure des vitesses des écoulements sous-marin

La torpille possède :

- un monoprocesseur
- des capteurs
- deux ensembles de tâches : acquisition et traitement
- des contraintes de précédence entre ses tâches
- des contraintes de compatibilité entre des acquisitions simultanées

La torpille possède :

- un monoprocesseur
- des capteurs
- deux ensembles de tâches : acquisition et traitement
- des contraintes de précédence entre ses tâches
- des contraintes de compatibilité entre des acquisitions simultanées

Double problème pour nous : Décision et Optimisation

La torpille possède :

- un monoprocesseur
- des capteurs
- deux ensembles de tâches : acquisition et traitement
- des contraintes de précédence entre ses tâches
- des contraintes de compatibilité entre des acquisitions simultanées

Double problème pour nous : Décision et Optimisation

La torpille possède :

- un monoprocesseur
- des capteurs
- deux ensembles de tâches : acquisition et traitement
- des contraintes de précédence entre ses tâches
- des contraintes de compatibilité entre des acquisitions simultanées

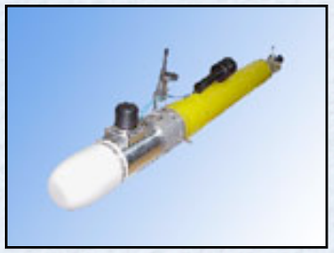
Double problème pour nous : Décision et Optimisation

Introduction - Modélisation

Définition des tâches d'acquisition

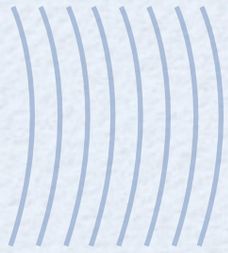
Introduction - Modélisation

Définition des tâches d'acquisition



Introduction - Modélisation

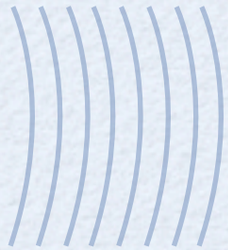
Définition des tâches d'acquisition



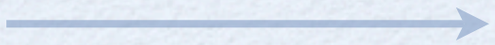
Le capteur envoie

Introduction - Modélisation

Définition des tâches d'acquisition



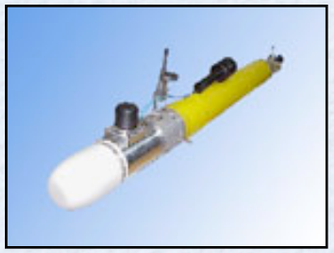
Propagation de l'onde

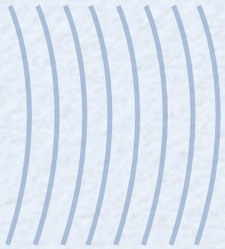


Le capteur envoie

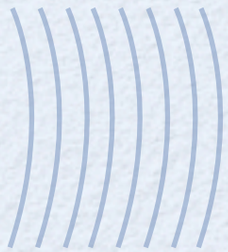
Introduction - Modélisation

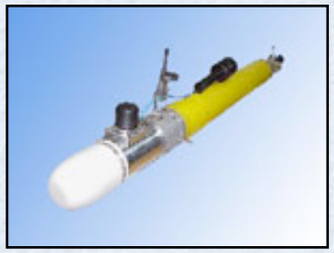
Définition des tâches d'acquisition





Propagation de l'onde





Le capteur envoie

Le capteur reçoit

Introduction - Modélisation

Définition des tâches d'acquisition

Le capteur envoie

Le capteur reçoit

On modélise :

Première sous-tâche

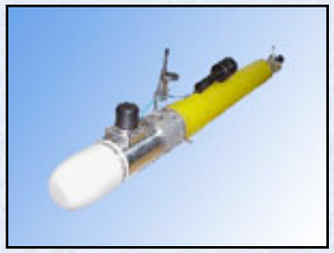
Deuxième sous-tâche

Introduction - Modélisation

Définition des tâches d'acquisition

Introduction - Modélisation

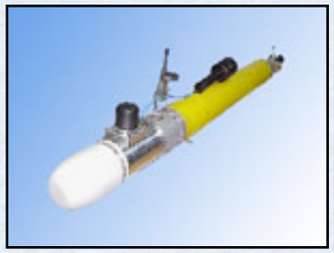
Définition des tâches d'acquisition



Le capteur fait une mesure

Introduction - Modélisation

Définition des tâches d'acquisition

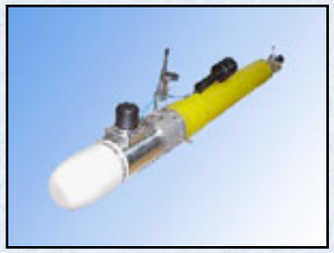


Le capteur fait une mesure

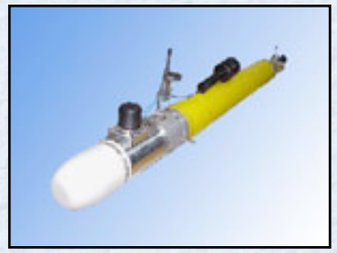
Attente obligatoire

Introduction - Modélisation

Définition des tâches d'acquisition



Attente obligatoire

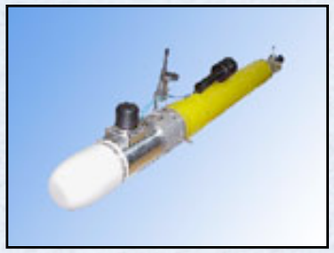


Le capteur fait une mesure

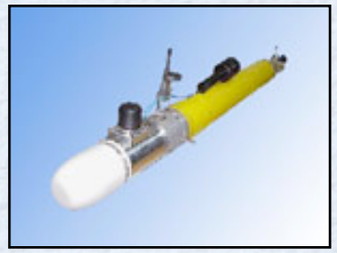
Le capteur fait une autre mesure

Introduction - Modélisation

Définition des tâches d'acquisition



Attente obligatoire



Le capteur fait une mesure

Le capteur fait une autre mesure

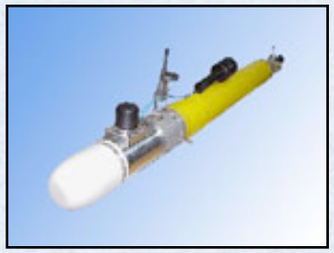
On modélise :

Première sous-tâche

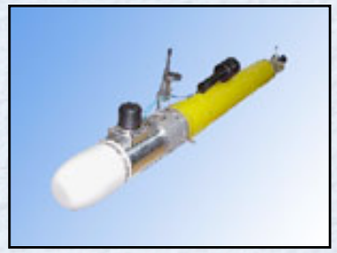
Deuxième sous-tâche

Introduction - Modélisation

Définition des tâches d'acquisition



Attente obligatoire



Le capteur fait une mesure

Le capteur fait une autre mesure

On modélise :

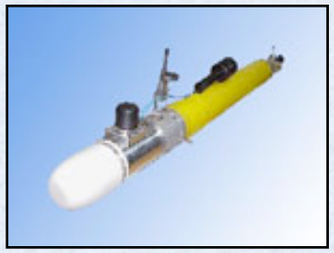
Première sous-tâche

Deuxième sous-tâche

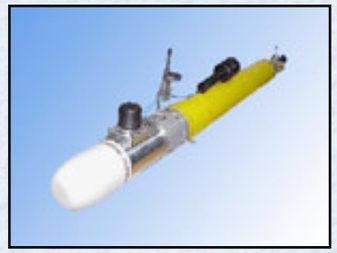
Tâches d'acquisition ⟷ Tâches-couplées

Introduction - Modélisation

Définition des tâches d'acquisition



Attente obligatoire



Le capteur fait une mesure

Le capteur fait une autre mesure

On modélise :

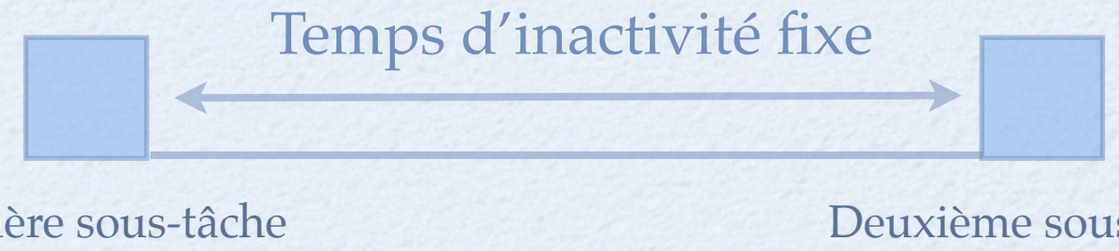
Première sous-tâche

Deuxième sous-tâche

Tâches d'acquisition ⟷ Tâches-couplées

Introduction - Modélisation

On modélise :



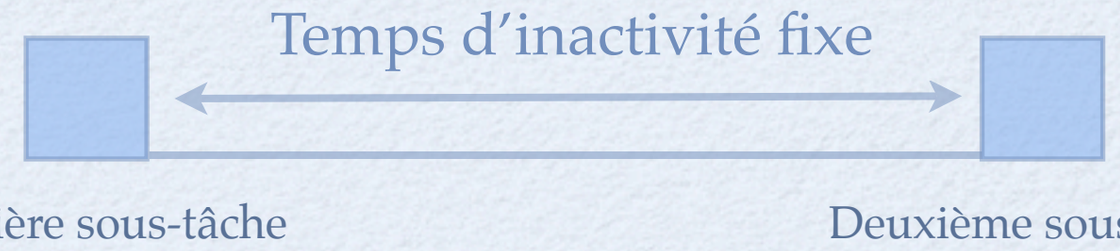
Tâches d'acquisition ⟷ Tâches-couplées

Définition d'une tâche-couplée (Shapiro [1980]) :

- Deux sous-tâches a_i et b_i
- Un délai d'inactivité incompressible et indilatable L_i

Introduction - Modélisation

On modélise :



Tâches d'acquisition ⟷ Tâches-couplées

Définition d'une tâche-couplée (Shapiro [1980]) :

- Deux sous-tâches a_i et b_i
- Un délai d'inactivité incompressible et indilatable L_i

Introduction - Modélisation

Définition d'une tâche-couplée (Shapiro [1980]) :

- Deux sous-tâches a_i et b_i
- Un délai d'inactivité incompressible et indilatable L_i

Introduction - Modélisation

Définition d'une tâche-couplée (Shapiro [1980]) :

- Deux sous-tâches a_i et b_i
- Un délai d'inactivité incompressible et indilatable L_i

Introduction - Modélisation

Définition d'une tâche-couplée (Shapiro [1980]) :

- Deux sous-tâches a_i et b_i
- Un délai d'inactivité incompressible et indilatable L_i

Définition d'une tâche d'acquisition :

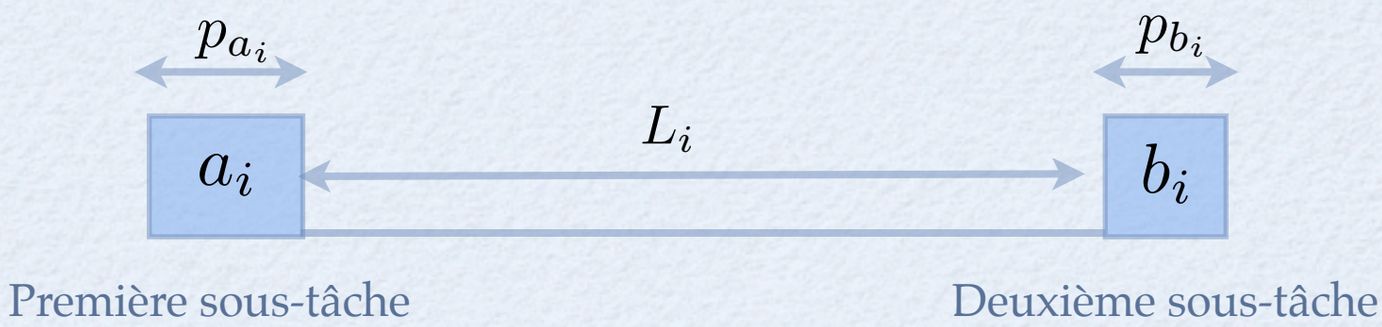
- Deux sous-tâches a_i et b_i de durée d'exécution p_{a_i} et p_{b_i}
- Un délai d'inactivité fixe incompressible et indilatable L_i

Introduction - Modélisation

Définition d'une tâche d'acquisition :

- Deux sous-tâches a_i et b_i de durée d'exécution p_{a_i} et p_{b_i}
- Un délai d'inactivité fixe incompressible et indilatable L_i

Représentation d'une tâche d'acquisition A_i



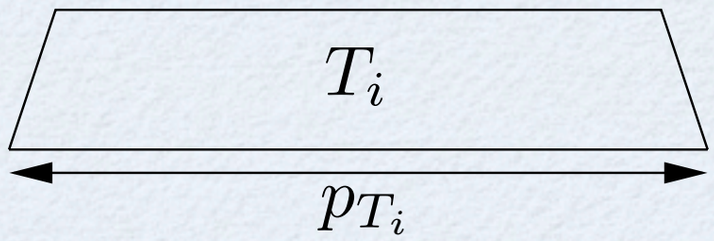
Définition des tâches de traitement :

- Une durée d'exécution notée p_{T_i}
- Elles sont préemptives

Définition des tâches de traitement :

- Une durée d'exécution notée p_{T_i}
- Elles sont préemptives

Représentation d'une tâche de traitement A_i



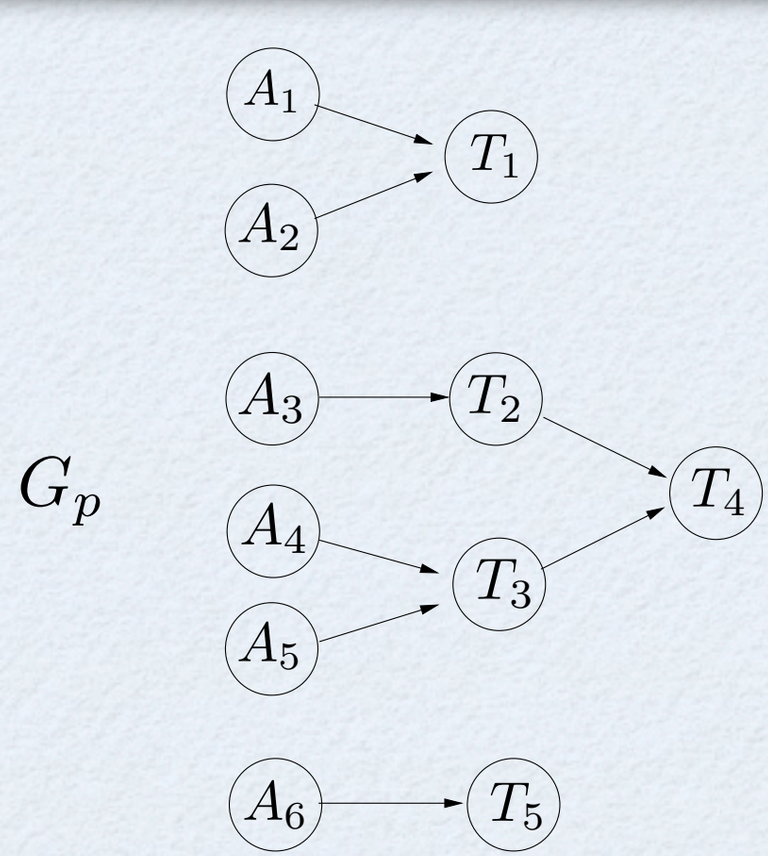
La torpille possède :

- un monoprocesseur
- des capteurs
- deux ensembles de tâches : acquisition et traitement
- des contraintes de précédence entre ses tâches
- des contraintes de compatibilité entre des acquisitions simultanées

Contraintes de précédence :

Soit G_p un graphe non connexe de précédence entre l'ensemble des tâches d'acquisition $\mathcal{A}$ et l'ensemble des tâches de traitement $\mathcal{T}$. Les sources de chaque composante connexe seront toujours les tâches d'acquisition A_i .

G_p

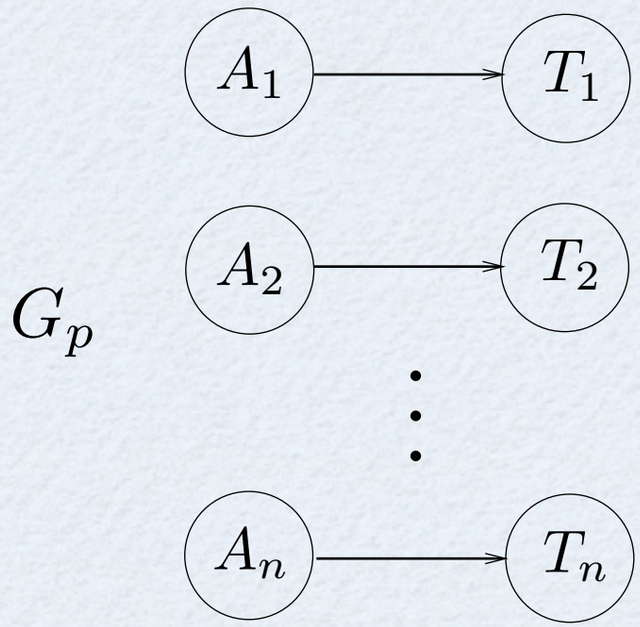


Contraintes de précédence particulière :

Dans un premier temps, le graphe de précédence entre l'ensemble des tâches d'acquisition $\mathcal{A}$ et l'ensemble des tâches de traitement $\mathcal{T}$ sera de telle manière que pour chaque tâche A_i de $\mathcal{A}$, nous avons une tâche T_i de $\mathcal{T}$ qui lui succède.

Contraintes de précédence particulière :

Dans un premier temps, le graphe de précédence entre l'ensemble des tâches d'acquisition $\mathcal{A}$ et l'ensemble des tâches de traitement $\mathcal{T}$ sera de telle manière que pour chaque tâche A_i de $\mathcal{A}$, nous avons une tâche T_i de $\mathcal{T}$ qui lui succède.



La torpille possède :

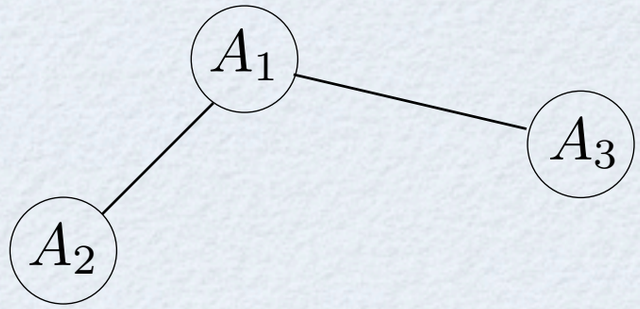
- un monoprocesseur
- des capteurs
- deux ensembles de tâches : acquisition et traitement
- des contraintes de précédence entre ses tâches
- des contraintes de compatibilité entre des acquisitions simultanées

Les tâches d'acquisition sont également soumises à un graphe de compatibilité différent du graphe de précédence

Les tâches d'acquisition sont également soumises à un graphe de compatibilité différent du graphe de précédence

Contrainte de compatibilité :

Soit $G_c = (\mathcal{A}, E)$ un graphe de compatibilité où l'ensemble des tâches d'acquisition $\mathcal{A}$ définit les sommets du graphe et E représente l'ensemble des arêtes reliant deux tâches d'acquisition qui sont non exclusives.

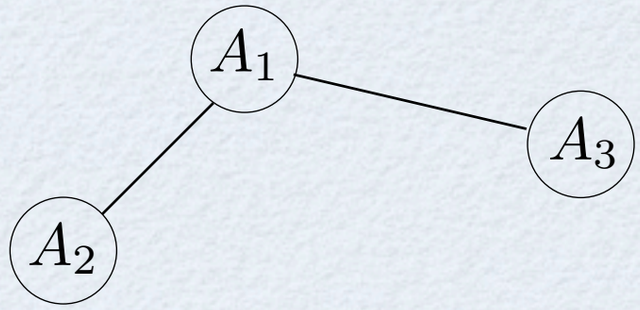


Graphe de compatibilité G_c

Les tâches d'acquisition sont également soumises à un graphe de compatibilité différent du graphe de précédence

Contrainte de compatibilité :

Soit $G_c = (\mathcal{A}, E)$ un graphe de compatibilité où l'ensemble des tâches d'acquisition $\mathcal{A}$ définit les sommets du graphe et E représente l'ensemble des arêtes reliant deux tâches d'acquisition qui sont non exclusives.



Graphe de compatibilité G_c

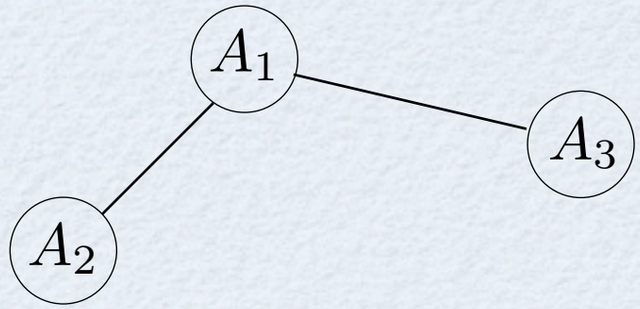
Tâches d'acquisition non exclusives :

Deux tâches d'acquisition A_i et A_j sont dites non exclusives lorsque leur temps d'inactivité respectif L_i et L_j peuvent avoir lieu en même temps.

Les tâches d'acquisition sont également soumises à un graphe de compatibilité différent du graphe de précédence

Contrainte de compatibilité :

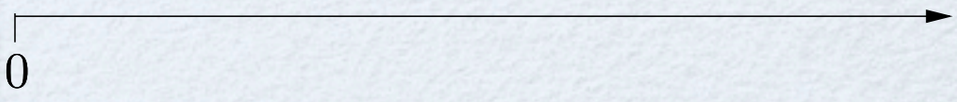
Soit $G_c = (\mathcal{A}, E)$ un graphe de compatibilité où l'ensemble des tâches d'acquisition $\mathcal{A}$ définit les sommets du graphe et E représente l'ensemble des arêtes reliant deux tâches d'acquisition qui sont non exclusives.



Graphe de compatibilité G_c

Tâches d'acquisition non exclusives :

Deux tâches d'acquisition A_i et A_j sont dites non exclusives lorsque leur temps d'inactivité respectif L_i et L_j peuvent avoir lieu en même temps.

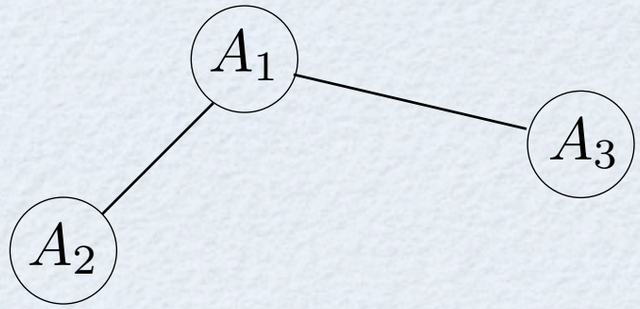


Exemple

Les tâches d'acquisition sont également soumises à un graphe de compatibilité différent du graphe de précédence

Contrainte de compatibilité :

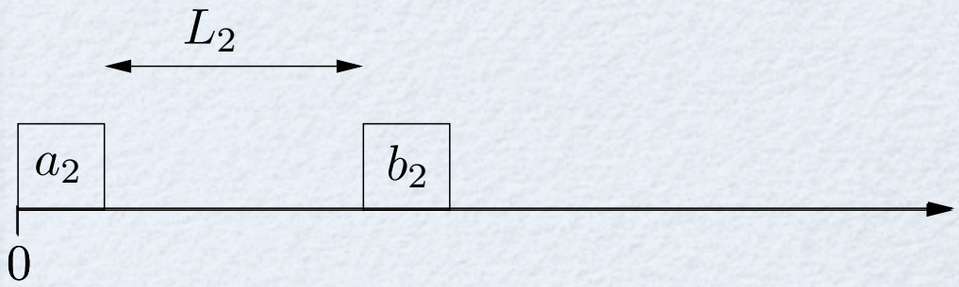
Soit $G_c = (\mathcal{A}, E)$ un graphe de compatibilité où l'ensemble des tâches d'acquisition $\mathcal{A}$ définit les sommets du graphe et E représente l'ensemble des arêtes reliant deux tâches d'acquisition qui sont non exclusives.



Graphe de compatibilité G_c

Tâches d'acquisition non exclusives :

Deux tâches d'acquisition A_i et A_j sont dites non exclusives lorsque leur temps d'inactivité respectif L_i et L_j peuvent avoir lieu en même temps.

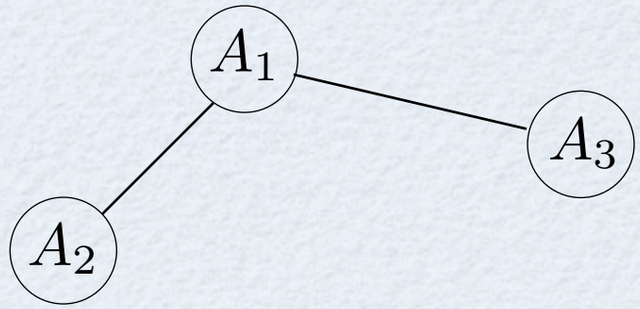


Exemple

Les tâches d'acquisition sont également soumises à un graphe de compatibilité différent du graphe de précédence

Contrainte de compatibilité :

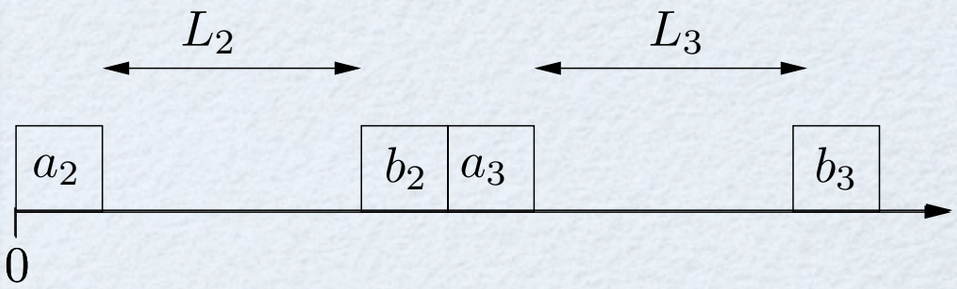
Soit $G_c = (\mathcal{A}, E)$ un graphe de compatibilité où l'ensemble des tâches d'acquisition $\mathcal{A}$ définit les sommets du graphe et E représente l'ensemble des arêtes reliant deux tâches d'acquisition qui sont non exclusives.



Graphe de compatibilité G_c

Tâches d'acquisition non exclusives :

Deux tâches d'acquisition A_i et A_j sont dites non exclusives lorsque leur temps d'inactivité respectif L_i et L_j peuvent avoir lieu en même temps.

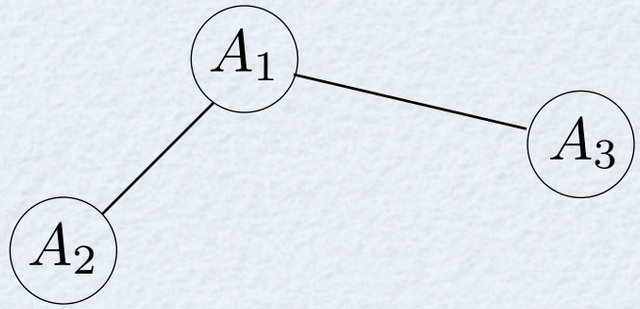


Exemple

Les tâches d'acquisition sont également soumises à un graphe de compatibilité différent du graphe de précédence

Contrainte de compatibilité :

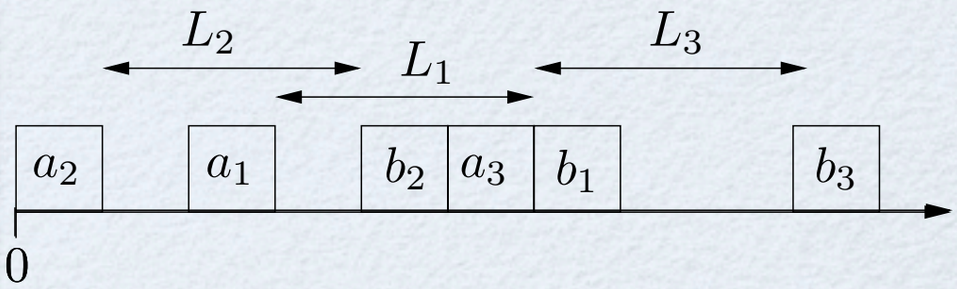
Soit $G_c = (\mathcal{A}, E)$ un graphe de compatibilité où l'ensemble des tâches d'acquisition $\mathcal{A}$ définit les sommets du graphe et E représente l'ensemble des arêtes reliant deux tâches d'acquisition qui sont non exclusives.



Graphe de compatibilité G_c

Tâches d'acquisition non exclusives :

Deux tâches d'acquisition A_i et A_j sont dites non exclusives lorsque leur temps d'inactivité respectif L_i et L_j peuvent avoir lieu en même temps.



Exemple

Notre problème peut être représenté par la notation de Graham [1979] : $\alpha|\beta|\gamma$

$$1|\ prec,\ tâche-couplée,\ (a_i, L_i, b_i) \cup (p_{T_i}, pmtn), G_c|C_{max}$$

Notre problème peut être représenté par la notation de Graham [1979] : $\alpha|\beta|\gamma$

$$1|\ prec,\ tâche-couplée,\ (a_i, L_i, b_i) \cup (p_{T_i}, pmtn), G_c|C_{max}$$

- 1 : monoprocesseur
- $prec$: contrainte de précédence
- $tâche-couplée$: modèle utilisé
- (a_i, L_i, b_i) : tâches d'aquisition
- $(p_{T_i}, pmtn)$: tâches de traitement
- G_c : graphe de compatibilité
- C_{max} : fonction objective, l'ordonnancement le plus court

Table des matières

- Introduction
  - Présentation du problème
  - Modélisation
- Complexité de ces problèmes
  - Etat de l'art
  - Preuve de NP-complétude sur un cas particulier
- Les techniques d'approximation
  - Couplage maximum
  - Clique maximale
  - Poupées russes
- Les techniques de non approximation
  - Clique partition
  - Triangle packing
- Conclusion et perspectives

Etat de l'art d'un point de vue de la complexité

Etat de l'art d'un point de vue de la complexité

- Ordonnancement de tâches-couplées sur monoprocesseur :

Etat de l'art d'un point de vue de la complexité

- Ordonnancement de tâches-couplées sur monoprocesseur :
  - Problèmes de complexité :
    - J. Blazewicz, K.H. Ecker, T. Kis, and M. Tanas. A note on the complexity of scheduling coupled-tasks on a single processor. *Journal of the Brazilian Computer Society*, 7(3) : 23–26, 2001.
    - A.J. Orman and C.N. Potts. On the complexity of coupled-task scheduling. *Discrete Applied Mathematics*, 72 :141–154, 1997.

Etat de l'art d'un point de vue de la complexité

- Ordonnancement de tâches-couplées sur monoprocesseur :
  - Problèmes de complexité :
    - J. Blazewicz, K.H. Ecker, T. Kis, and M. Tanas. A note on the complexity of scheduling coupled-tasks on a single processor. *Journal of the Brazilian Computer Society*, 7(3) : 23–26, 2001.
    - A.J. Orman and C.N. Potts. On the complexity of coupled-task scheduling. *Discrete Applied Mathematics*, 72 :141–154, 1997.

Etat de l'art d'un point de vue de la complexité

- Ordonnancement de tâches-couplées sur monoprocesseur :
  - A.J. Orman and C.N. Potts. On the complexity of coupled-task scheduling. *Discrete Applied Mathematics*, 72 :141–154, 1997.

Etat de l'art d'un point de vue de la complexité

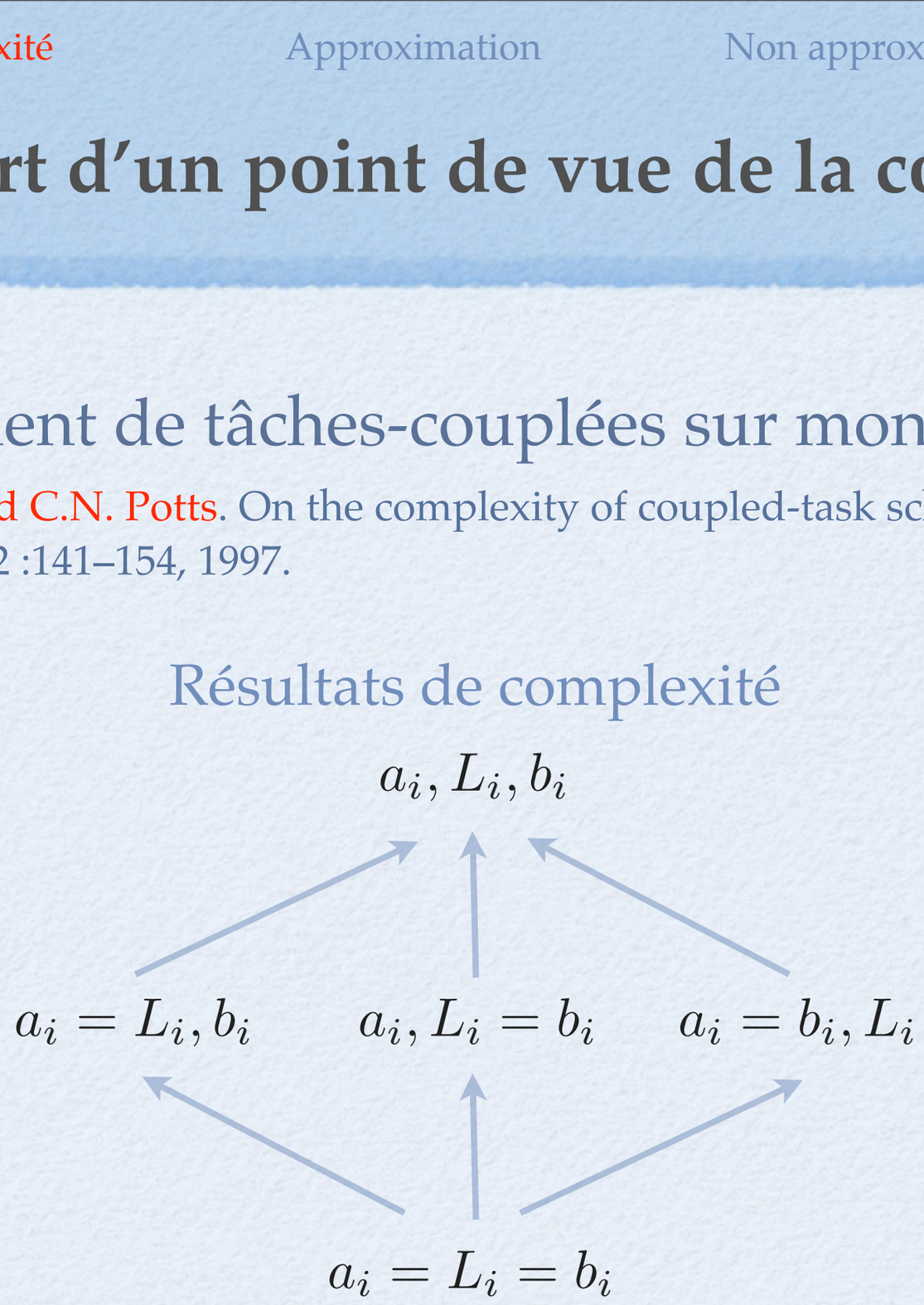
- Ordonnancement de tâches-couplées sur monoprocesseur :
  - A.J. Orman and C.N. Potts. On the complexity of coupled-task scheduling. *Discrete Applied Mathematics*, 72 :141–154, 1997.

Résultats de complexité

Etat de l'art d'un point de vue de la complexité

- Ordonnancement de tâches-couplées sur monoprocesseur :
  - A.J. Orman and C.N. Potts. On the complexity of coupled-task scheduling. *Discrete Applied Mathematics*, 72 :141–154, 1997.

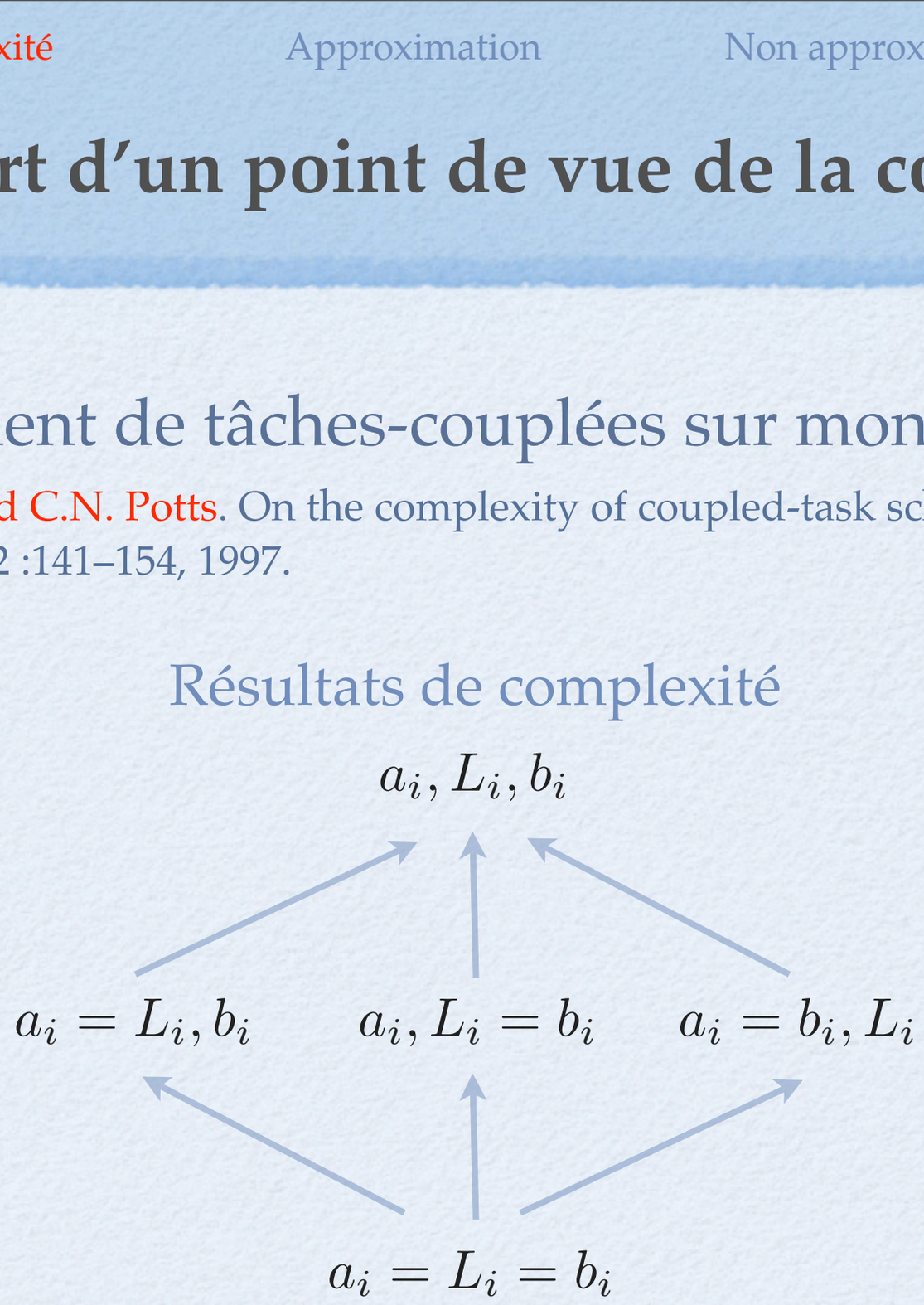
Résultats de complexité



Etat de l'art d'un point de vue de la complexité

- Ordonnancement de tâches-couplées sur monoprocesseur :
  - A.J. Orman and C.N. Potts. On the complexity of coupled-task scheduling. *Discrete Applied Mathematics*, 72 :141–154, 1997.

Résultats de complexité

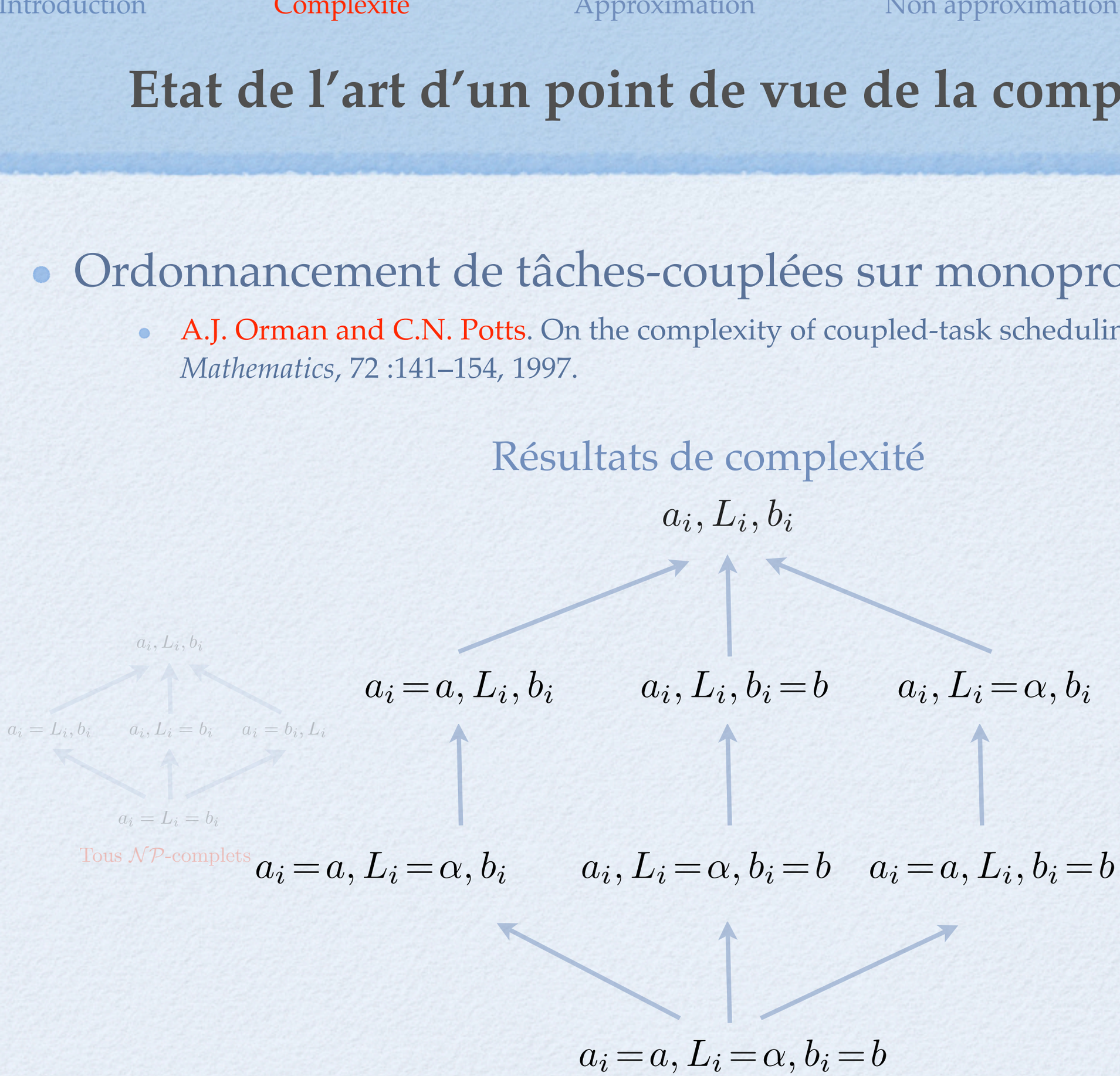


Tous $\mathcal{NP}$-complets

Etat de l'art d'un point de vue de la complexité

- Ordonnancement de tâches-couplées sur monoprocesseur :
  - A.J. Orman and C.N. Potts. On the complexity of coupled-task scheduling. *Discrete Applied Mathematics*, 72 :141–154, 1997.

Résultats de complexité

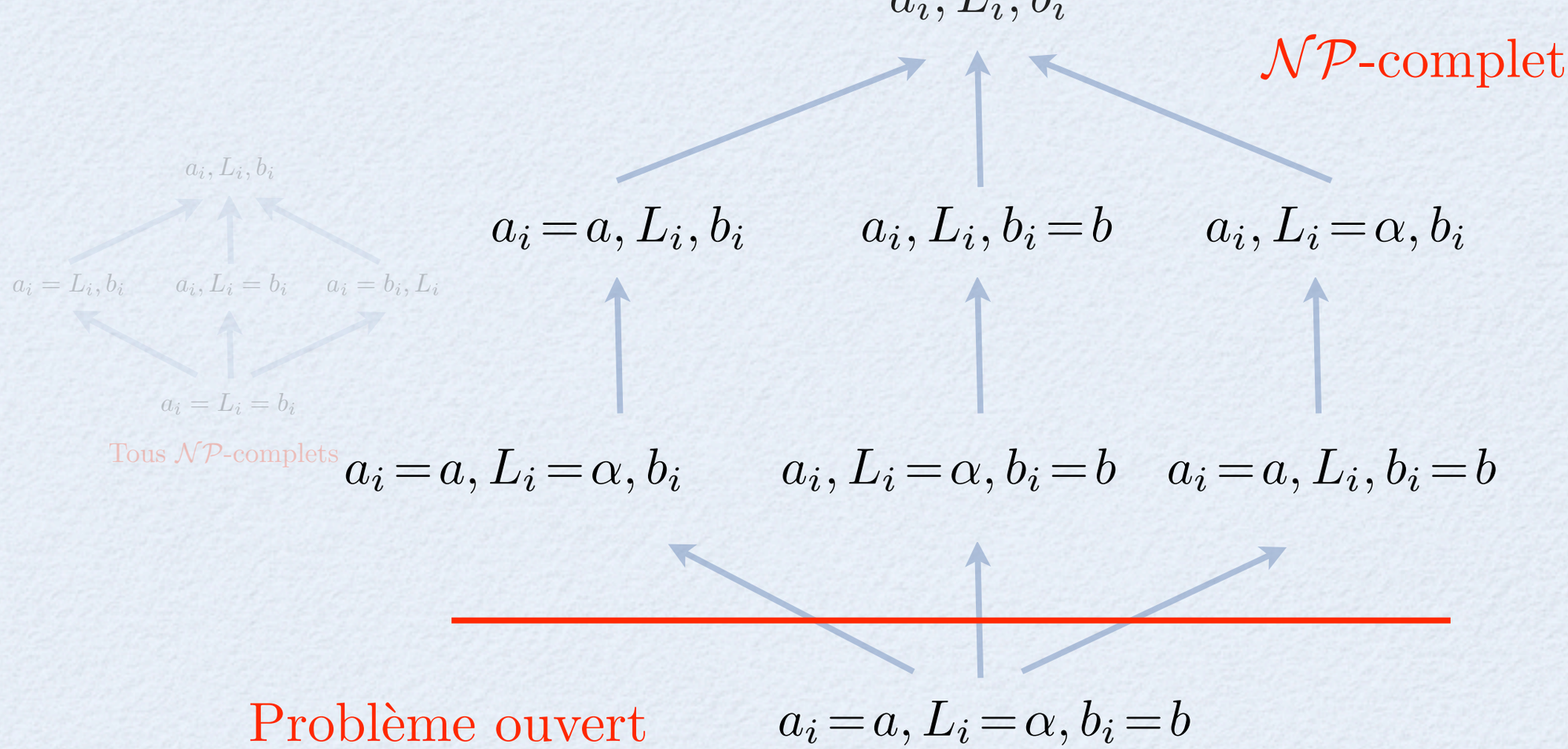


Tous $\mathcal{NP}$-complets

Etat de l'art d'un point de vue de la complexité

- Ordonnancement de tâches-couplées sur monoprocesseur :
  - A.J. Orman and C.N. Potts. On the complexity of coupled-task scheduling. *Discrete Applied Mathematics*, 72 :141–154, 1997.

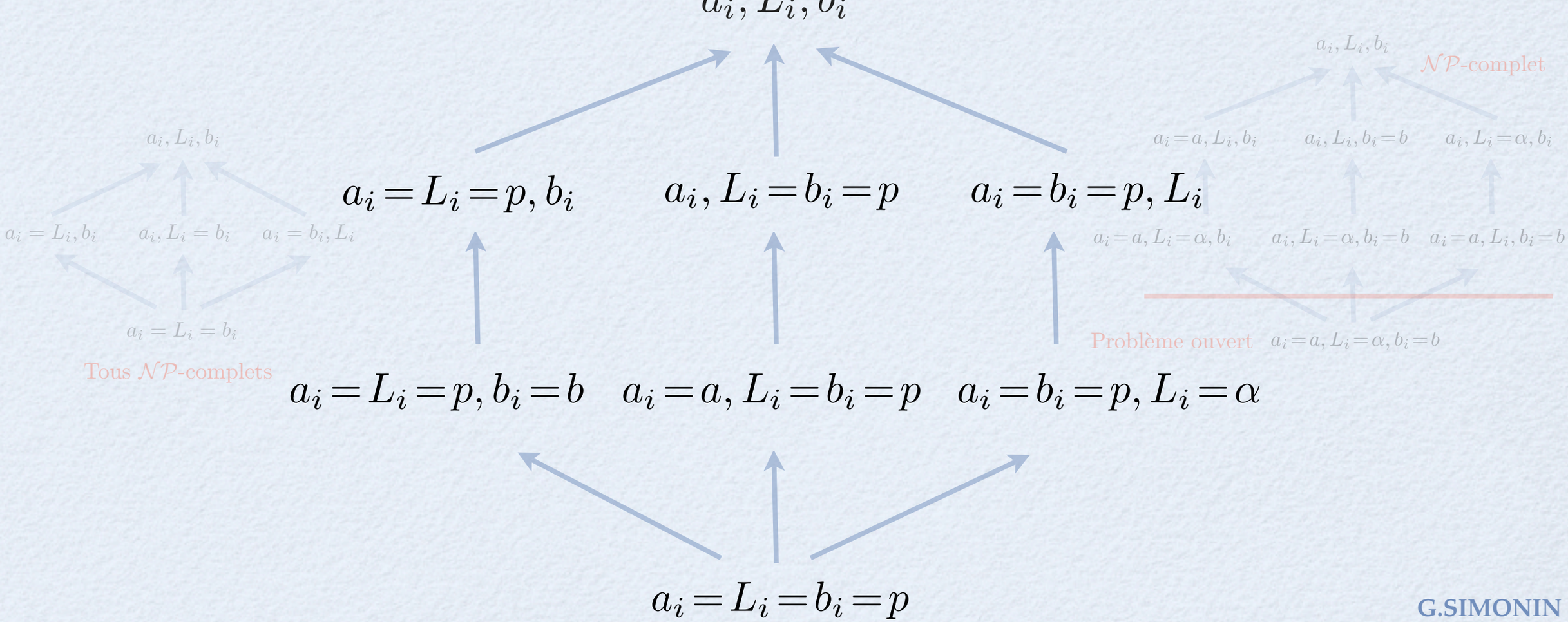
Résultats de complexité



Etat de l'art d'un point de vue de la complexité

- Ordonnancement de tâches-couplées sur monoprocesseur :
  - A.J. Orman and C.N. Potts. On the complexity of coupled-task scheduling. *Discrete Applied Mathematics*, 72 :141–154, 1997.

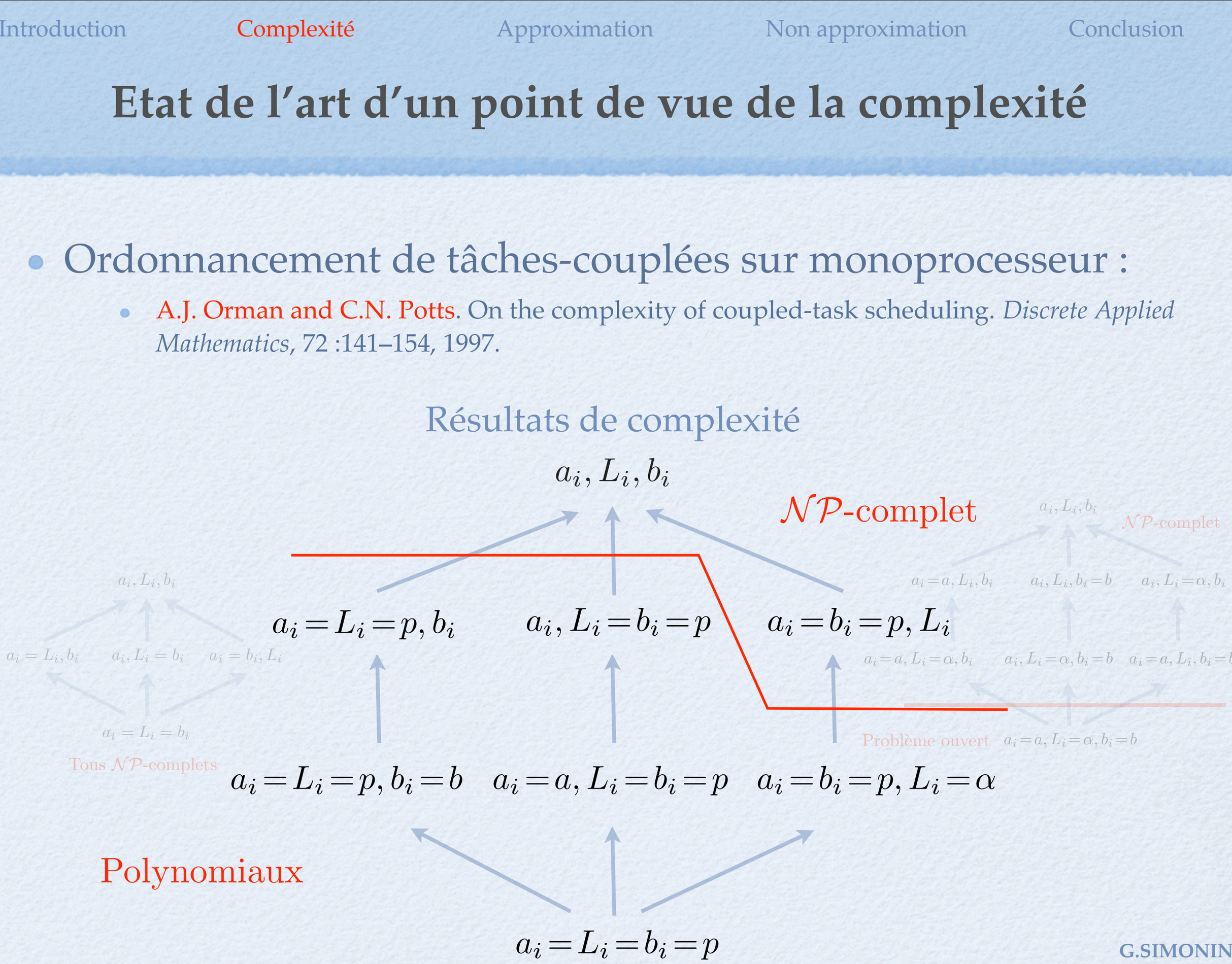
Résultats de complexité



Etat de l'art d'un point de vue de la complexité

- Ordonnancement de tâches-couplées sur monoprocesseur :
  - A.J. Orman and C.N. Potts. On the complexity of coupled-task scheduling. *Discrete Applied Mathematics*, 72 :141–154, 1997.

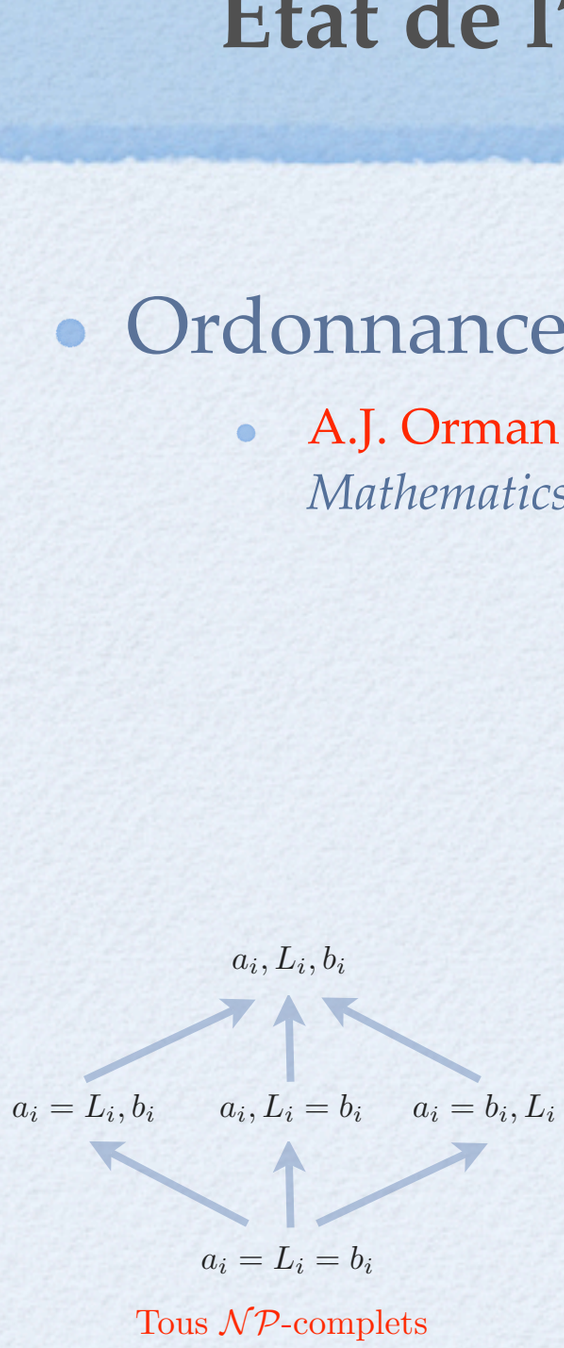
Résultats de complexité



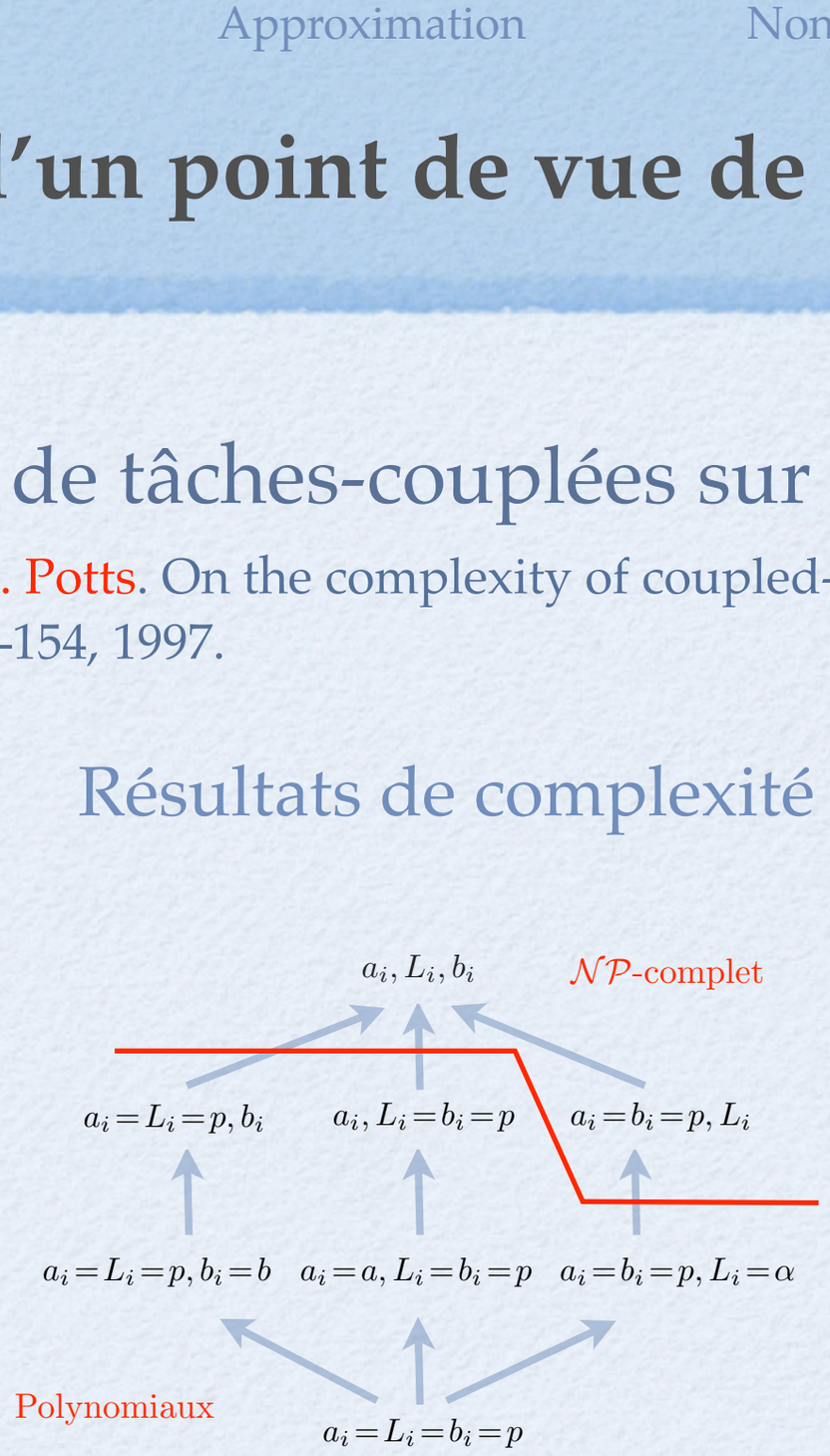
Etat de l'art d'un point de vue de la complexité

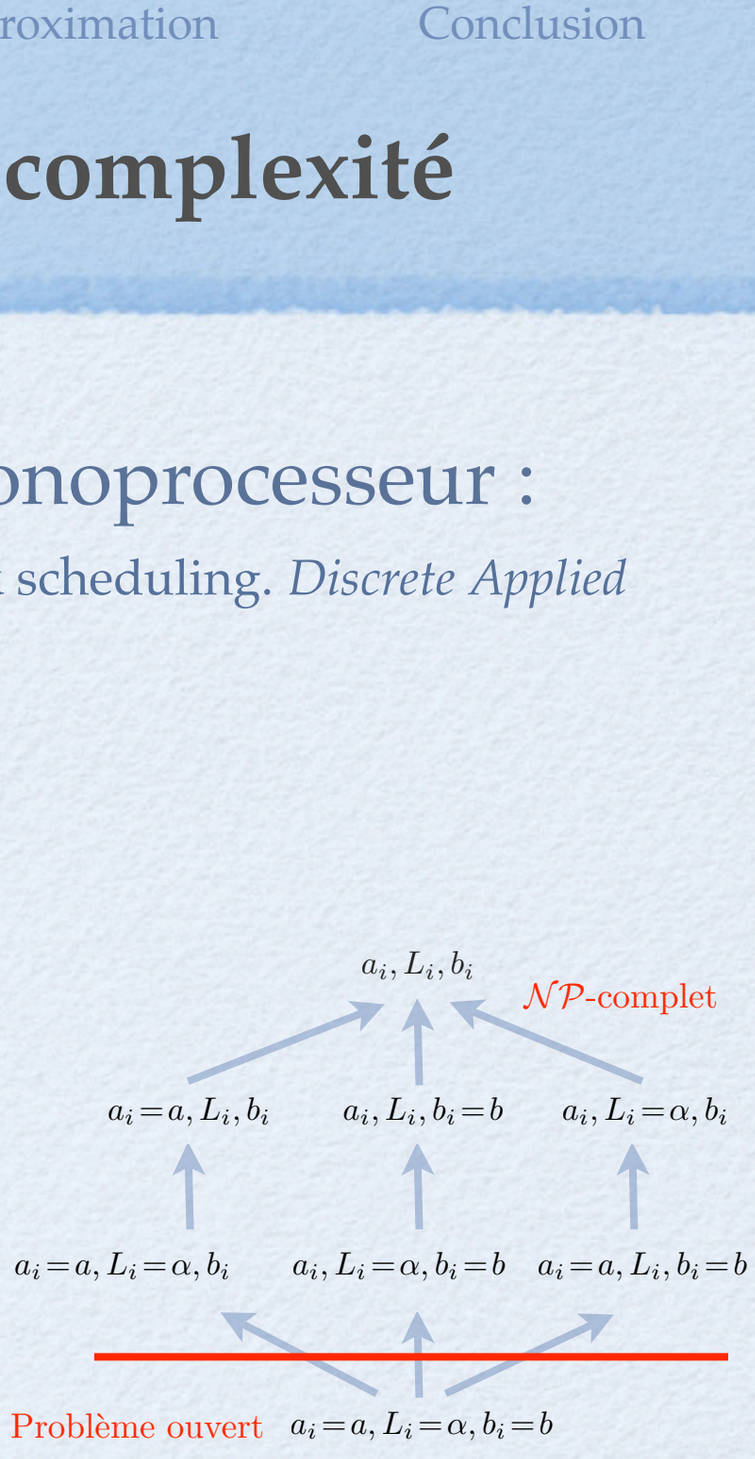
- Ordonnancement de tâches-couplées sur monoprocesseur :
  - A.J. Orman and C.N. Potts. On the complexity of coupled-task scheduling. *Discrete Applied Mathematics*, 72 :141–154, 1997.

Résultats de complexité



Tous $\mathcal{NP}$-complets

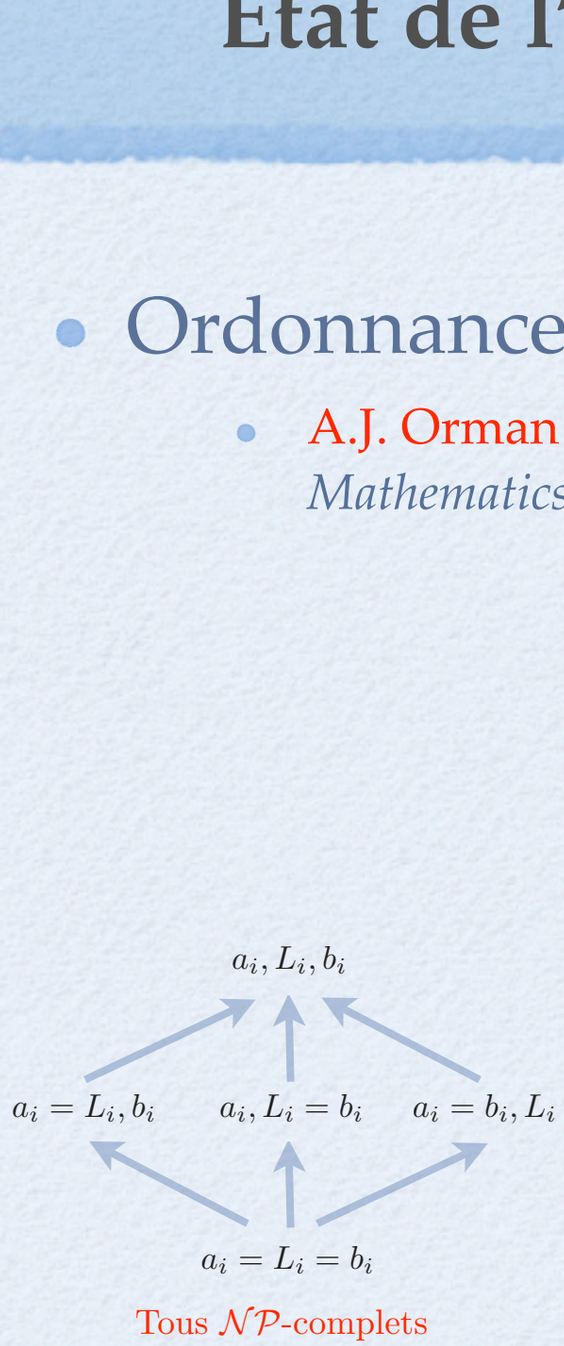




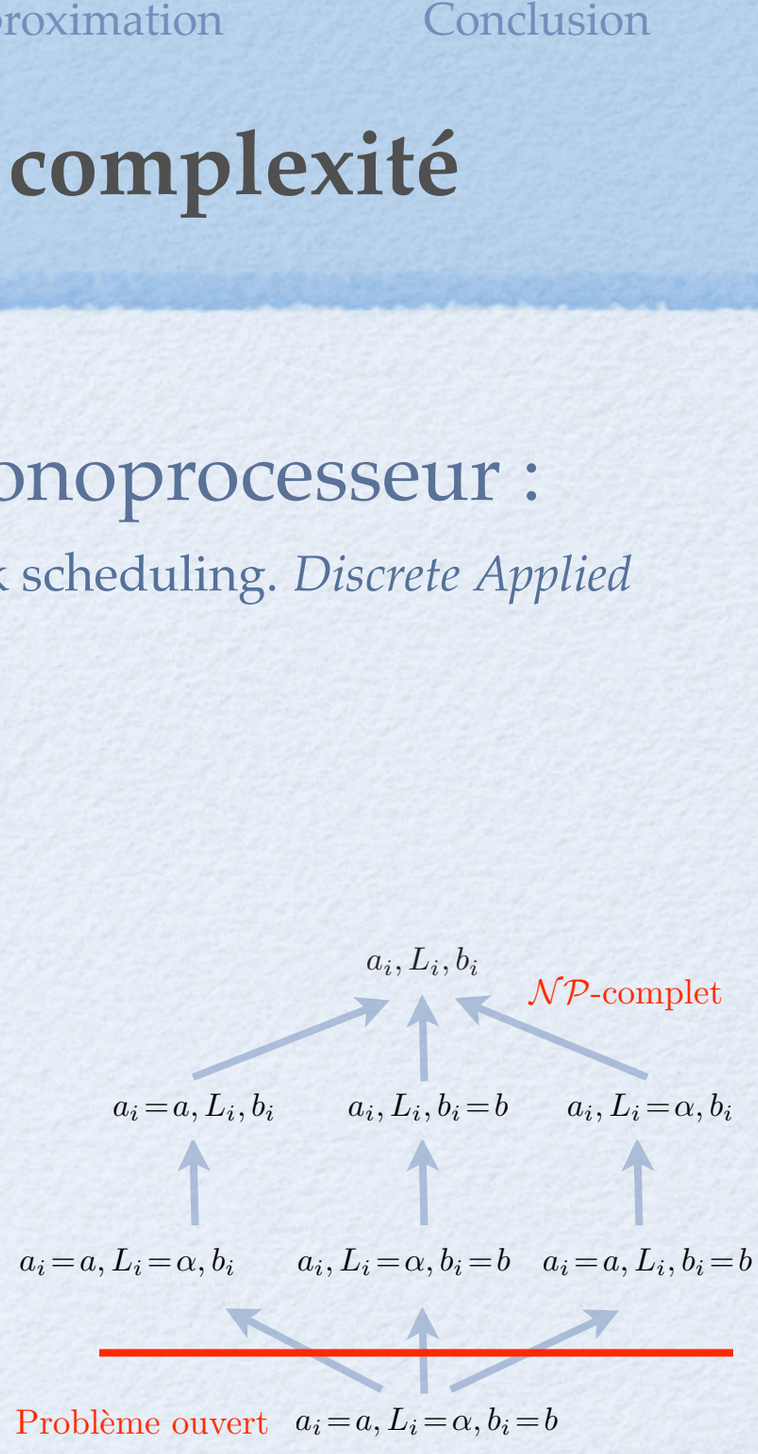
Etat de l'art d'un point de vue de la complexité

- Ordonnancement de tâches-couplées sur monoprocesseur :
  - A.J. Orman and C.N. Potts. On the complexity of coupled-task scheduling. *Discrete Applied Mathematics*, 72 :141–154, 1997.

Résultats de complexité



Tous $\mathcal{NP}$-complets

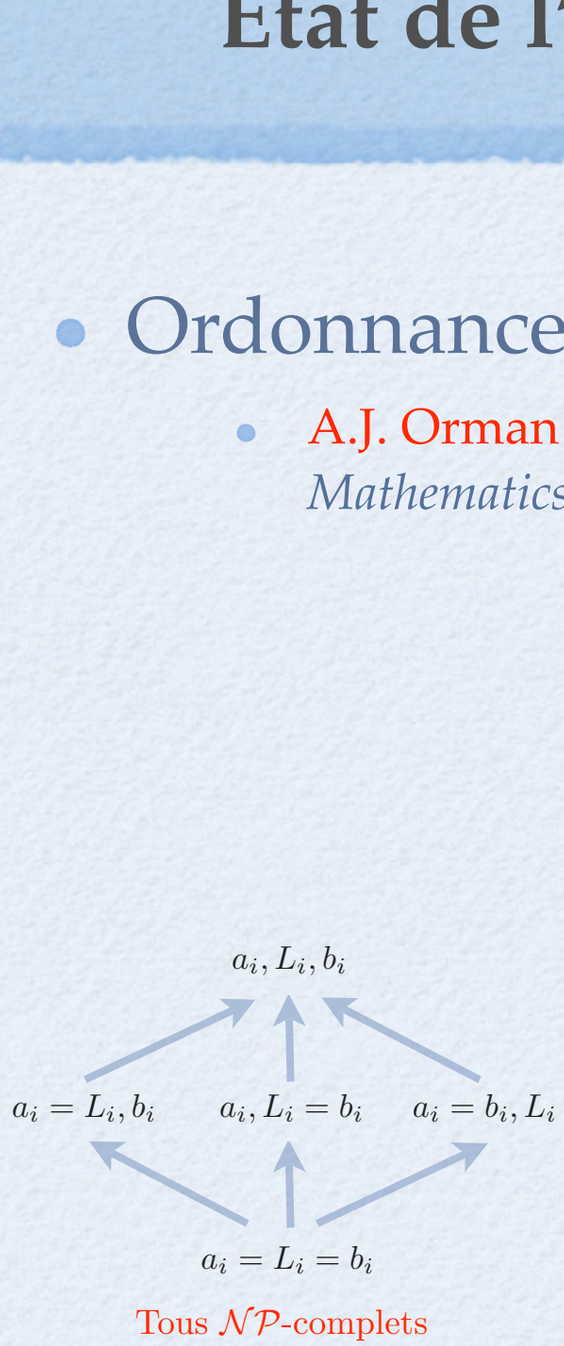


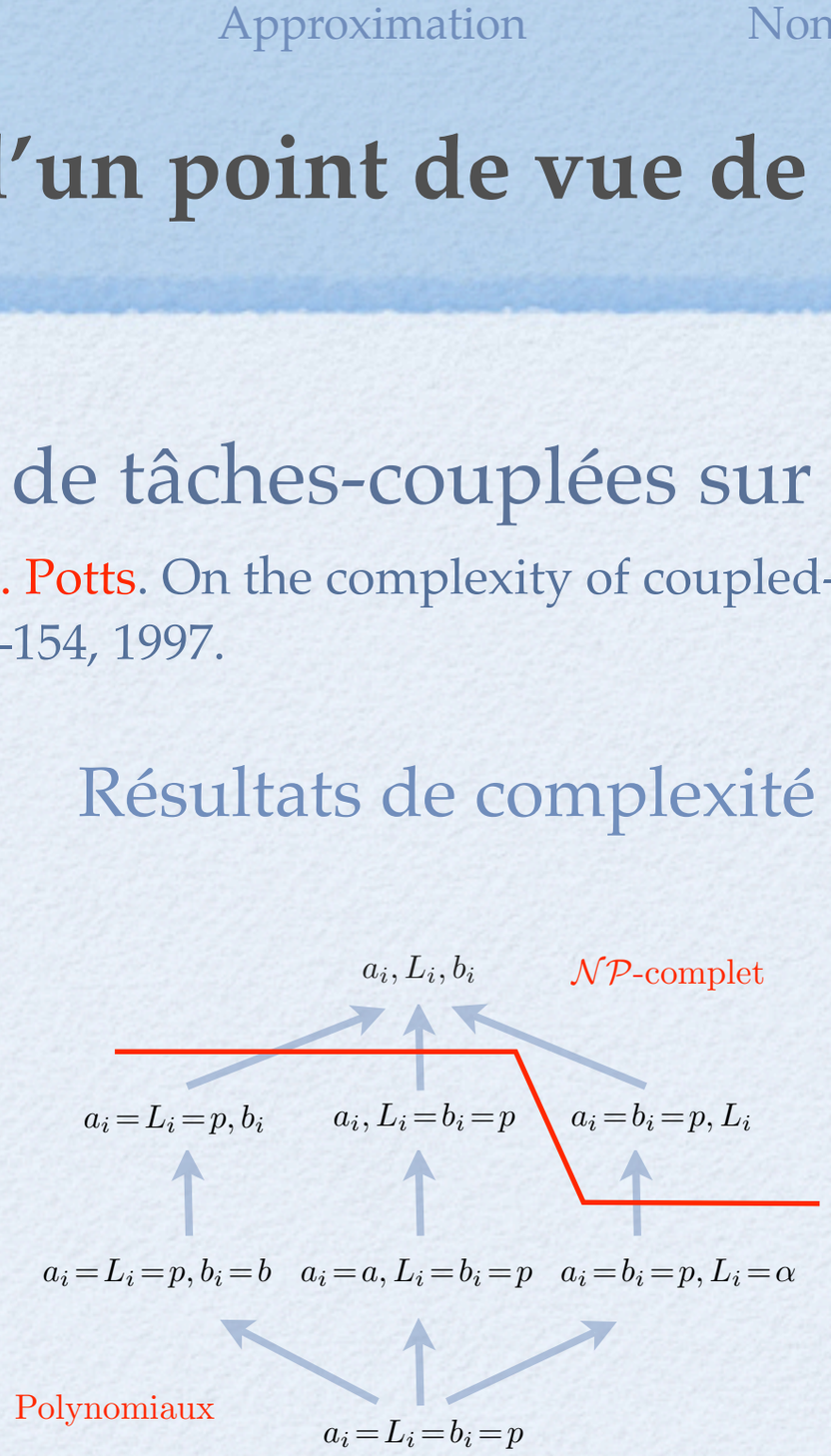
Avec les T_i et G_p : rien ne change

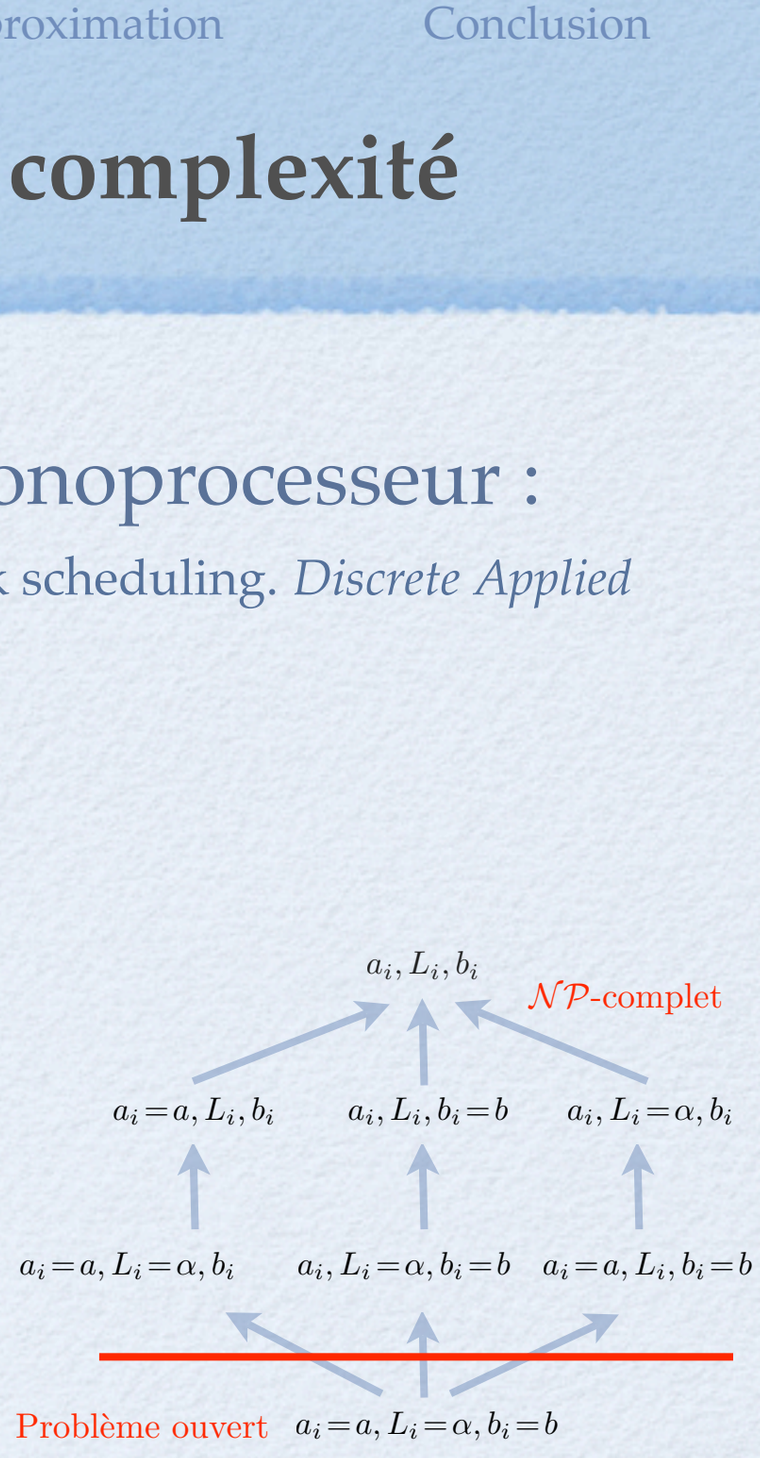
Etat de l'art d'un point de vue de la complexité

- Ordonnancement de tâches-couplées sur monoprocesseur :
  - A.J. Orman and C.N. Potts. On the complexity of coupled-task scheduling. *Discrete Applied Mathematics*, 72 :141–154, 1997.

Résultats de complexité







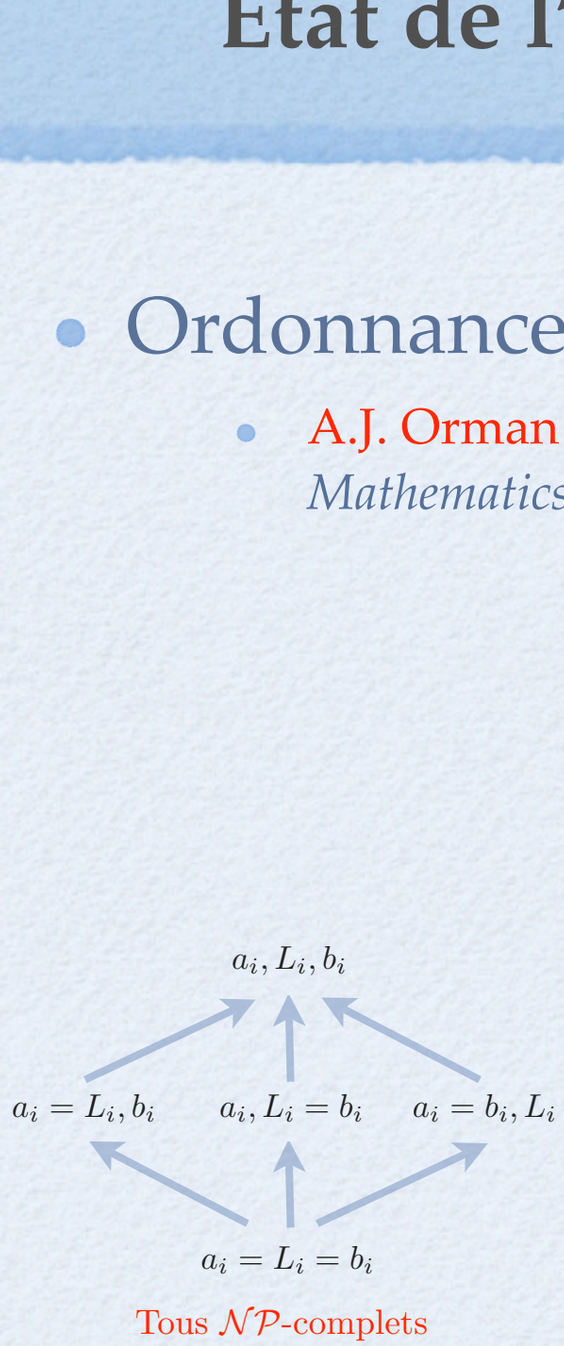
Avec les T_i et G_p : rien ne change

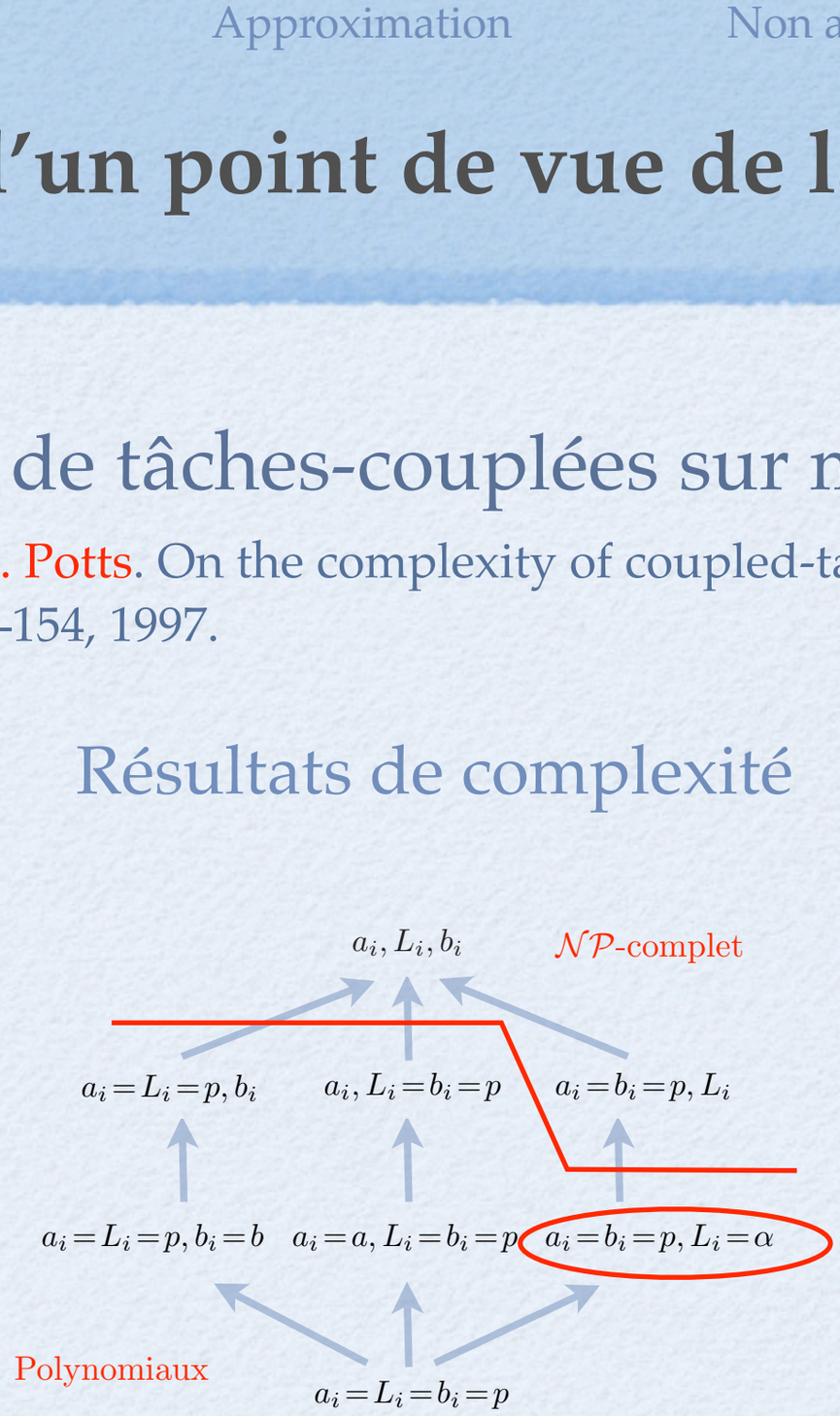
Qu'en est-il avec G_c ?

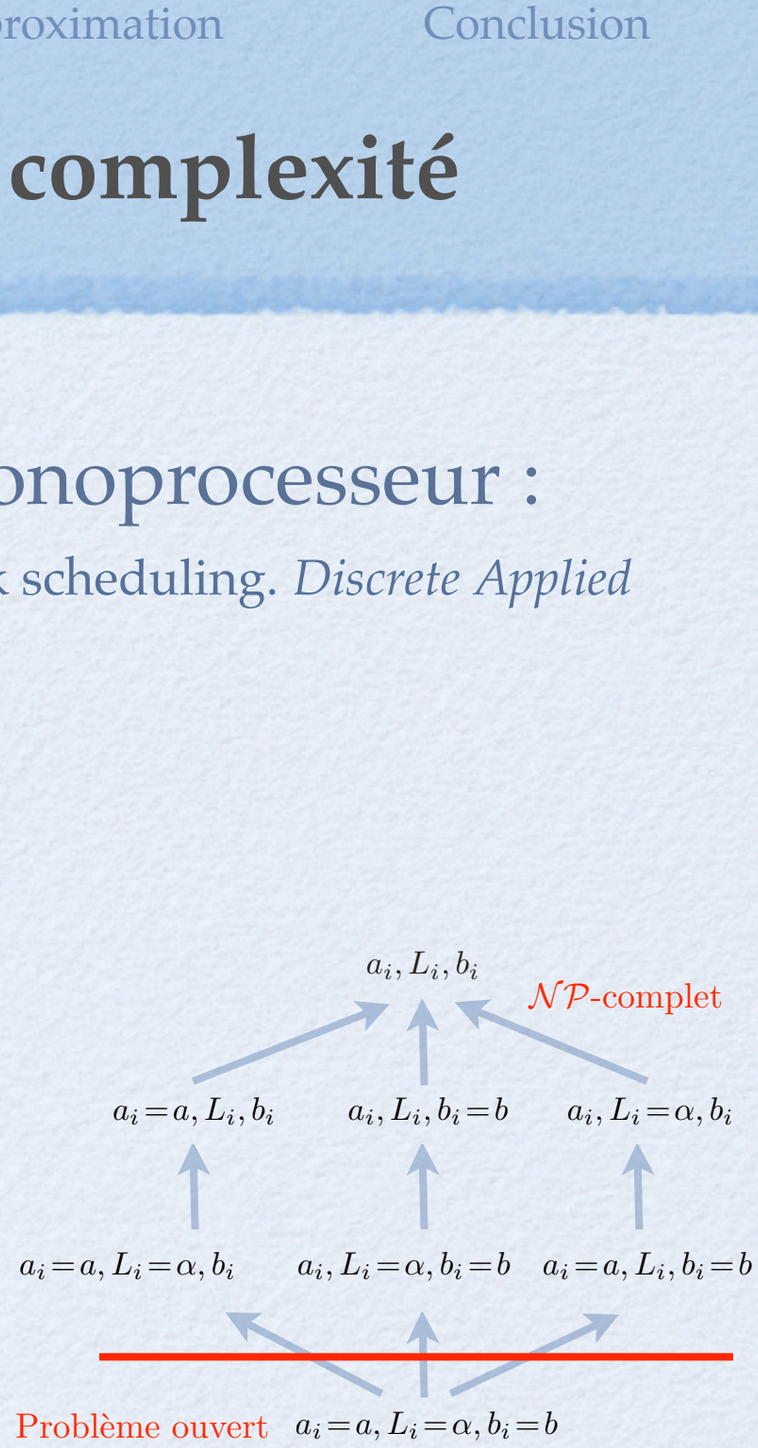
Etat de l'art d'un point de vue de la complexité

- Ordonnancement de tâches-couplées sur monoprocesseur :
  - A.J. Orman and C.N. Potts. On the complexity of coupled-task scheduling. *Discrete Applied Mathematics*, 72 :141–154, 1997.

Résultats de complexité







Avec les T_i et G_p : rien ne change

Qu'en est-il avec G_c ?

Un cas pose problème : $a_i = b_i = p, L_i = \alpha$

Complexité : étude d'un cas particulier

Le cas qui nous intéresse se note :

$$1|\ prec,\ tâche-couplée,\ (p_{a_i} = p_{b_i} = p, L_i = \alpha) \cup (p_{T_i}, pmtn), G_c|C_{max}$$

Complexité : étude d'un cas particulier

Le cas qui nous intéresse se note :

$$1|\ prec,\ tâche-couplée,\ (p_{a_i} = p_{b_i} = p, L_i = \alpha) \cup (p_{T_i}, pmtn), G_c|C_{max}$$

- Nous allons étudier un cas spécifique de ce problème :

Complexité : étude d'un cas particulier

Le cas qui nous intéresse se note :

$$1|\ prec,\ tâche-couplée,\ (p_{a_i} = p_{b_i} = p, L_i = \alpha) \cup (p_{T_i}, pmtn), G_c|C_{max}$$

- Nous allons étudier un cas spécifique de ce problème :

Le problème que nous allons étudier se note :

$$1|\ prec,\ tâche-couplée,\ (p_{a_i} = p_{b_i} = 1, L_i = \alpha) \cup (p_{T_i}, pmtn), G_c|C_{max}$$

Nous le noterons Π

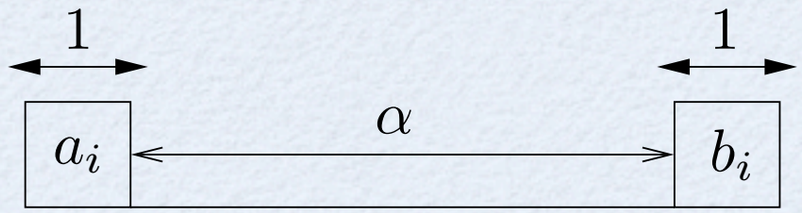
Complexité : étude d'un cas particulier

Le problème que nous allons étudier se note :

$1|\ prec,\ tâche-couplée,\ (p_{a_i} = p_{b_i} = 1, L_i = \alpha) \cup (p_{T_i}, pmtn), G_c|C_{max}$

Nous le noterons Π

- Toutes les tâches d'acquisition sont de la forme :



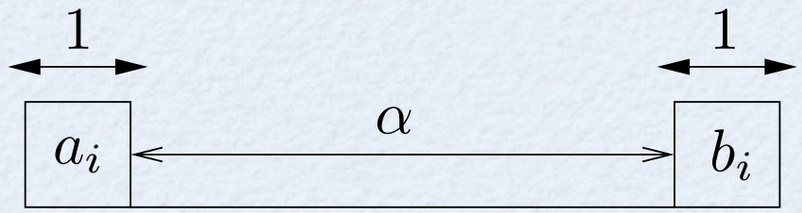
Complexité : étude d'un cas particulier

Le problème que nous allons étudier se note :

$1|\ prec,\ tâche-couplée,\ (p_{a_i} = p_{b_i} = 1, L_i = \alpha) \cup (p_{T_i}, pmtn), G_c|C_{max}$

Nous le noterons Π

- Toutes les tâches d'acquisition sont de la forme :



- Le cas $\alpha = 1$ est polynomial

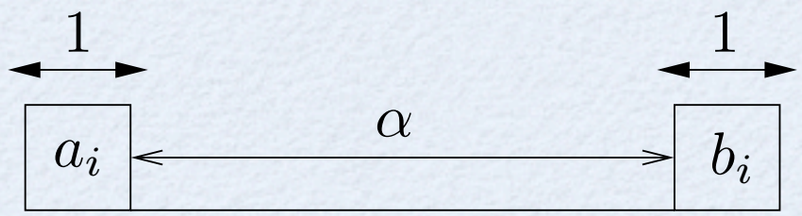
Complexité : étude d'un cas particulier

Le problème que nous allons étudier se note :

$1|\ prec,\ tâche-couplée,\ (p_{a_i} = p_{b_i} = 1, L_i = \alpha) \cup (p_{T_i}, pmtn), G_c|C_{max}$

Nous le noterons Π

- Toutes les tâches d'acquisition sont de la forme :



- Le cas $\alpha = 1$ est polynomial → Couplage maximum

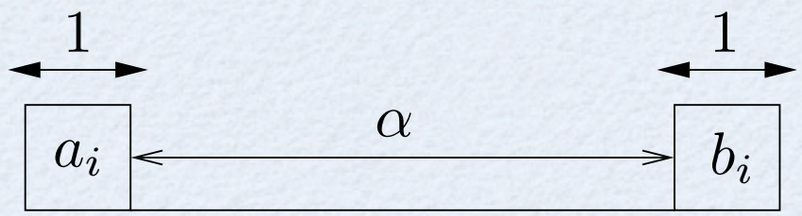
Complexité : étude d'un cas particulier

Le problème que nous allons étudier se note :

$1|\ prec,\ tâche-couplée,\ (p_{a_i} = p_{b_i} = 1, L_i = \alpha) \cup (p_{T_i}, pmtn), G_c|C_{max}$

Nous le noterons Π

- Toutes les tâches d'acquisition sont de la forme :



- Le cas $\alpha = 1$ est polynomial → Couplage maximum
- Le cas $\alpha = 2$ est polynomial si G_c est sans triangle

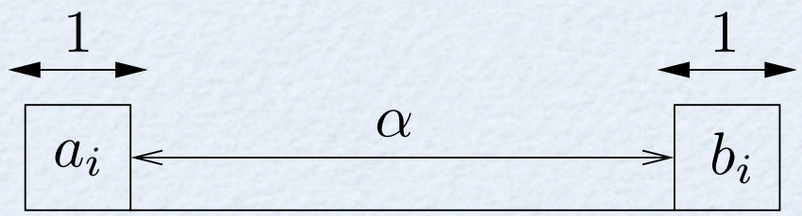
Complexité : étude d'un cas particulier

Le problème que nous allons étudier se note :

$1|\ prec,\ tâche-couplée,\ (p_{a_i} = p_{b_i} = 1, L_i = \alpha) \cup (p_{T_i}, pmtn), G_c|C_{max}$

Nous le noterons Π

- Toutes les tâches d'acquisition sont de la forme :



- Le cas $\alpha = 1$ est polynomial → Couplage maximum
- Le cas $\alpha = 2$ est polynomial si G_c est sans triangle
- Montrons que pour $\alpha > 2$ notre problème Π est $\mathcal{NP}$-complet

Complexité : Etude de la complexité

Théorème

Le problème de décision noté Π

$$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=1, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}$$

est $\mathcal{NP}$-complet

Complexité : Etude de la complexité

Théorème

Le problème de décision noté Π

$1|\ prec,\ tâche-couplée,\ (p_{a_i} = p_{b_i} = 1, L_i = \alpha) \cup (p_{T_i}, pmtn), G_c|C_{max}$

est $\mathcal{NP}$-complet

Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j = 1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

Complexité : Etude de la complexité

Théorème

Le problème de décision noté Π

$1|\ prec,\ tâche-couplée,\ (p_{a_i} = p_{b_i} = 1, L_i = \alpha) \cup (p_{T_i}, pmtn), G_c|C_{max}$

est $\mathcal{NP}$-complet

Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j = 1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

INSTANCE : Un graphe $G = (V, E)$ où $|V| = n$, et un entier K

QUESTION : Peut on trouver un sous-graphe complet de taille K dans G ?

Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j = 1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

Illustration de la transformation :

Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

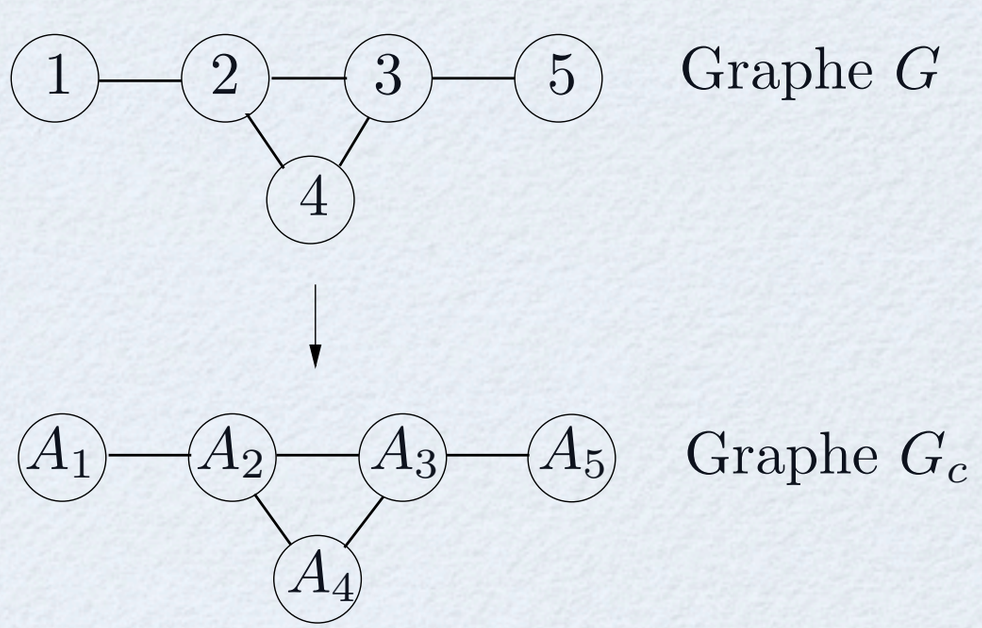
Illustration de la transformation :

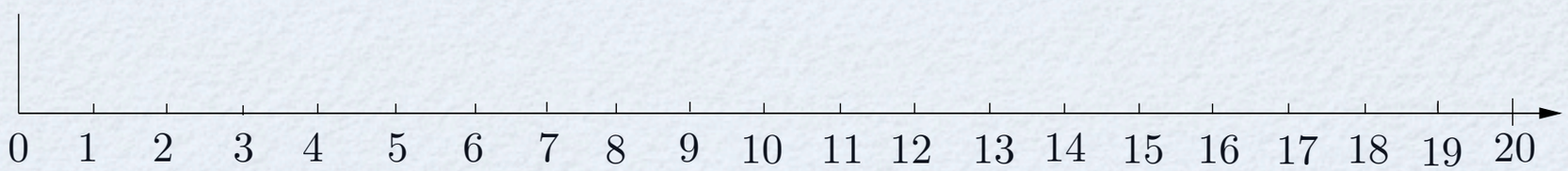
$n=5 \qquad L_i=\alpha=2$

$p_{T_i}=\alpha=2$

$C_{max}^{opt}=20 \quad K=\alpha+1=3$

G_p : $A_1 \to T_1$, $A_2 \to T_2$, $A_3 \to T_3$, $A_4 \to T_4$, $A_5 \to T_5$





Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

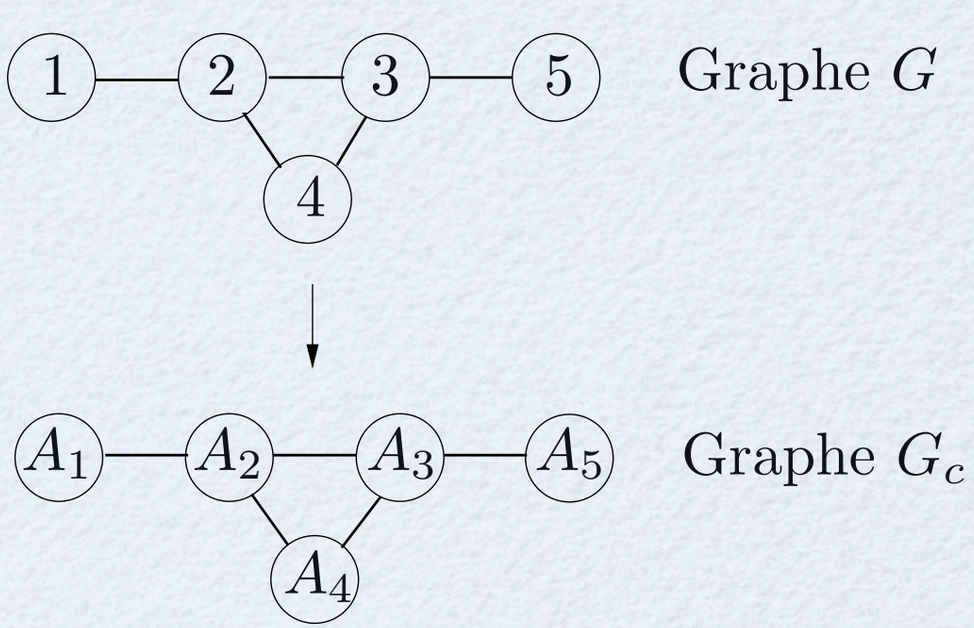
Illustration de la transformation :

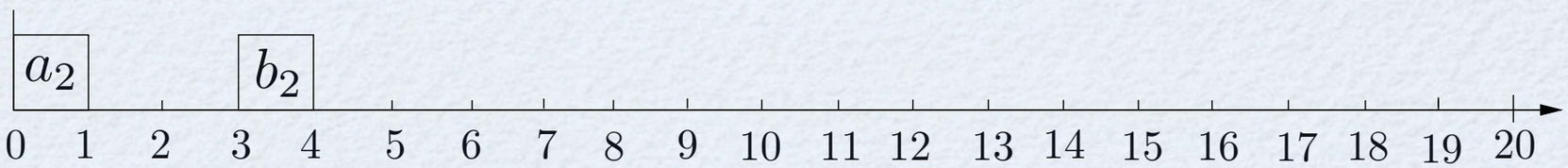
$n=5 \qquad L_i=\alpha=2$

$p_{T_i}=\alpha=2$

$C_{max}^{opt}=20 \quad K=\alpha+1=3$

G_p : $A_1 \to T_1$, $A_2 \to T_2$, $A_3 \to T_3$, $A_4 \to T_4$, $A_5 \to T_5$





Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

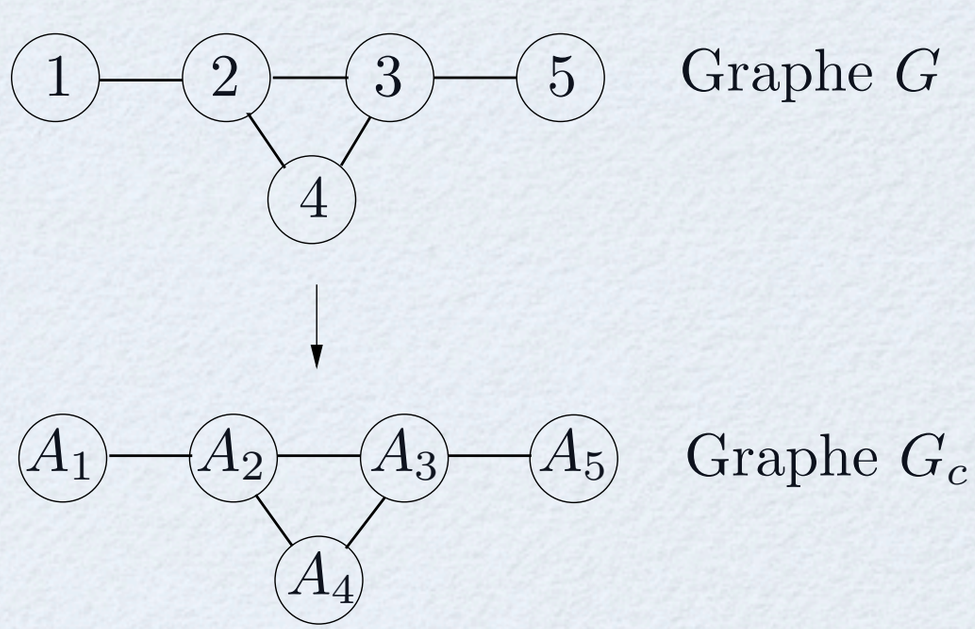
Illustration de la transformation :

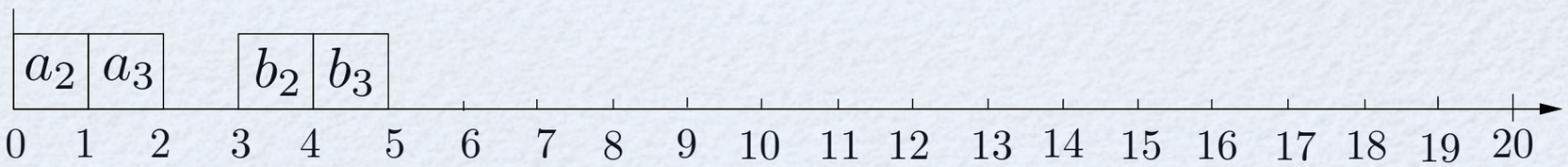
$n=5 \qquad L_i=\alpha=2$

$p_{T_i}=\alpha=2$

$C_{max}^{opt}=20 \quad K=\alpha+1=3$

G_p : A_1 A_2 A_3 A_4 A_5 ↓ ↓ ↓ ↓ ↓ T_1 T_2 T_3 T_4 T_5





Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

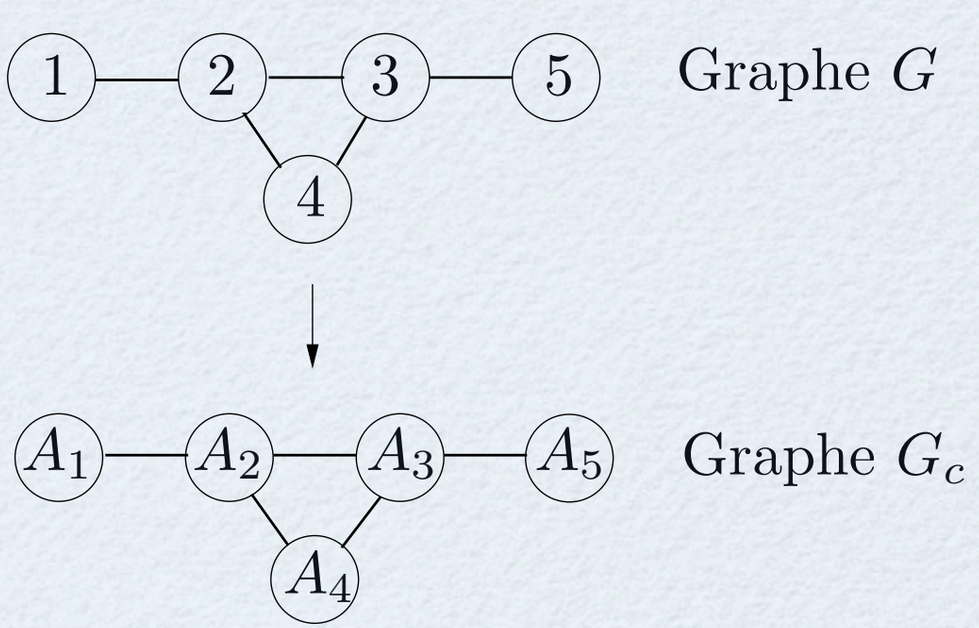
Illustration de la transformation :

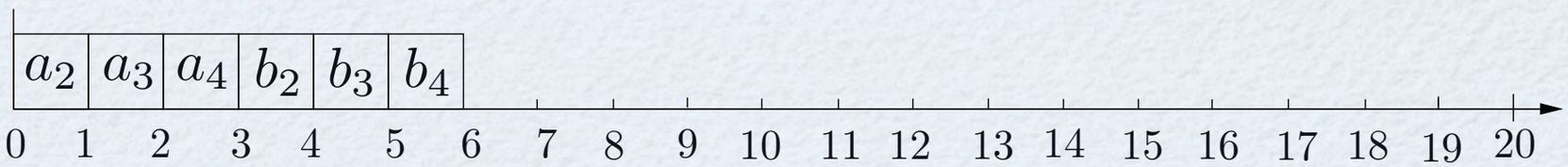
$n=5 \qquad L_i=\alpha=2$

$p_{T_i}=\alpha=2$

$C_{max}^{opt}=20 \quad K=\alpha+1=3$

G_p : $A_1 \to T_1$, $A_2 \to T_2$, $A_3 \to T_3$, $A_4 \to T_4$, $A_5 \to T_5$





Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

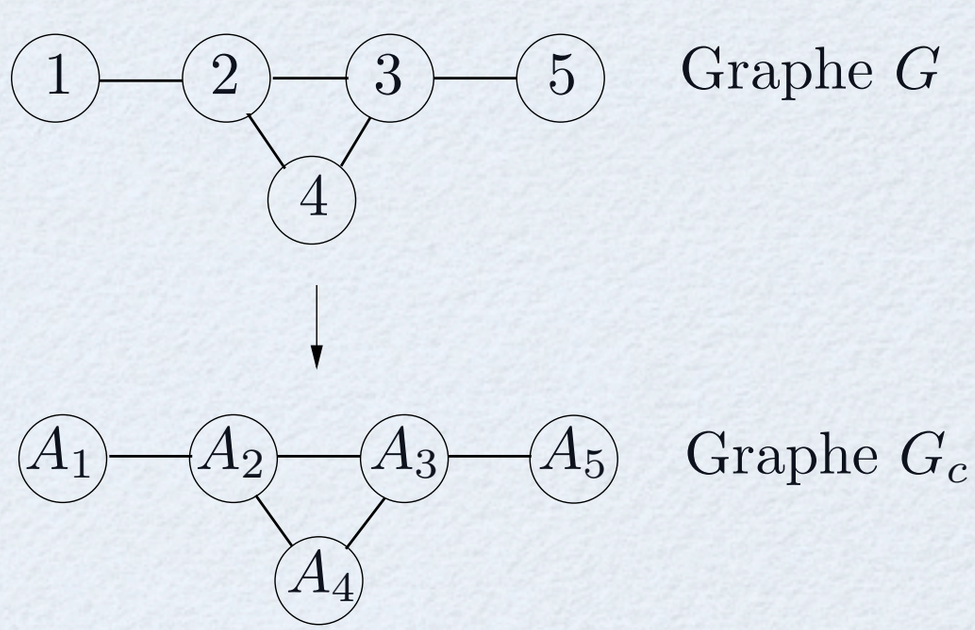
Illustration de la transformation :

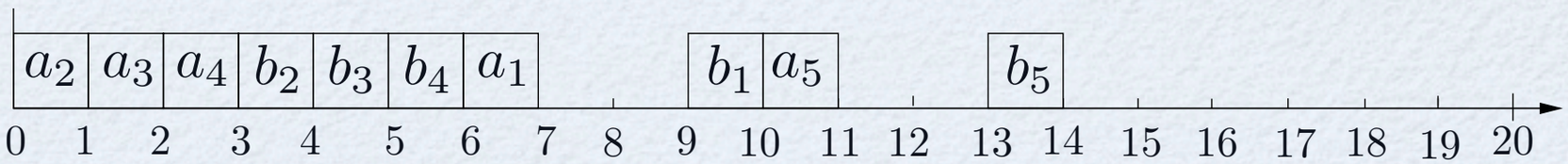
$n=5 \qquad L_i=\alpha=2$

$p_{T_i}=\alpha=2$

$C_{max}^{opt}=20 \quad K=\alpha+1=3$

G_p : A_1 A_2 A_3 A_4 A_5 (↓) T_1 T_2 T_3 T_4 T_5





Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

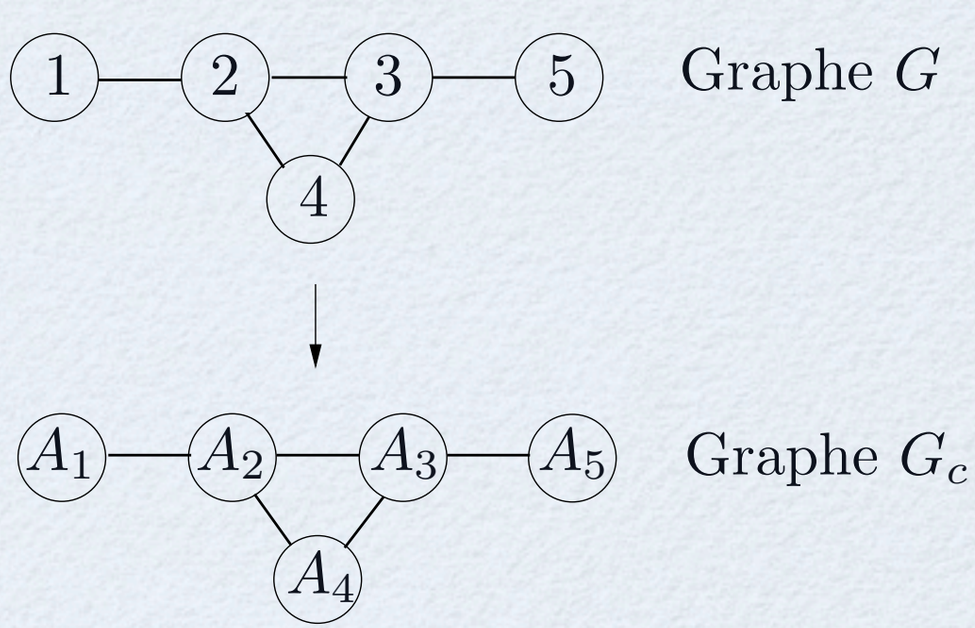
Illustration de la transformation :

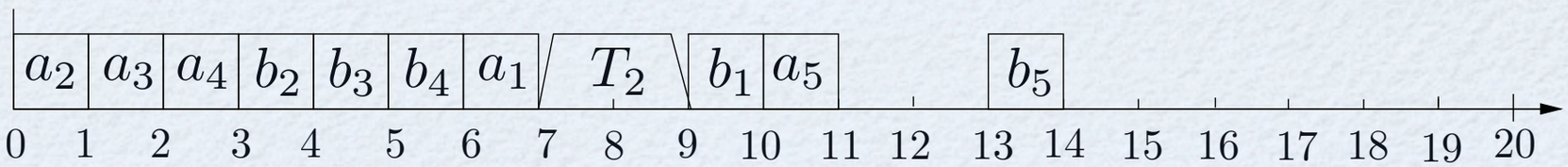
$n=5 \qquad L_i=\alpha=2$

$p_{T_i}=\alpha=2$

$C_{max}^{opt}=20 \quad K=\alpha+1=3$

G_p : A_1 A_2 A_3 A_4 A_5 (↓) T_1 T_2 T_3 T_4 T_5





Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

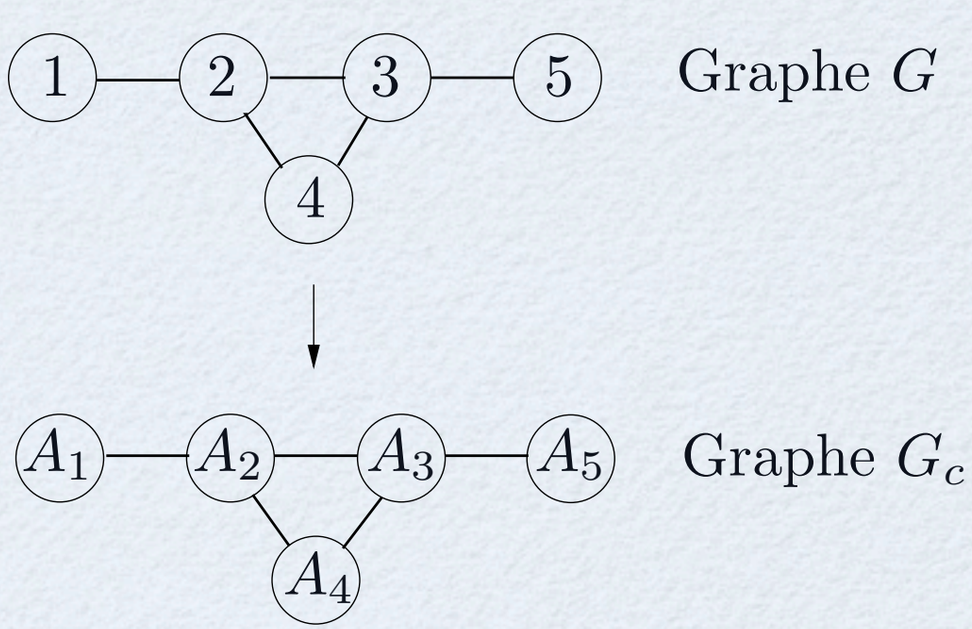
Illustration de la transformation :

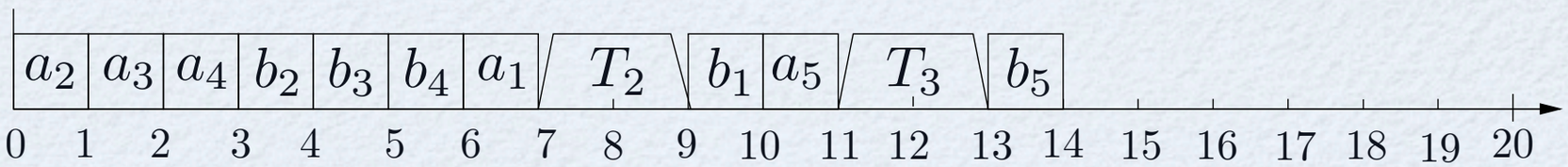
$n=5 \qquad L_i=\alpha=2$

$p_{T_i}=\alpha=2$

$C_{max}^{opt}=20 \quad K=\alpha+1=3$

G_p : A_1 A_2 A_3 A_4 A_5 ↓ ↓ ↓ ↓ ↓ T_1 T_2 T_3 T_4 T_5





Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

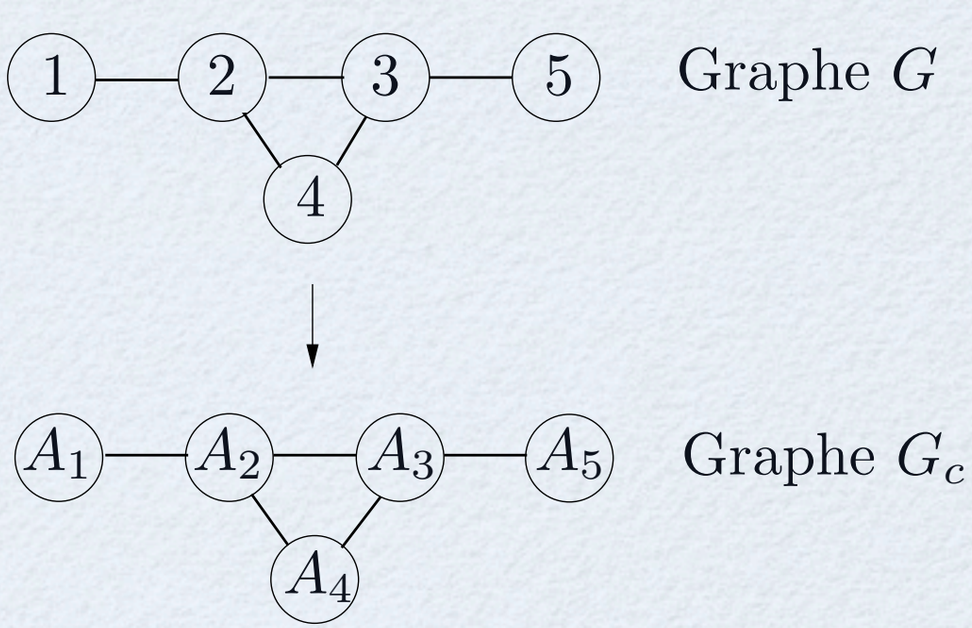
Illustration de la transformation :

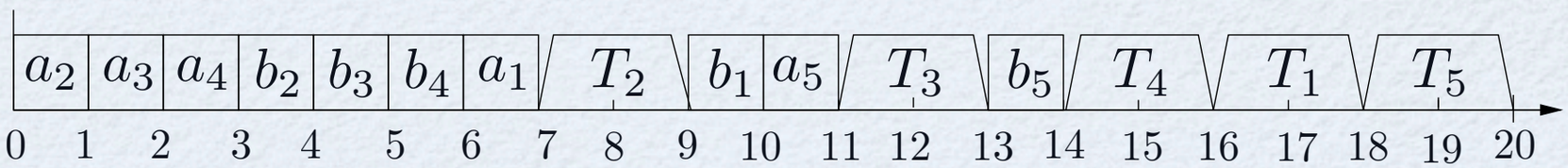
$n=5 \qquad L_i=\alpha=2$

$p_{T_i}=\alpha=2$

$C_{max}^{opt}=20 \quad K=\alpha+1=3$

G_p : A_1 A_2 A_3 A_4 A_5 (↓) T_1 T_2 T_3 T_4 T_5





Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

- Supposons l'existence d'une clique de taille K dans le graphe G

Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

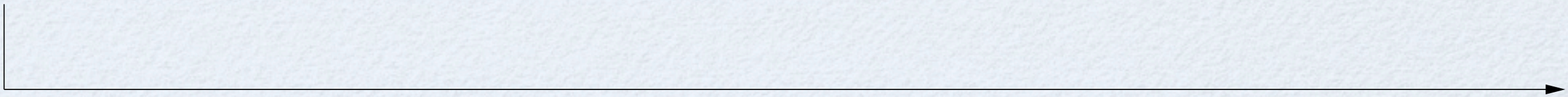
- Supposons l'existence d'une clique de taille K dans le graphe G

- Montrons qu'il existe un ordonnancement sans temps d'inactivité :

Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

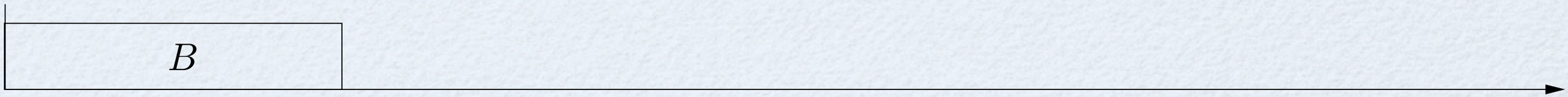
- Supposons l'existence d'une clique de taille K dans le graphe G
- Montrons qu'il existe un ordonnancement sans temps d'inactivité :



Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

- Supposons l'existence d'une clique de taille K dans le graphe G
- Montrons qu'il existe un ordonnancement sans temps d'inactivité :

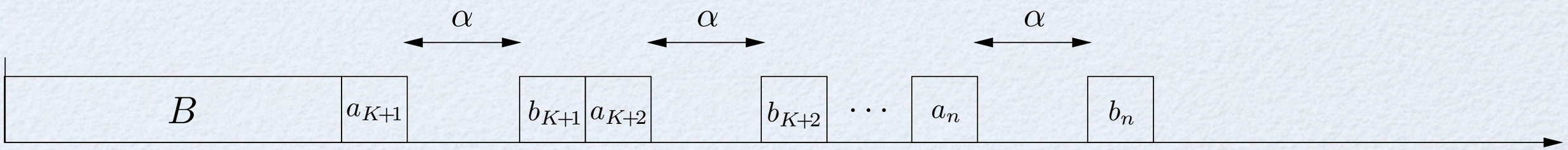


B est composé de K tâches A_i, pour $i=1,\ldots,K$

Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

- Supposons l'existence d'une clique de taille K dans le graphe G
- Montrons qu'il existe un ordonnancement sans temps d'inactivité :



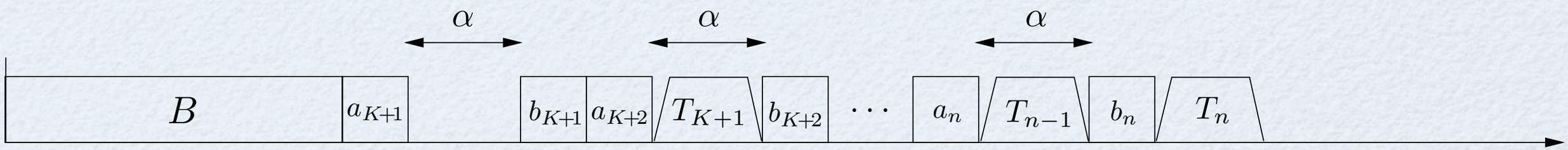
B est composé de K tâches A_i, pour $i=1,\ldots,K$

Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

- Supposons l'existence d'une clique de taille K dans le graphe G

- Montrons qu'il existe un ordonnancement sans temps d'inactivité :

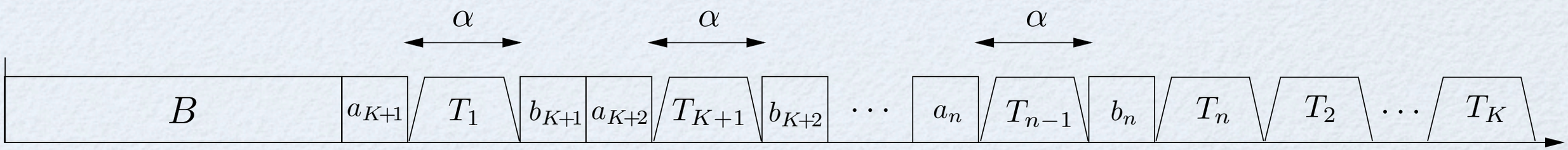


B est composé de K tâches A_i, pour $i=1,\ldots,K$

Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

- Supposons l'existence d'une clique de taille K dans le graphe G
- Montrons qu'il existe un ordonnancement sans temps d'inactivité :



B est composé de K tâches A_i, pour $i=1,\ldots,K$

Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j = 1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

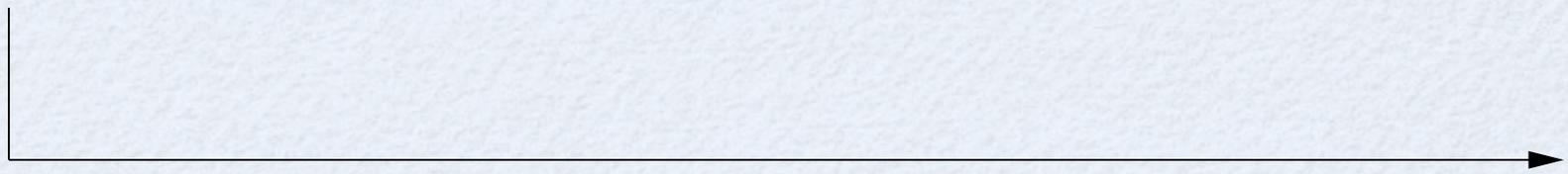
- Supposons l'existence d'un ordonnancement sans temps d'inactivité

Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

⇐ • Supposons l'existence d'un ordonnancement sans temps d'inactivité

• Montrons qu'il existe une clique de taille $K=(\alpha+1)$ dans G :

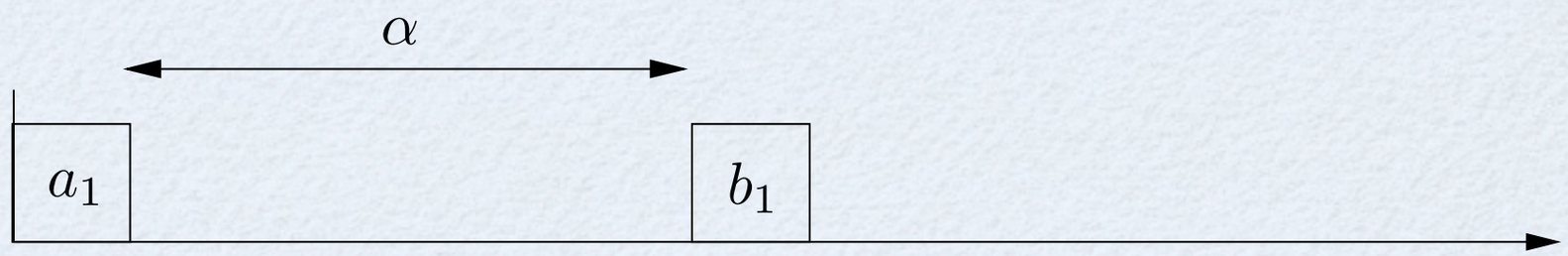


Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

⇐ • Supposons l'existence d'un ordonnancement sans temps d'inactivité

• Montrons qu'il existe une clique de taille $K=(\alpha+1)$ dans G :

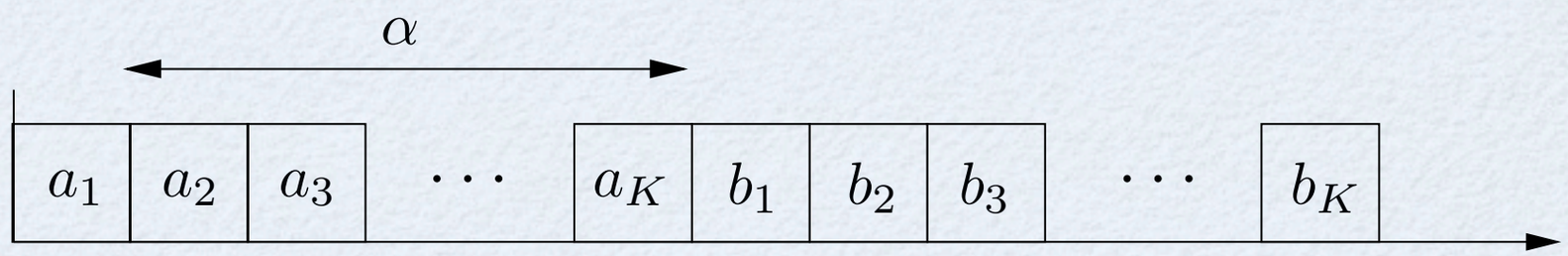


Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

⇐ • Supposons l'existence d'un ordonnancement sans temps d'inactivité

• Montrons qu'il existe une clique de taille $K=(\alpha+1)$ dans G :

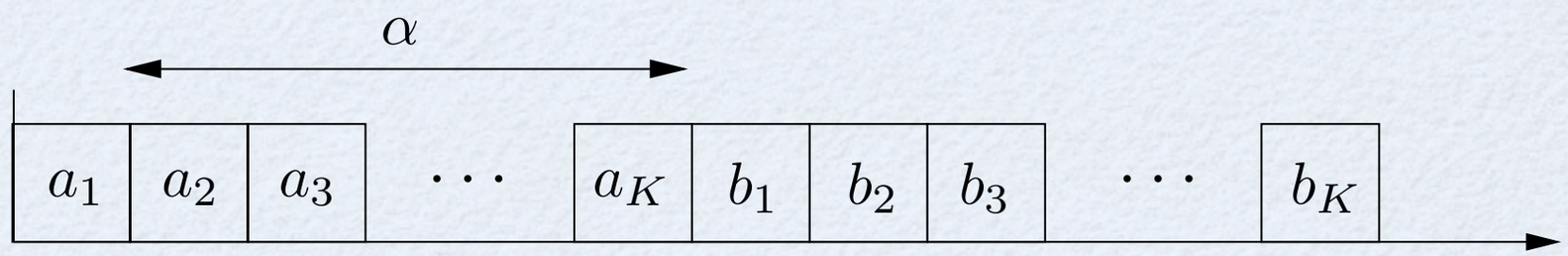


Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

⇐ • Supposons l'existence d'un ordonnancement sans temps d'inactivité

• Montrons qu'il existe une clique de taille $K=(\alpha+1)$ dans G :



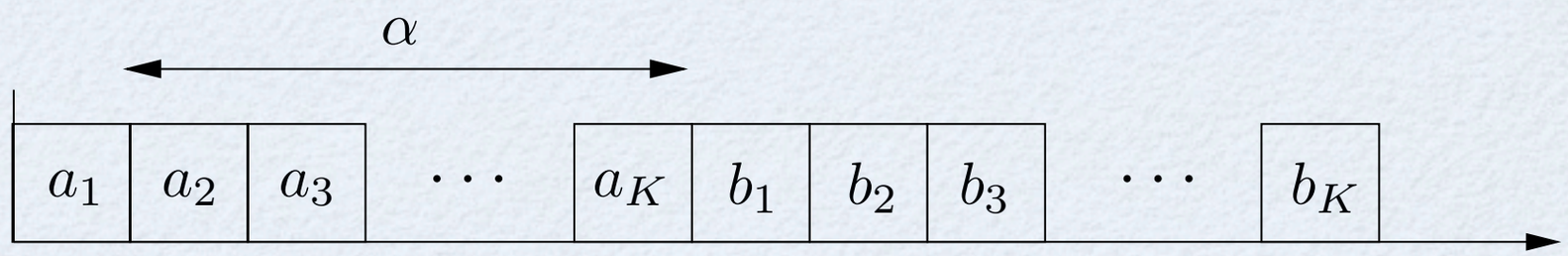
Il existe une clique de taille K dans G_c

Preuve :

- Approche similaire à celle de Lenstra pour $P|prec, P_j=1|C_{max}$
- Réduction faite à partir du problème $\mathcal{NP}$-complet Clique

⇐ • Supposons l'existence d'un ordonnancement sans temps d'inactivité

• Montrons qu'il existe une clique de taille $K=(\alpha+1)$ dans G :



Il existe une clique de taille K dans G_c ⟶ Il existe une clique de taille K dans G □

Ainsi, on a :

$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=1, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}$ $\mathcal{NP}$-complet

Ainsi, on a :

$$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=1, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}\ \mathcal{NP}\text{-complet}$$

$$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=p, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}\ \mathcal{NP}\text{-complet}$$

Ainsi, on a :

$$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=1, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}\ \mathcal{NP}\text{-complet}$$

$$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=p, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}\ \mathcal{NP}\text{-complet}$$

Ce qui entraîne :

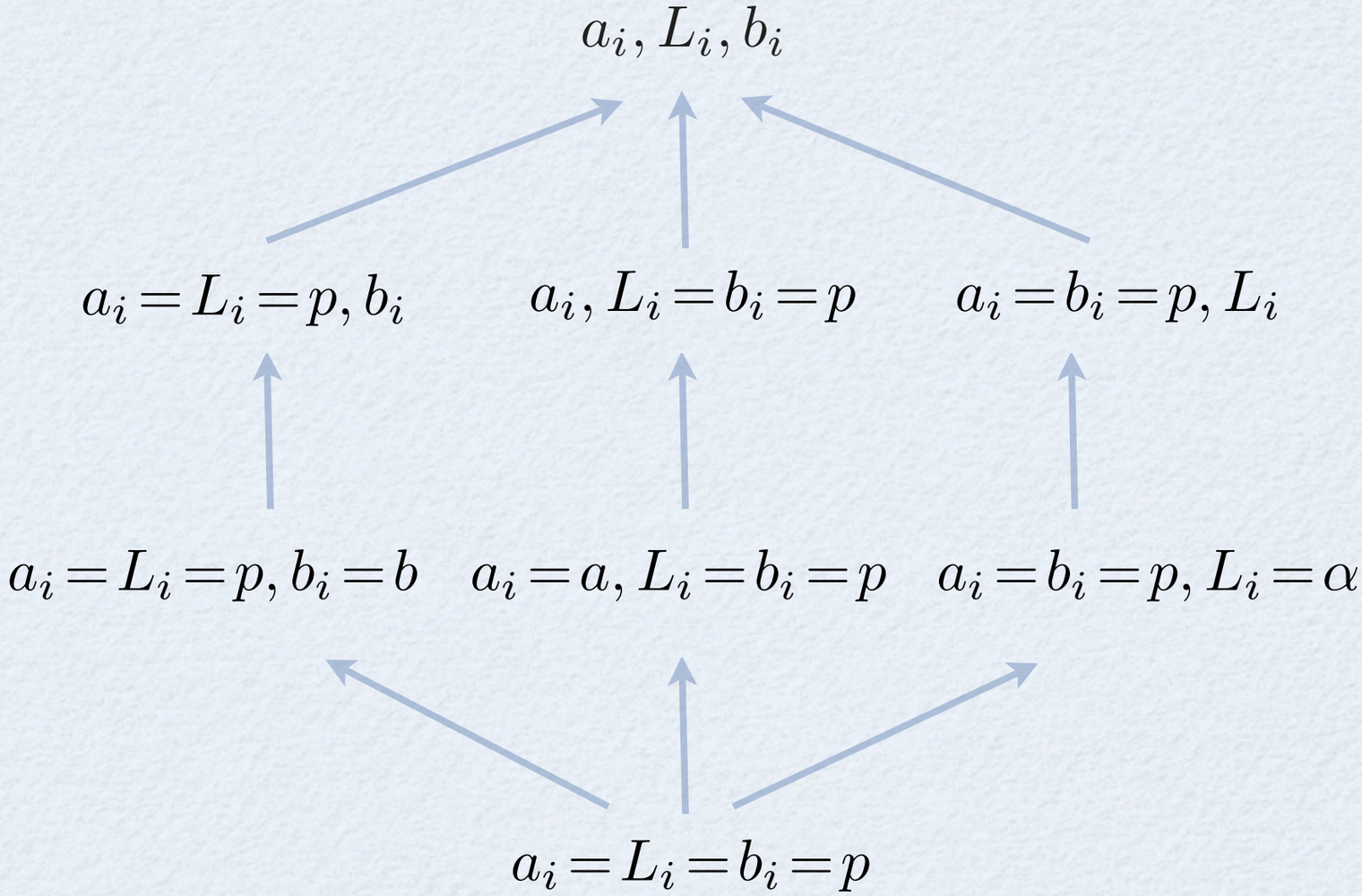
Ainsi, on a :

$$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=1, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}\ \mathcal{NP}\text{-complet}$$

$$\downarrow$$

$$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=p, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}\ \mathcal{NP}\text{-complet}$$

Ce qui entraîne :



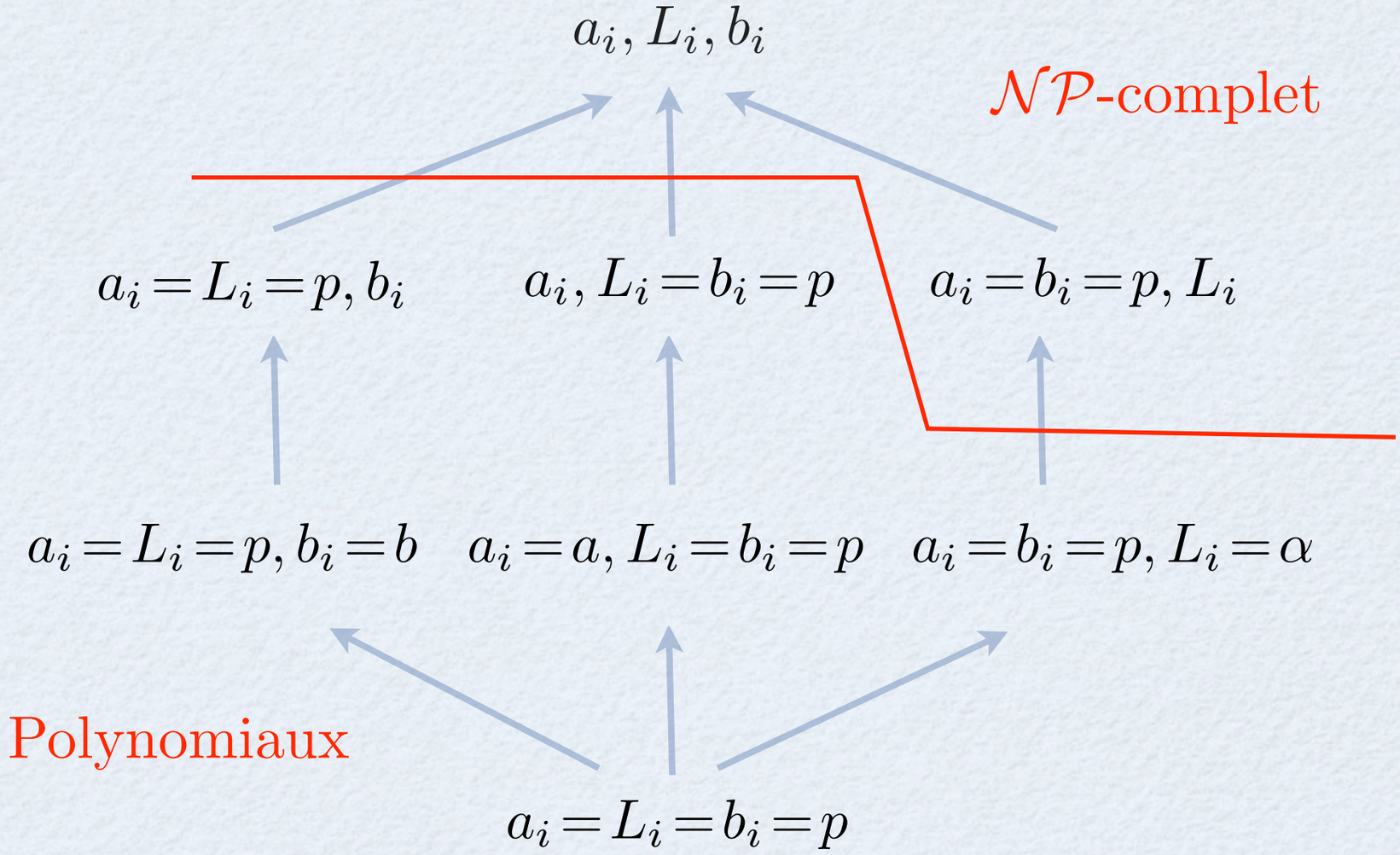
Ainsi, on a :

$$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=1, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}\ \mathcal{NP}\text{-complet}$$

$$\downarrow$$

$$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=p, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}\ \mathcal{NP}\text{-complet}$$

Ce qui entraîne :



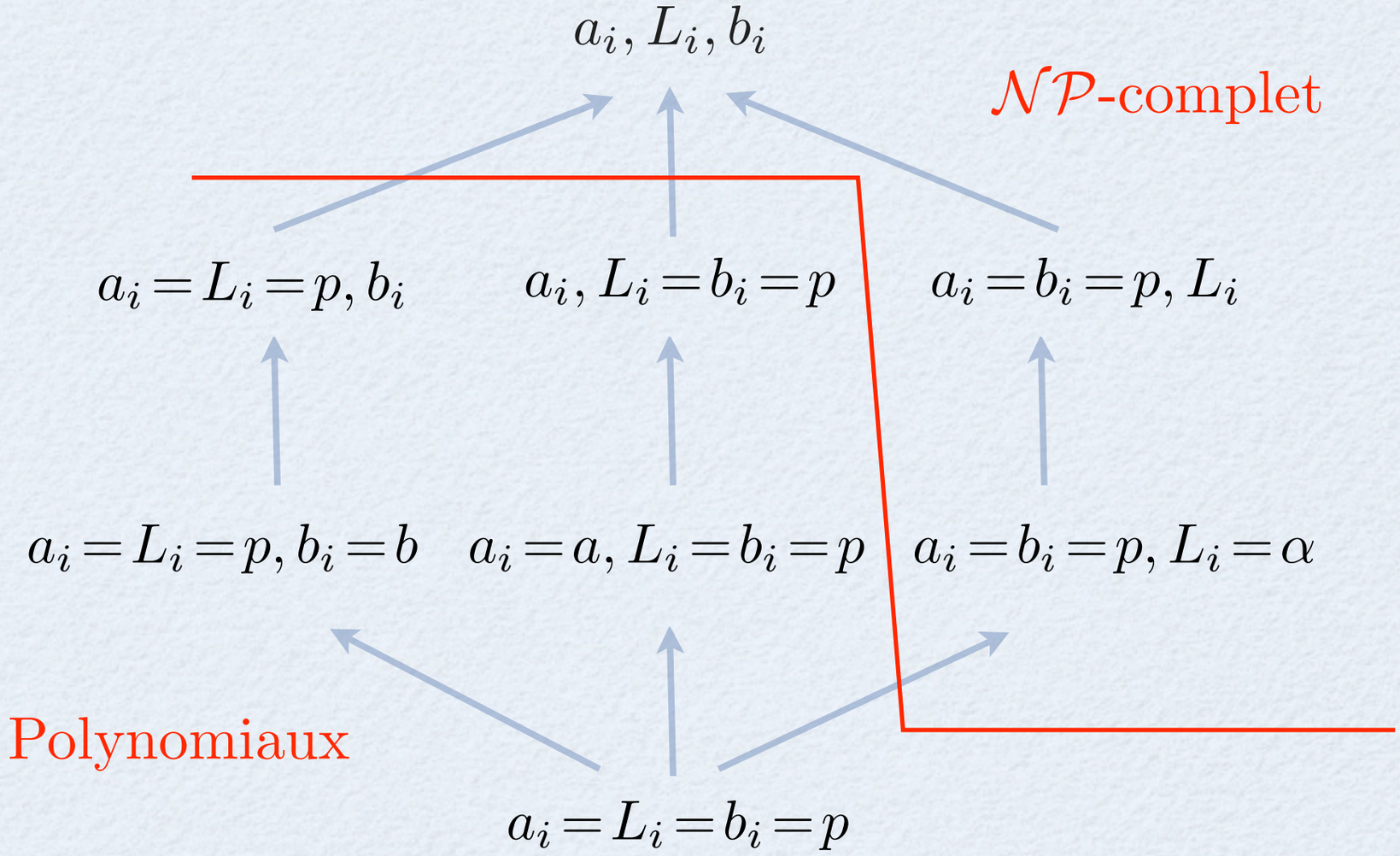
Ainsi, on a :

$$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=1, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}\ \mathcal{NP}\text{-complet}$$

$$\downarrow$$

$$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=p, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}\ \mathcal{NP}\text{-complet}$$

Ce qui entraîne :



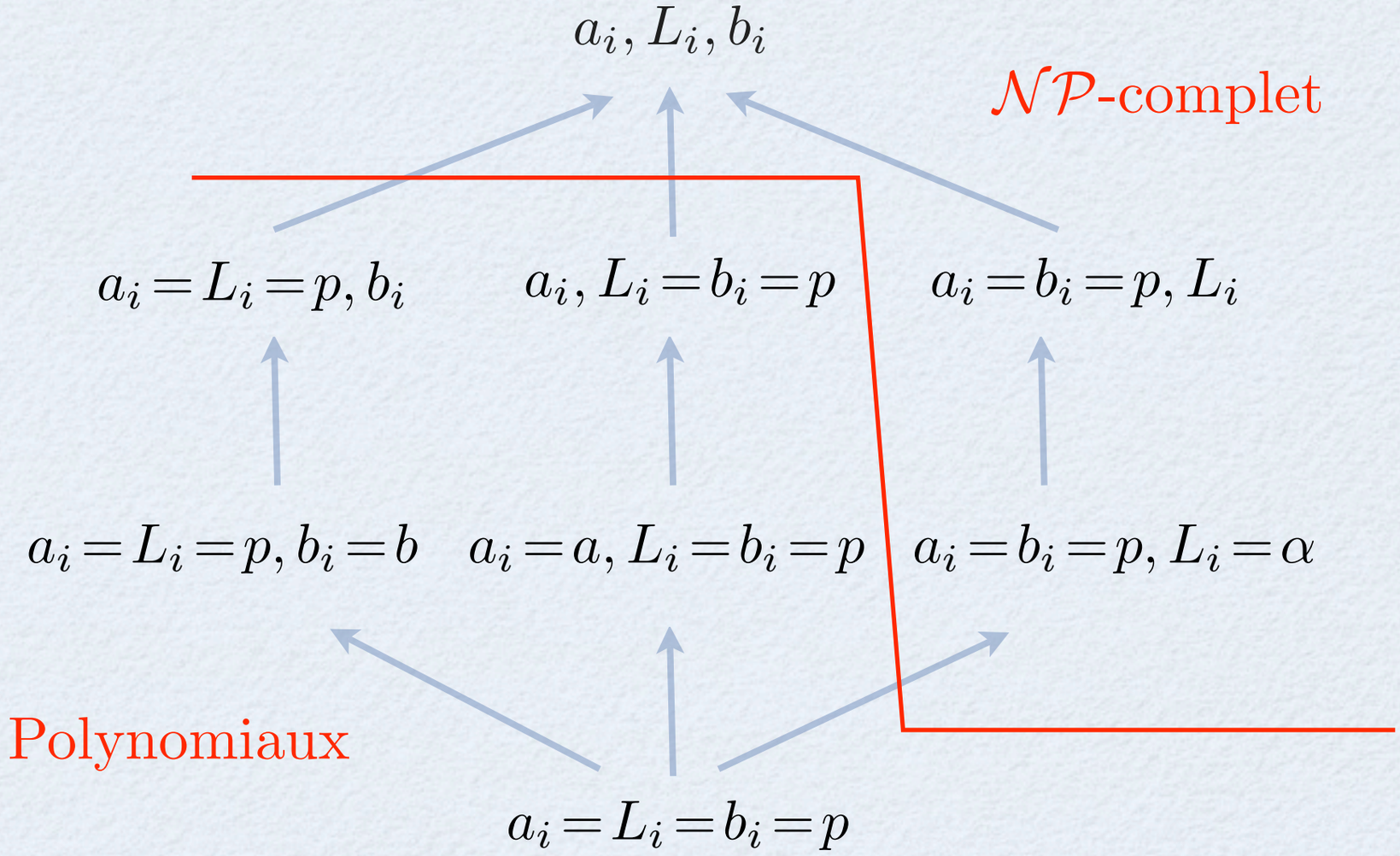
Ainsi, on a :

$$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=1, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}\ \mathcal{NP}\text{-complet}$$

$$\downarrow$$

$$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=p, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}\ \mathcal{NP}\text{-complet}$$

Ce qui entraîne :

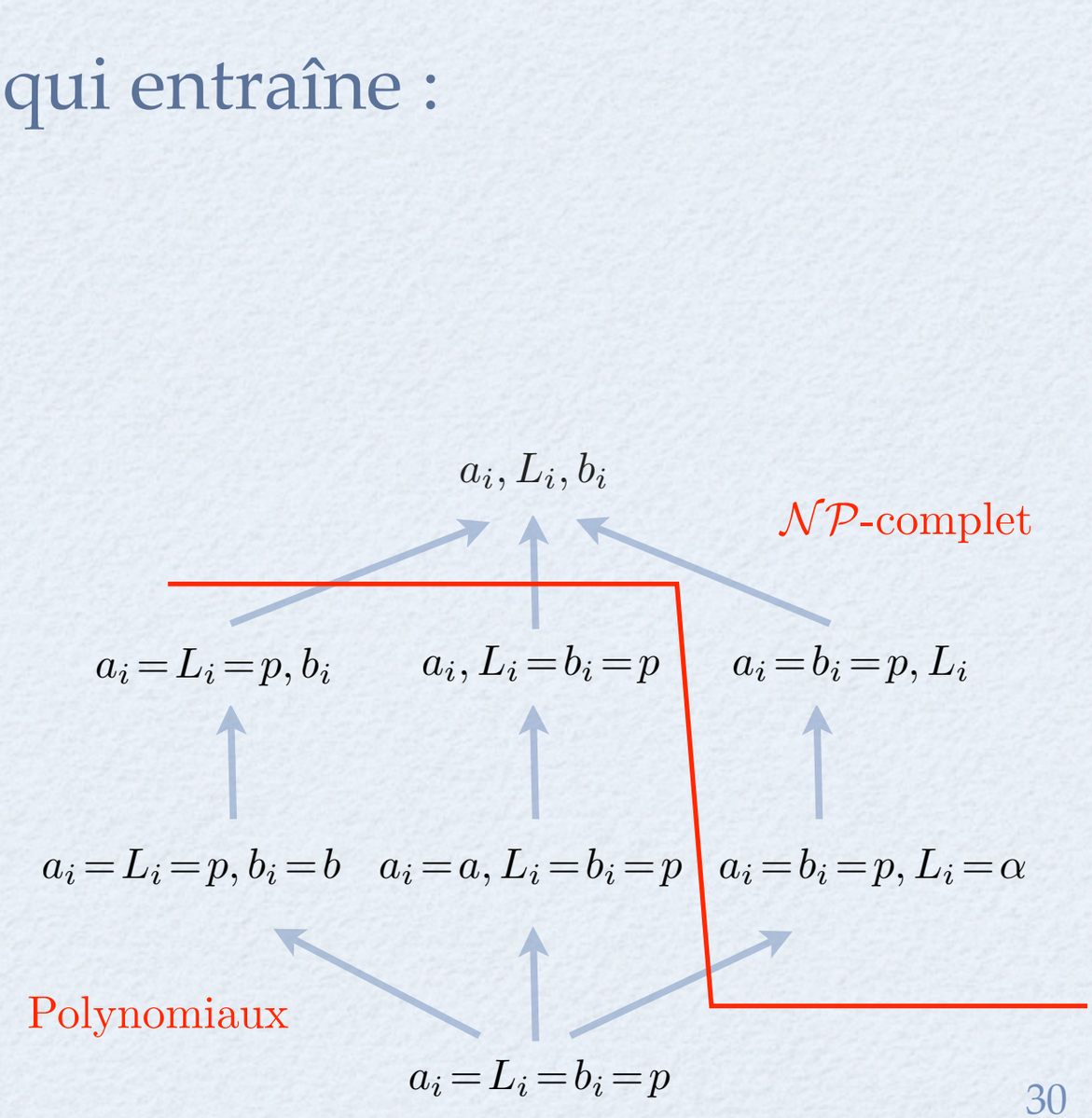


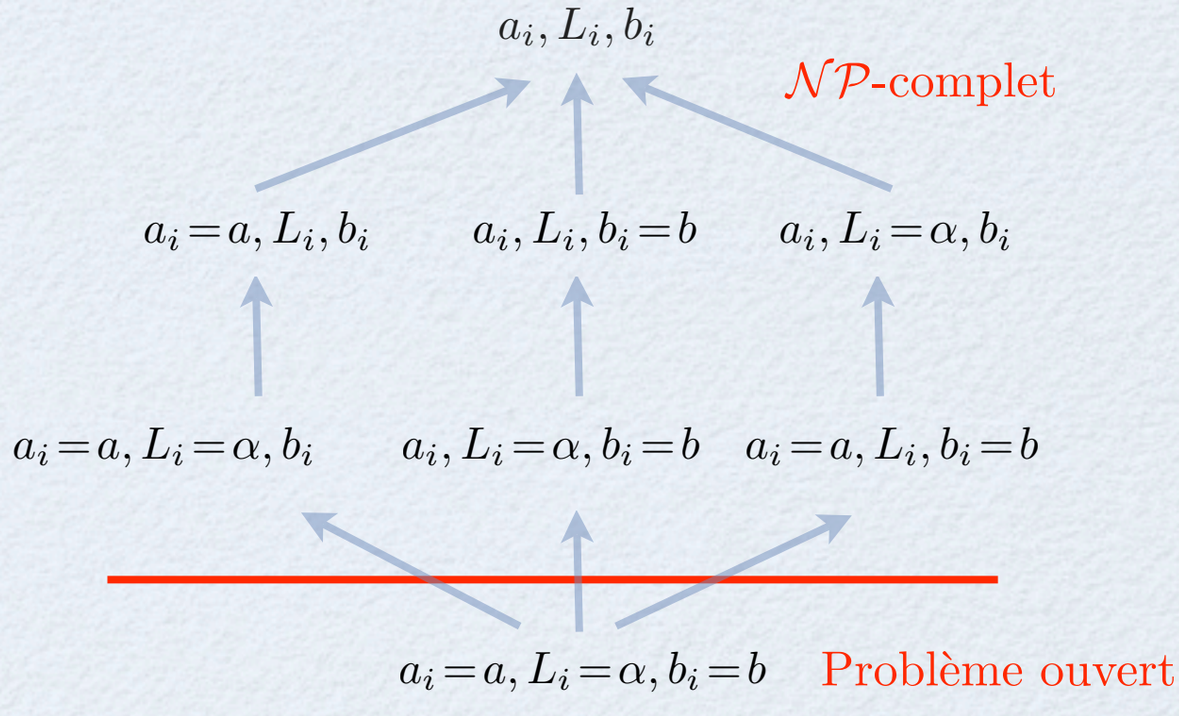
Ainsi, on a :

$$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=1, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}\ \mathcal{NP}\text{-complet}$$

$$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=p, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}\ \mathcal{NP}\text{-complet}$$

Ce qui entraîne :



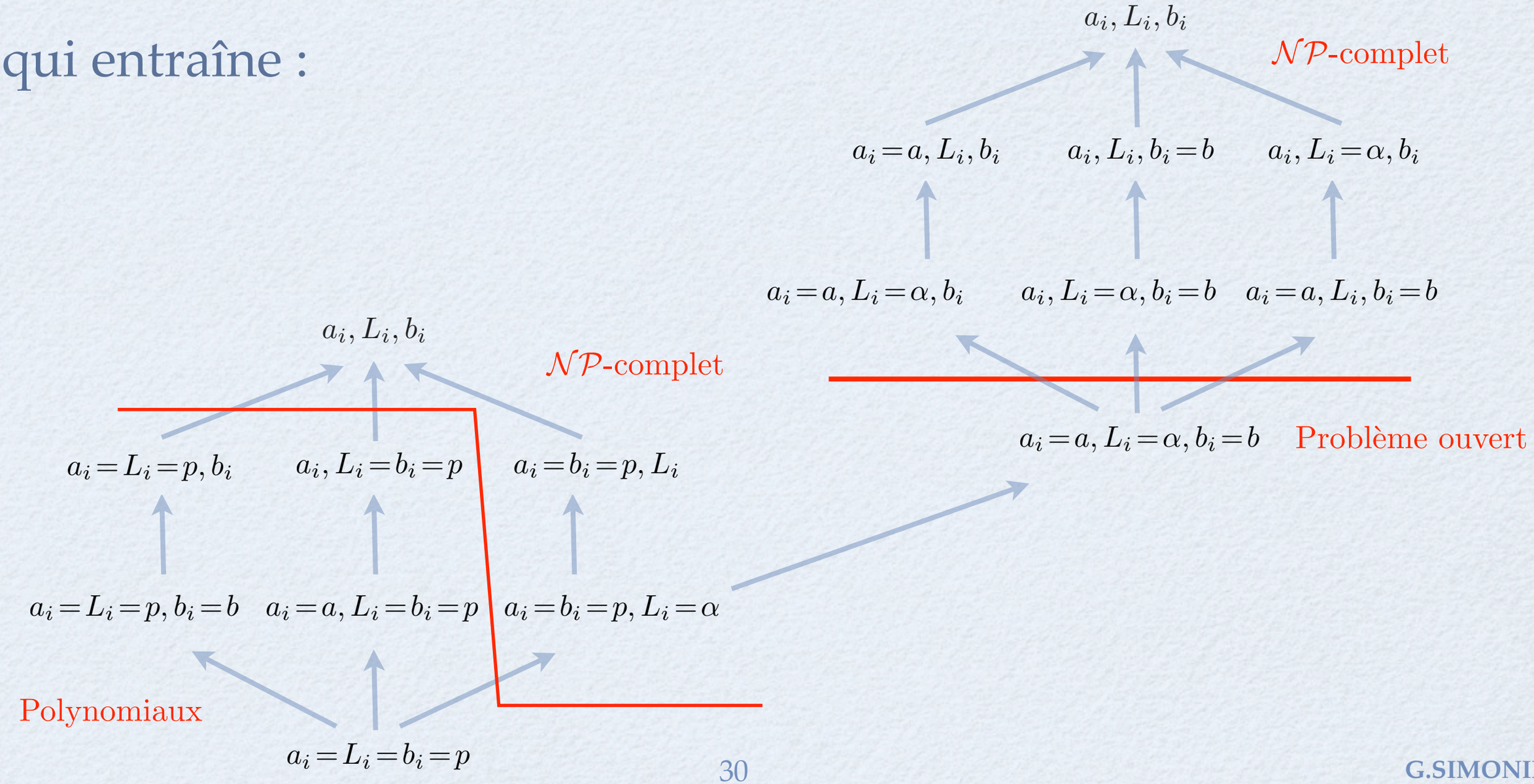


Ainsi, on a :

$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=1, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}$ $\mathcal{NP}$-complet

$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=p, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}$ $\mathcal{NP}$-complet

Ce qui entraîne :



Ainsi, on a :

$$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=1, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}\ \mathcal{NP}\text{-complet}$$

$$1|\ prec,\ tâche-couplée,\ (p_{a_i}=p_{b_i}=p, L_i=\alpha)\cup(p_{T_i}, pmtn), G_c|C_{max}\ \mathcal{NP}\text{-complet}$$

Ce qui entraîne :

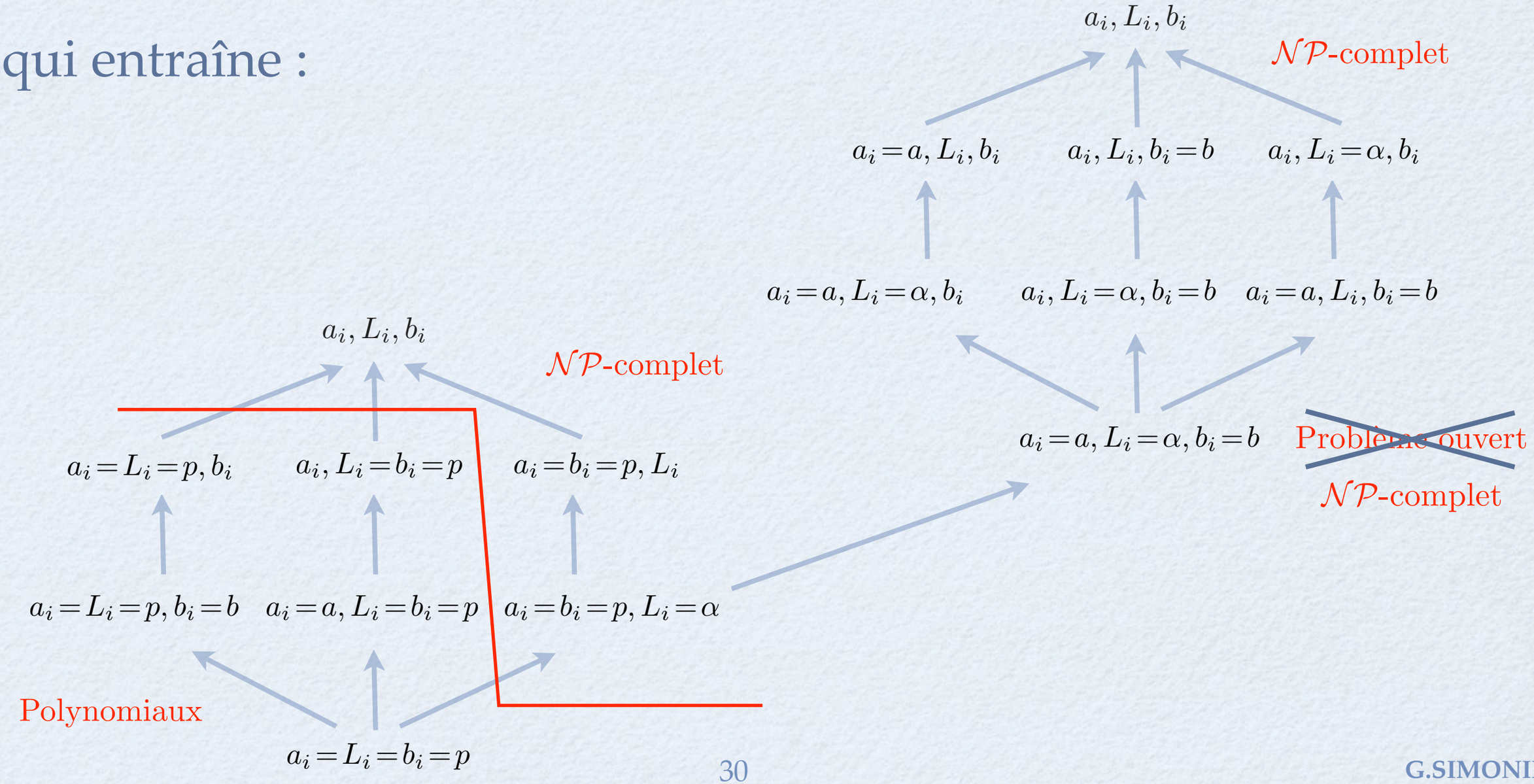


Table des matières

Approximation : Approche spécifique

- Techniques habituelles pour les tâches-couplées sur monoprocesseur non applicables.
- La solution :
  - Une approche différente
  - Penser à remplir les temps de latence des tâches d'acquisition tout en gérant les contraintes de compatibilité
  - Des heuristiques qui exploitent le graphe de compatibilité
  - Utiliser les tâches de traitement pour remplir les temps de latence

Exemples d'heuristiques pour nos problèmes

Remarque :

Le pire des cas sera lorsque les tâches de traitement auront un temps d'exécution unitaire $p_{T_i} = 1$

Exemples d'heuristiques pour nos problèmes

Remarque :

Le pire des cas sera lorsque les tâches de traitement auront un temps d'exécution unitaire $p_{T_i} = 1$

Les idées :

- Couplage maximum
- Cliques maximales
- Poupées russes

Première Heuristique : Couplage maximum

Algorithme en temps polynomial :

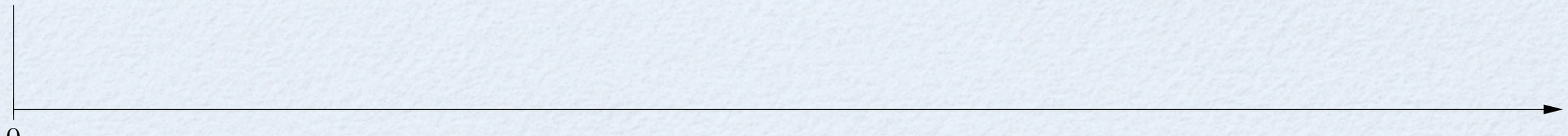
Première Heuristique : Couplage maximum

Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

Résultat : C^h_{max}

Début



Première Heuristique : Couplage maximum

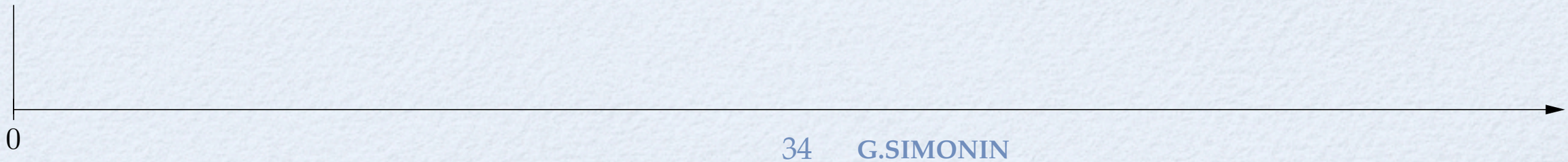
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

Résultat : C^h_{max}

Début

- Chercher un couplage maximum M dans G_c



Première Heuristique : Couplage maximum

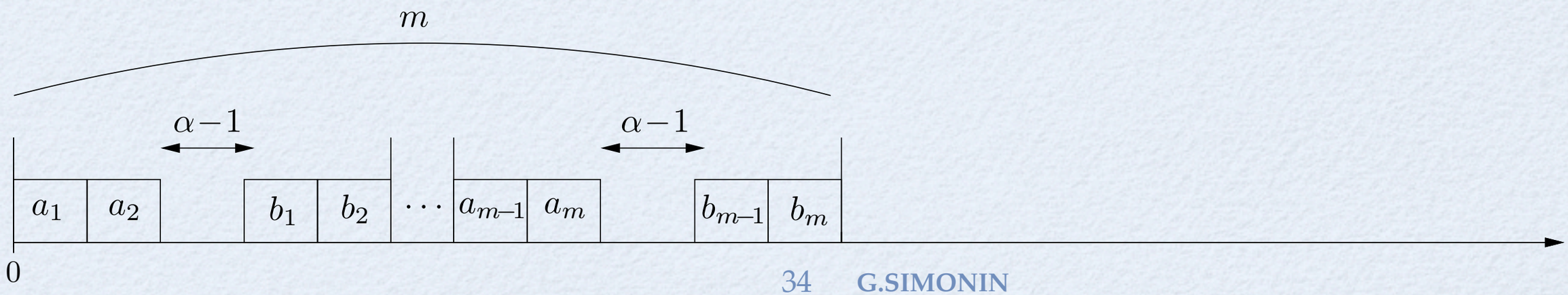
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

Résultat : C^h_{max}

Début

- Chercher un couplage maximum M dans G_c
- Pour chaque arête (i,j) du couplage, A_i et A_j sont ordonnancées telles que $t_{a_j} = t_{a_i} + p_{a_i}$



Première Heuristique : Couplage maximum

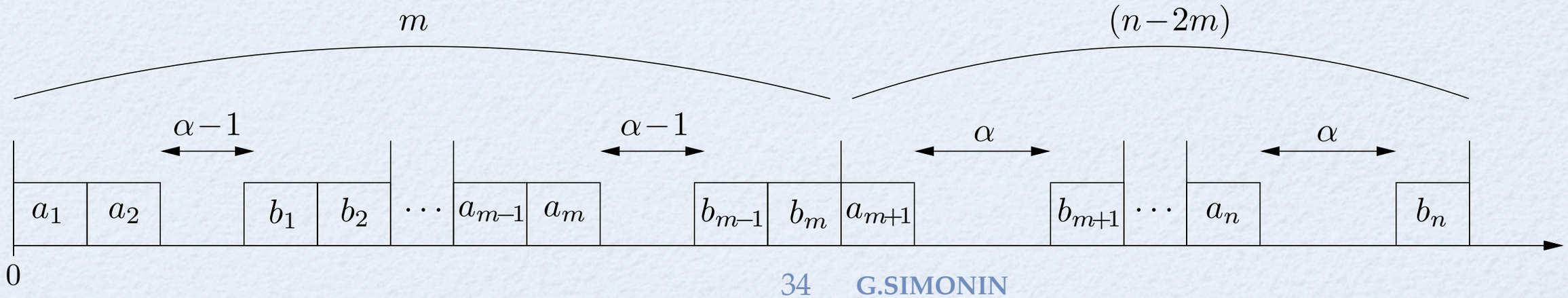
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

Résultat : C^h_{max}

Début

- Chercher un couplage maximum M dans G_c
- Pour chaque arête (i,j) du couplage, A_i et A_j sont ordonnancées telles que $t_{a_j} = t_{a_i} + p_{a_i}$
- Pour chaque sommet i restant, nous exécutons A_i



Première Heuristique : Couplage maximum

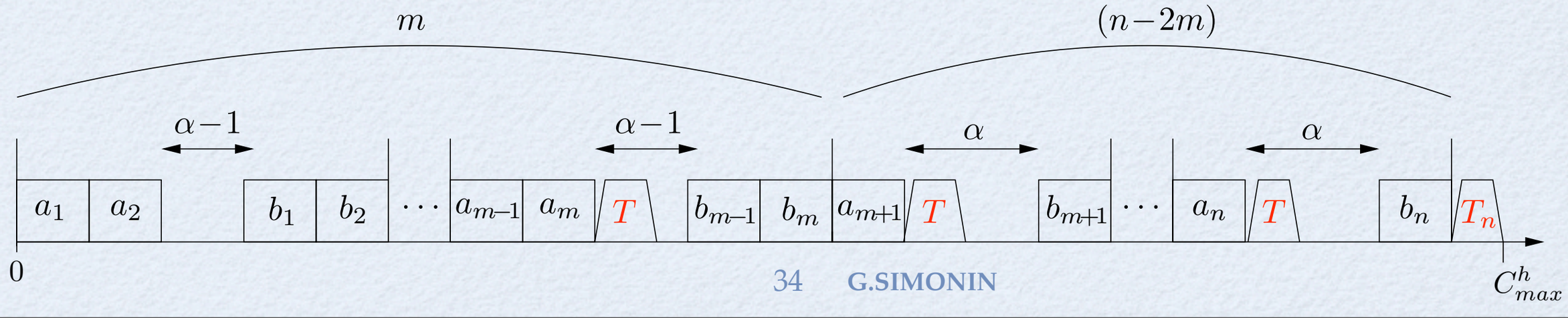
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

Résultat : C^h_{max}

Début

- Chercher un couplage maximum M dans G_c
- Pour chaque arête (i,j) du couplage, A_i et A_j sont ordonnancées telles que $t_{a_j} = t_{a_i} + p_{a_i}$
- Pour chaque sommet i restant, nous exécutons A_i
- Exécutons les tâches de traitement aux premiers emplacements libres en respectant le graphe G_p



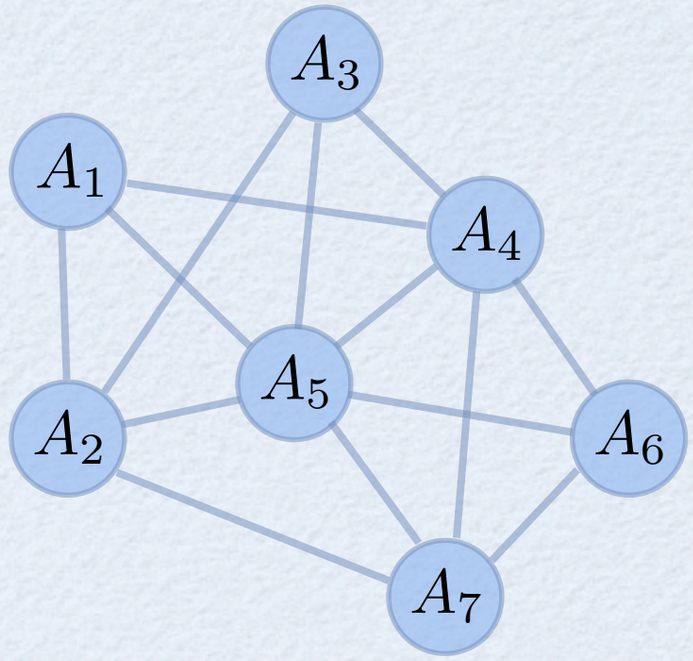
Deuxième Heuristique : Clique maximale

Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

Résultat : C^h_{max}

Début



G_c

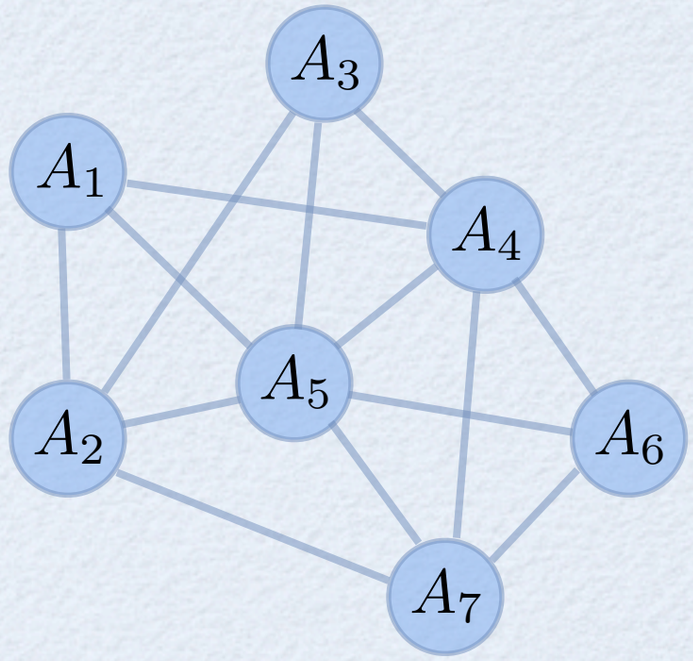
Deuxième Heuristique : Clique maximale

Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

Résultat : C^h_{max}

Début



G_c

Deuxième Heuristique : Clique maximale

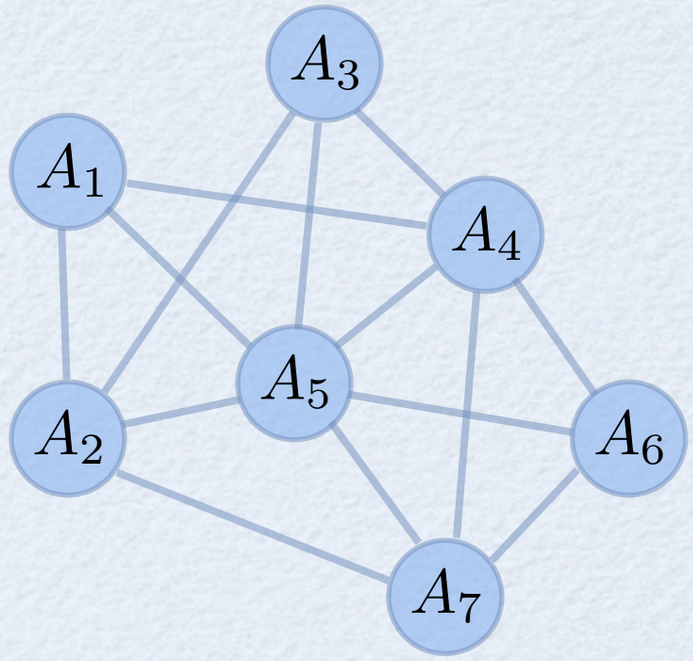
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

Résultat : C^h_{max}

Début

- Tant qu'il reste des sommets non visités dans G_c



G_c

Deuxième Heuristique : Clique maximale

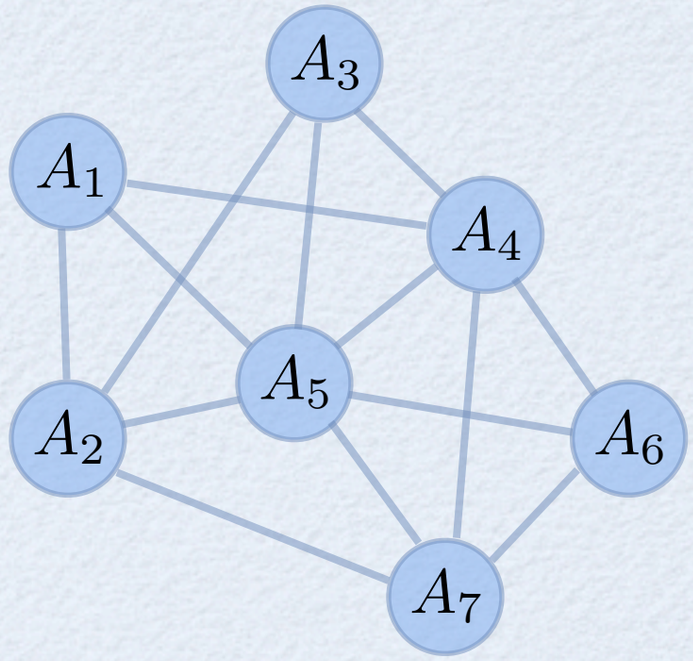
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

Résultat : C^h_{max}

Début

- Tant qu'il reste des sommets non visités dans G_c
- Prendre un sommet non visité A_i
- Chercher une clique maximale



G_c

Deuxième Heuristique : Clique maximale

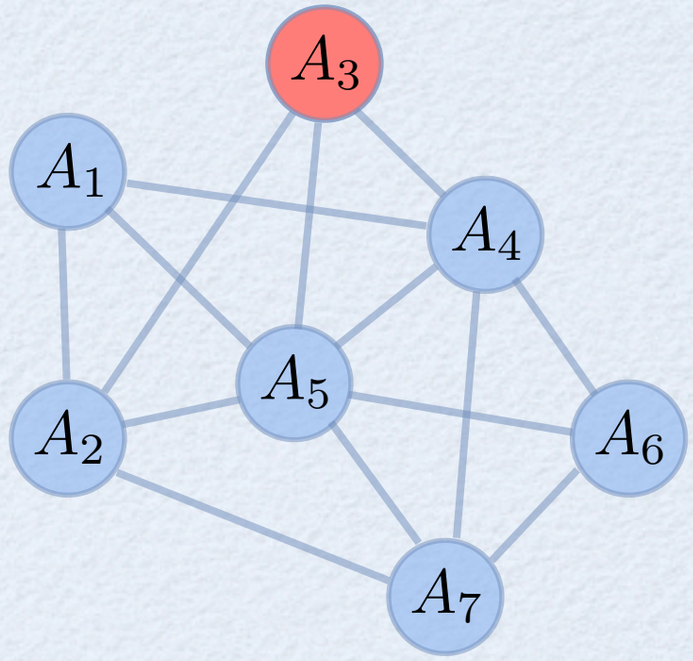
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

Résultat : C^h_{max}

Début

- Tant qu'il reste des sommets non visités dans G_c
- Prendre un sommet non visité A_i
- Chercher une clique maximale



G_c

Deuxième Heuristique : Clique maximale

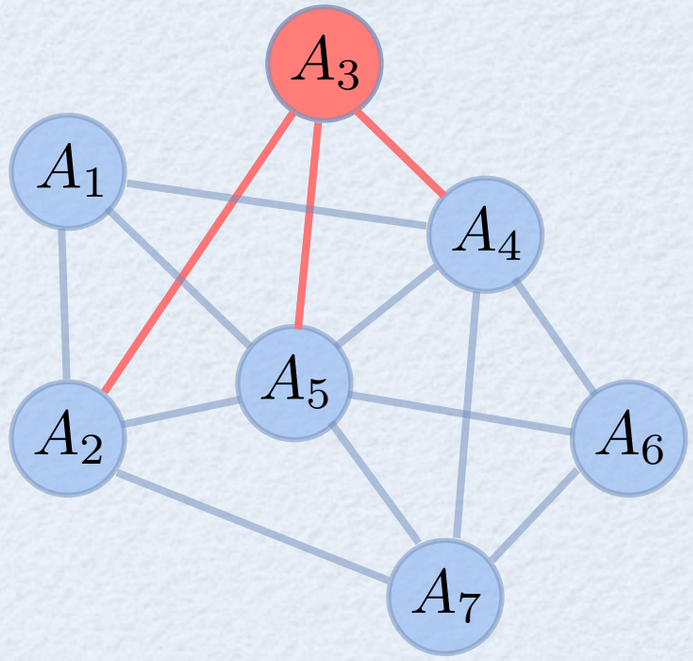
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

Résultat : C^h_{max}

Début

- Tant qu'il reste des sommets non visités dans G_c
- Prendre un sommet non visité A_i
- Chercher une clique maximale



G_c

Deuxième Heuristique : Clique maximale

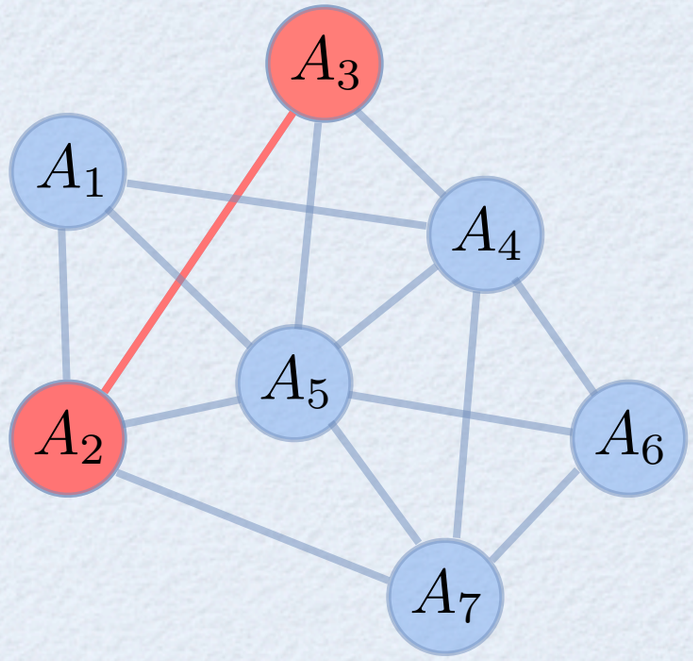
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

Résultat : C^h_{max}

Début

- Tant qu'il reste des sommets non visités dans G_c
- Prendre un sommet non visité A_i
- Chercher une clique maximale



G_c

Deuxième Heuristique : Clique maximale

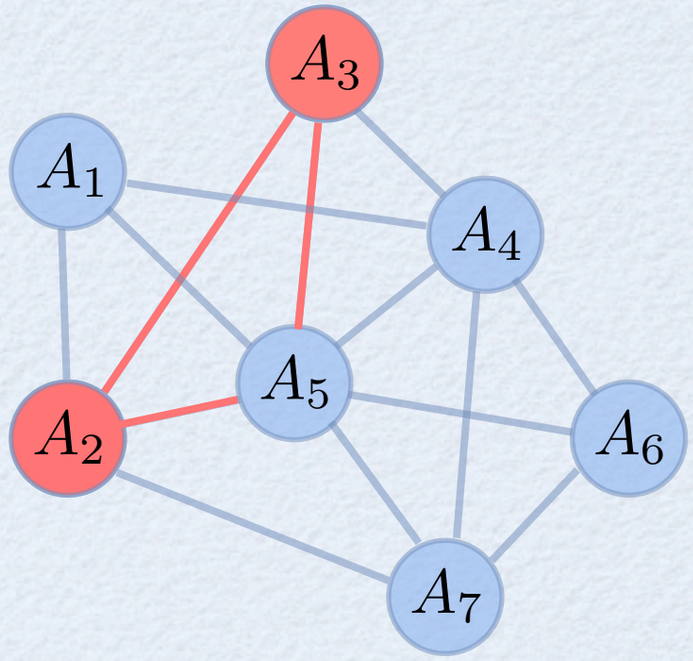
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

Résultat : C^h_{max}

Début

- Tant qu'il reste des sommets non visités dans G_c
- Prendre un sommet non visité A_i
- Chercher une clique maximale



G_c

Deuxième Heuristique : Clique maximale

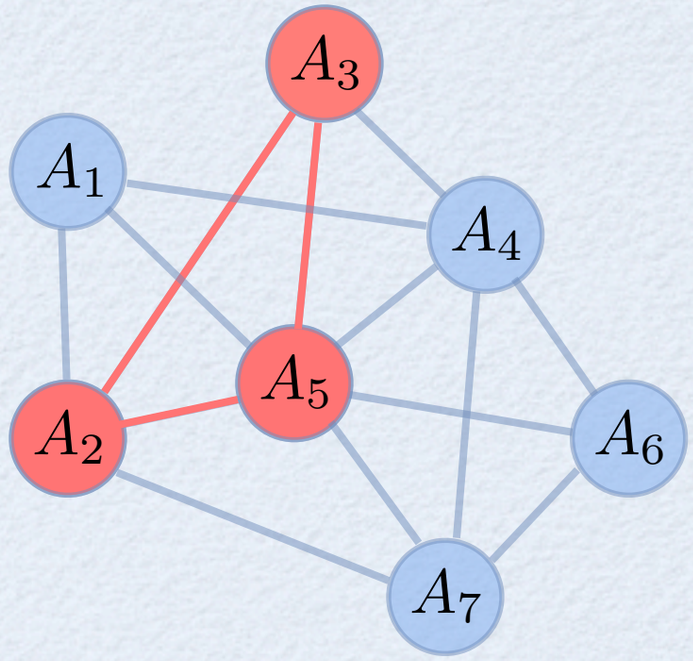
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

Résultat : C^h_{max}

Début

- Tant qu'il reste des sommets non visités dans G_c
- Prendre un sommet non visité A_i
- Chercher une clique maximale



G_c

Deuxième Heuristique : Clique maximale

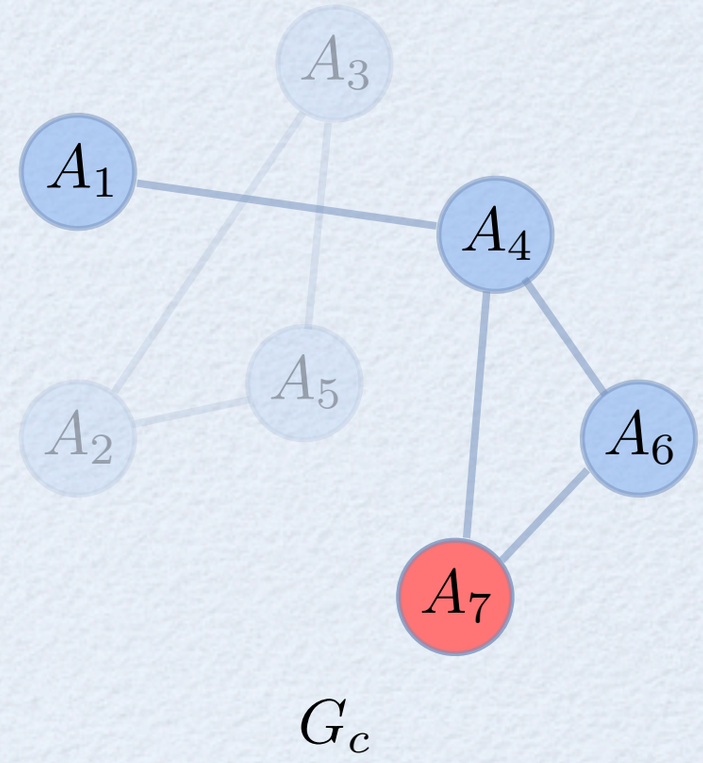
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

Résultat : C^h_{max}

Début

- Tant qu'il reste des sommets non visités dans G_c
- Prendre un sommet non visité A_i
- Chercher une clique maximale



G_c

Deuxième Heuristique : Clique maximale

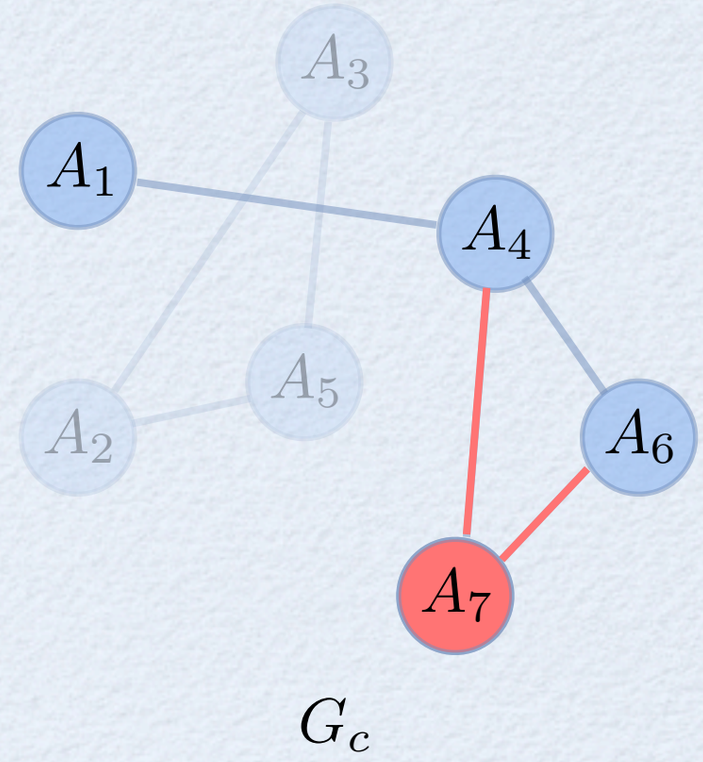
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

Résultat : C^h_{max}

Début

- Tant qu'il reste des sommets non visités dans G_c
- Prendre un sommet non visité A_i
- Chercher une clique maximale



G_c

Deuxième Heuristique : Clique maximale

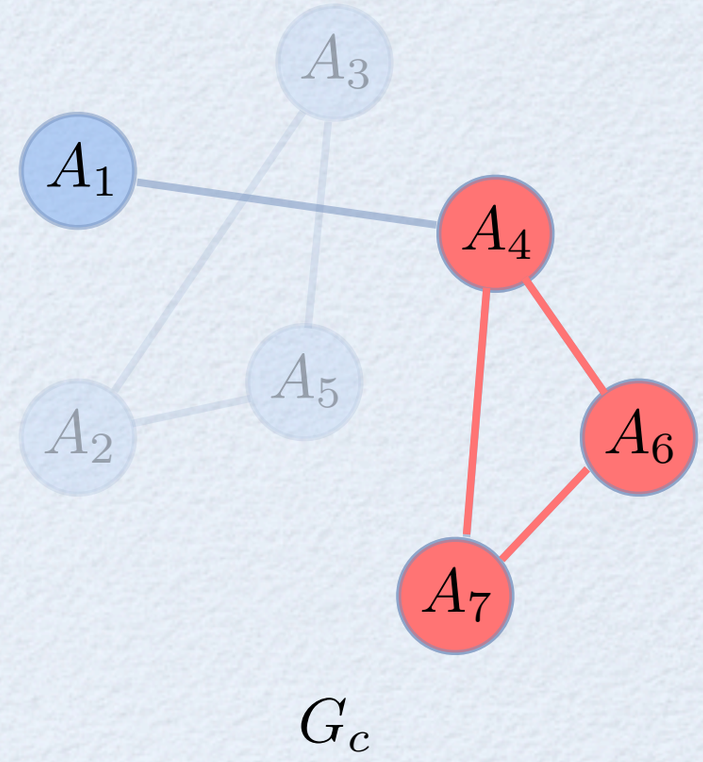
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

Résultat : C^h_{max}

Début

- Tant qu'il reste des sommets non visités dans G_c
- Prendre un sommet non visité A_i
- Chercher une clique maximale



G_c

Deuxième Heuristique : Clique maximale

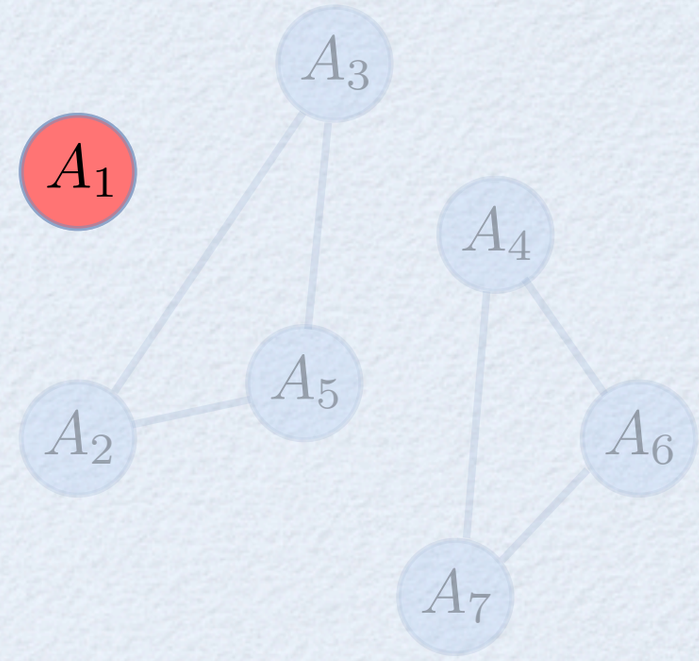
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

Résultat : C^h_{max}

Début

- Tant qu'il reste des sommets non visités dans G_c
- Prendre un sommet non visité A_i
- Chercher une clique maximale



G_c

Deuxième Heuristique : Clique maximale

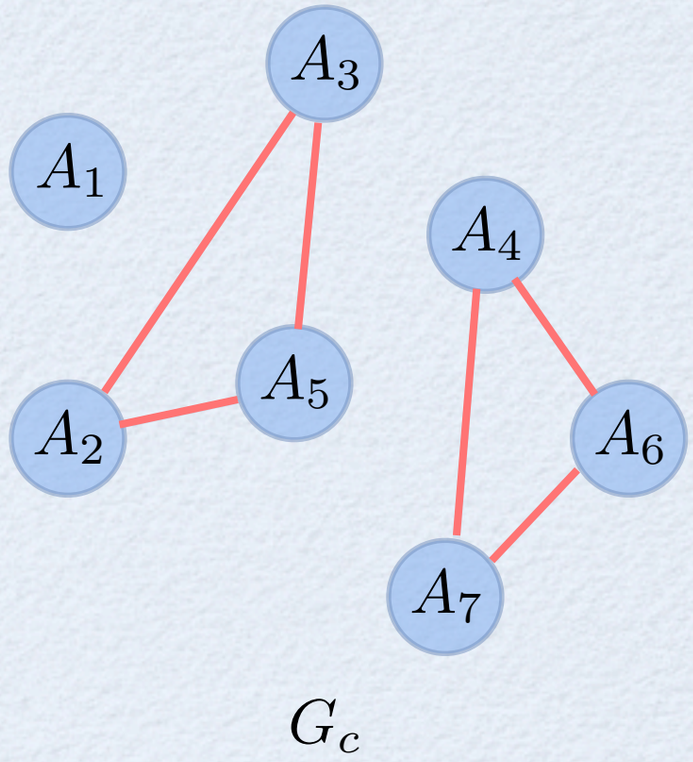
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

Résultat : C^h_{max}

Début

- Tant qu'il reste des sommets non visités dans G_c
  - Prendre un sommet non visité A_i
  - Chercher une clique maximale
- Ordonnancer les cliques par ordre décroissant



G_c

Deuxième Heuristique : Clique maximale

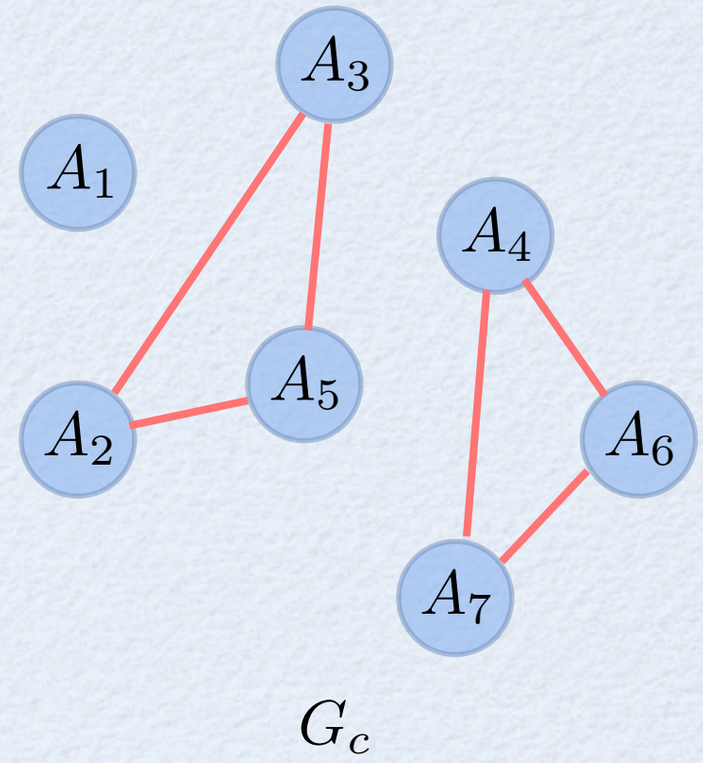
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

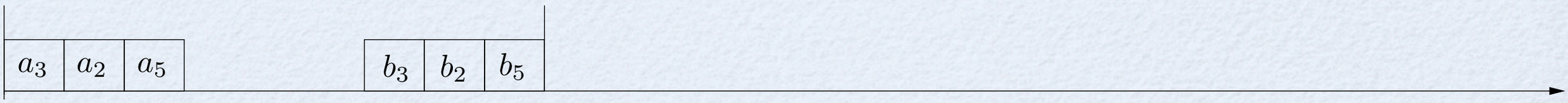
Résultat : C^h_{max}

Début

- Tant qu'il reste des sommets non visités dans G_c
  - Prendre un sommet non visité A_i
  - Chercher une clique maximale
- Ordonnancer les cliques par ordre décroissant



G_c



Deuxième Heuristique : Clique maximale

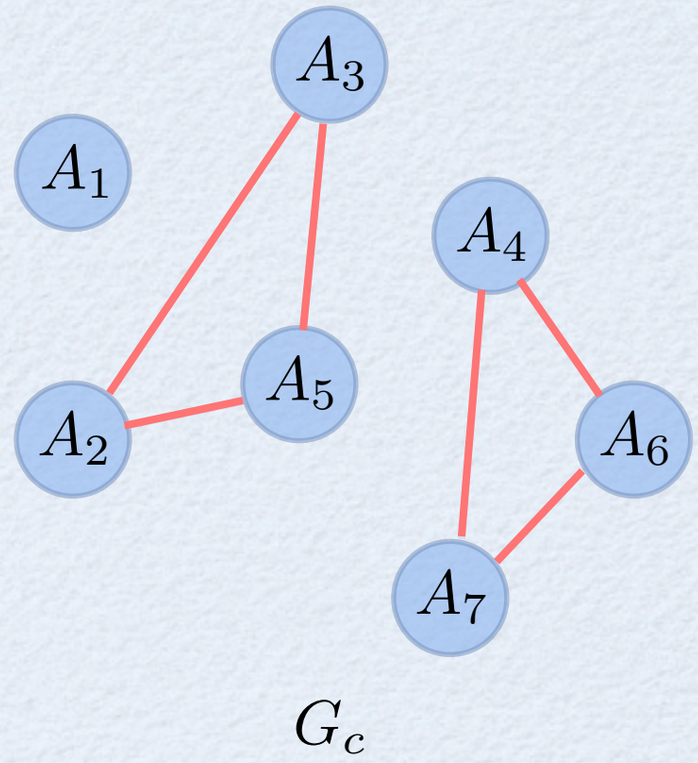
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

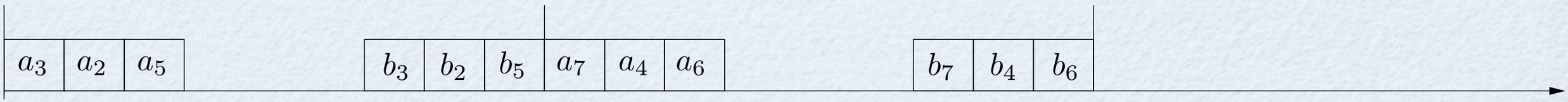
Résultat : C^h_{max}

Début

- Tant qu'il reste des sommets non visités dans G_c
  - Prendre un sommet non visité A_i
  - Chercher une clique maximale
- Ordonnancer les cliques par ordre décroissant



G_c



Deuxième Heuristique : Clique maximale

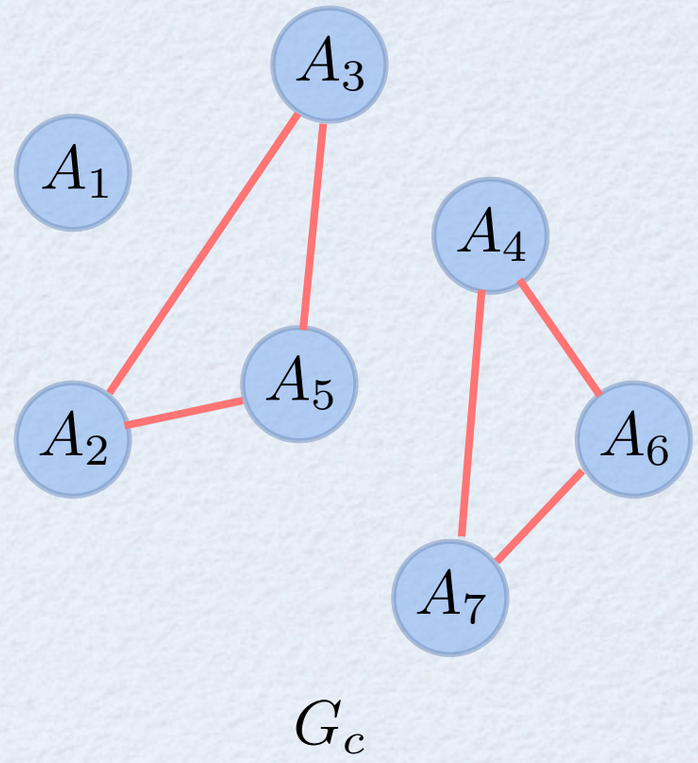
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

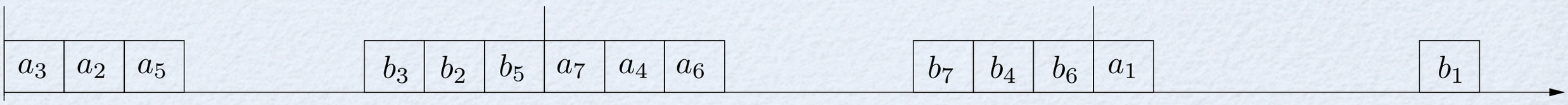
Résultat : C^h_{max}

Début

- Tant qu'il reste des sommets non visités dans G_c
  - Prendre un sommet non visité A_i
  - Chercher une clique maximale
- Ordonnancer les cliques par ordre décroissant



G_c



Deuxième Heuristique : Clique maximale

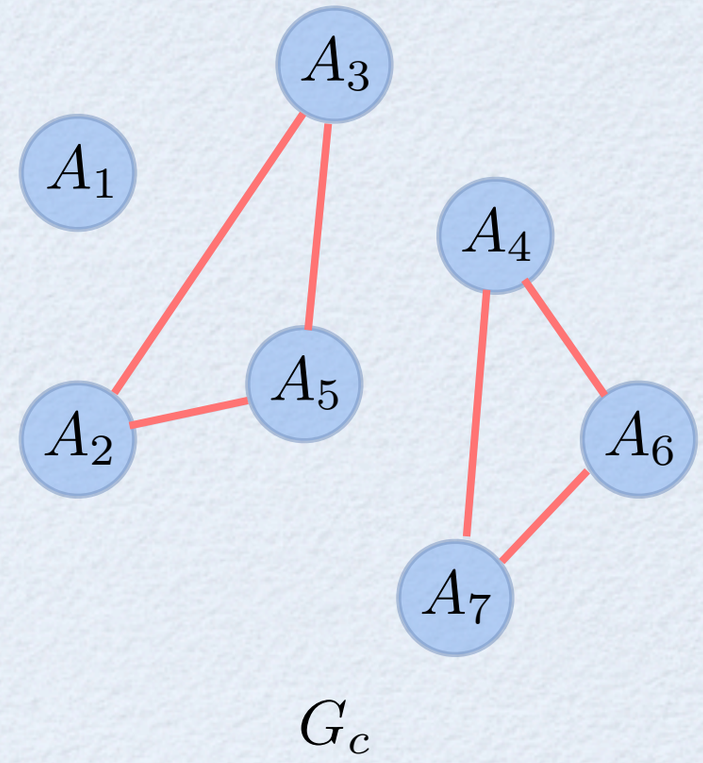
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c, G_p, \alpha \geq 1$

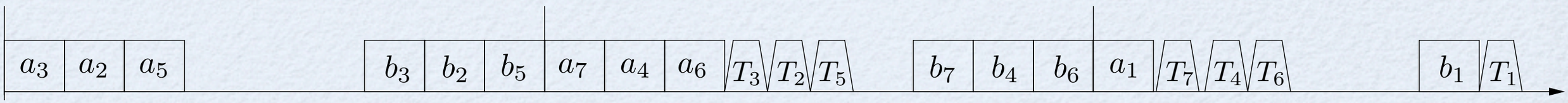
Résultat : C^h_{max}

Début

- Tant qu'il reste des sommets non visités dans G_c
  - Prendre un sommet non visité A_i
  - Chercher une clique maximale
- Ordonnancer les cliques par ordre décroissant
- Exécutons les tâches de traitement aux premiers emplacements libres en respectant le graphe G_p



G_c



Troisième Heuristique : Poupées russes

Troisième Heuristique : Poupées russes

Définition

Deux tâches d'acquisition A_i et A_j s'emboîtent l'une dans l'autre si et seulement si :

$$(a_i + L_i + b_i) < L_j \quad \text{ou} \quad (a_j + L_j + b_j) < L_i$$

Troisième Heuristique : Poupées russes

Définition

Deux tâches d'acquisition A_i et A_j s'emboîtent l'une dans l'autre si et seulement si :

$$(a_i + L_i + b_i) < L_j \quad \text{ou} \quad (a_j + L_j + b_j) < L_i$$

Exemple :

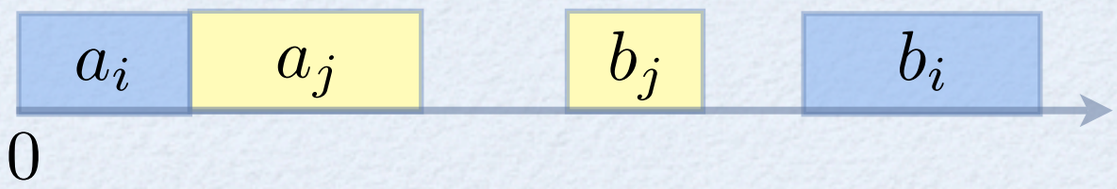
Troisième Heuristique : Poupées russes

Définition

Deux tâches d'acquisition A_i et A_j s'emboîtent l'une dans l'autre si et seulement si :

$$(a_i + L_i + b_i) < L_j \quad \text{ou} \quad (a_j + L_j + b_j) < L_i$$

Exemple :



Troisième Heuristique : Poupées russes

Définition

Deux tâches d'acquisition A_i et A_j s'emboîtent l'une dans l'autre si et seulement si :

$$(a_i + L_i + b_i) < L_j \quad \text{ou} \quad (a_j + L_j + b_j) < L_i$$

A partir de cette définition nous obtenons un nouveau graphe de compatibilité :

Définition

Nous notons G_c^m le graphe de compatibilité modifié où nous laissons les arêtes entre deux tâches d'acquisition qui s'emboîtent l'une dans l'autre.

Troisième Heuristique : Poupées russes

Définition

Deux tâches d'acquisition A_i et A_j s'emboîtent l'une dans l'autre si et seulement si :

$$(a_i + L_i + b_i) < L_j \quad \text{ou} \quad (a_j + L_j + b_j) < L_i$$

A partir de cette définition nous obtenons un nouveau graphe de compatibilité :

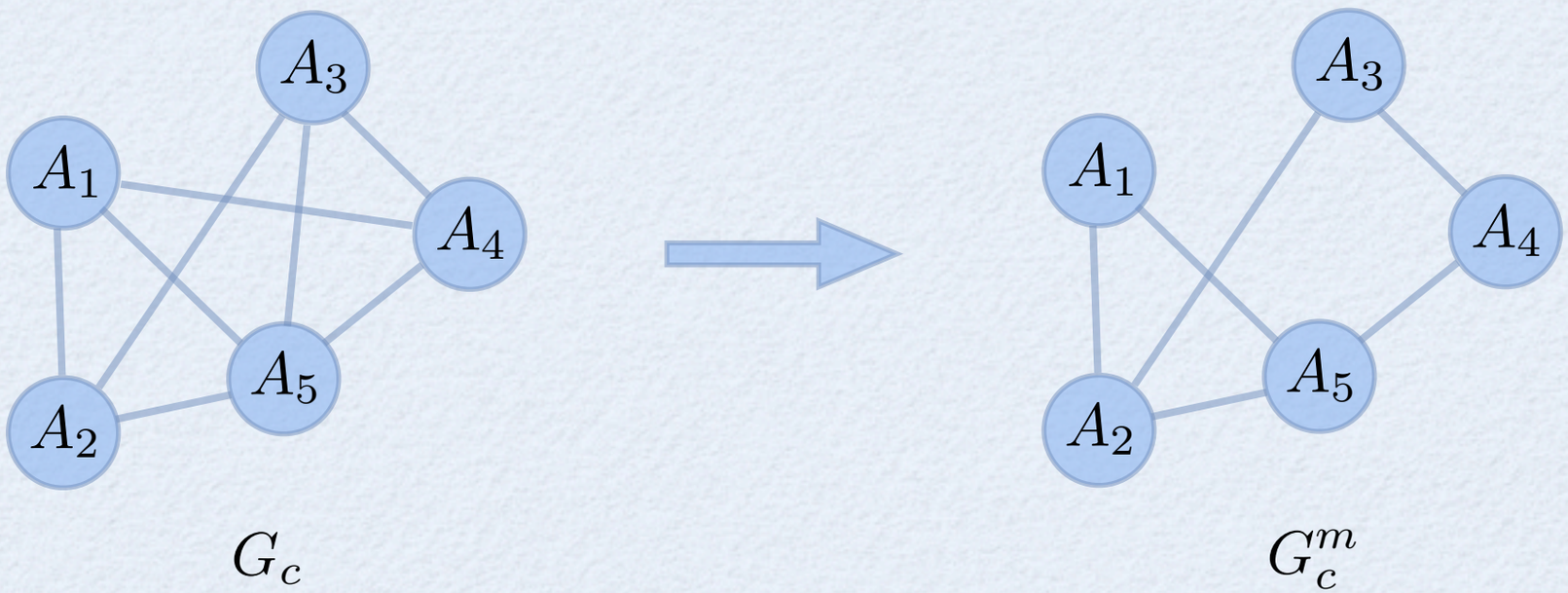
Définition

Nous notons G_c^m le graphe de compatibilité modifié où nous laissons les arêtes entre deux tâches d'acquisition qui s'emboîtent l'une dans l'autre.

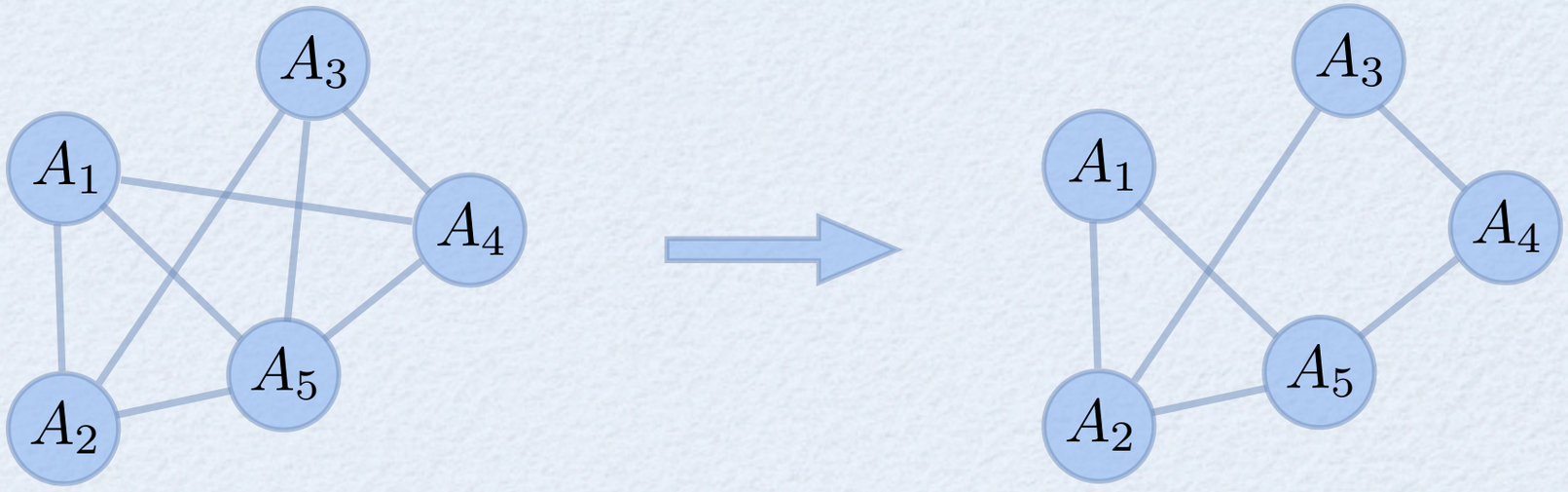
Troisième Heuristique : Poupées russes

Définition

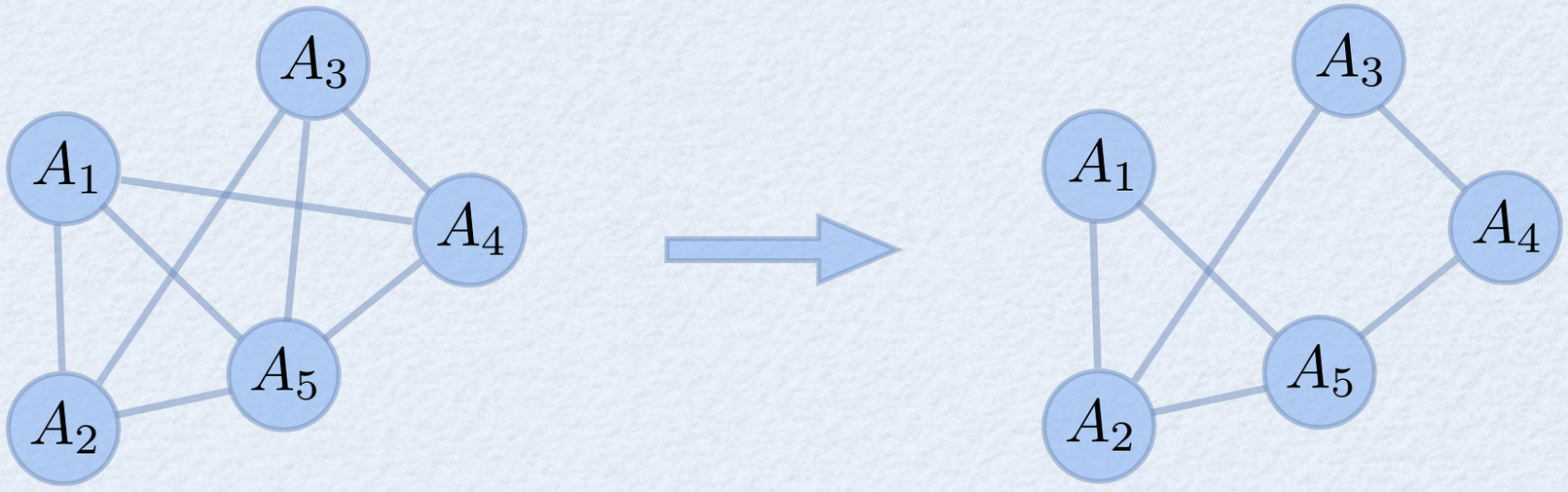
Nous notons G_c^m le graphe de compatibilité modifié où nous laissons les arêtes entre deux tâches d'acquisition qui s'emboîtent l'une dans l'autre.



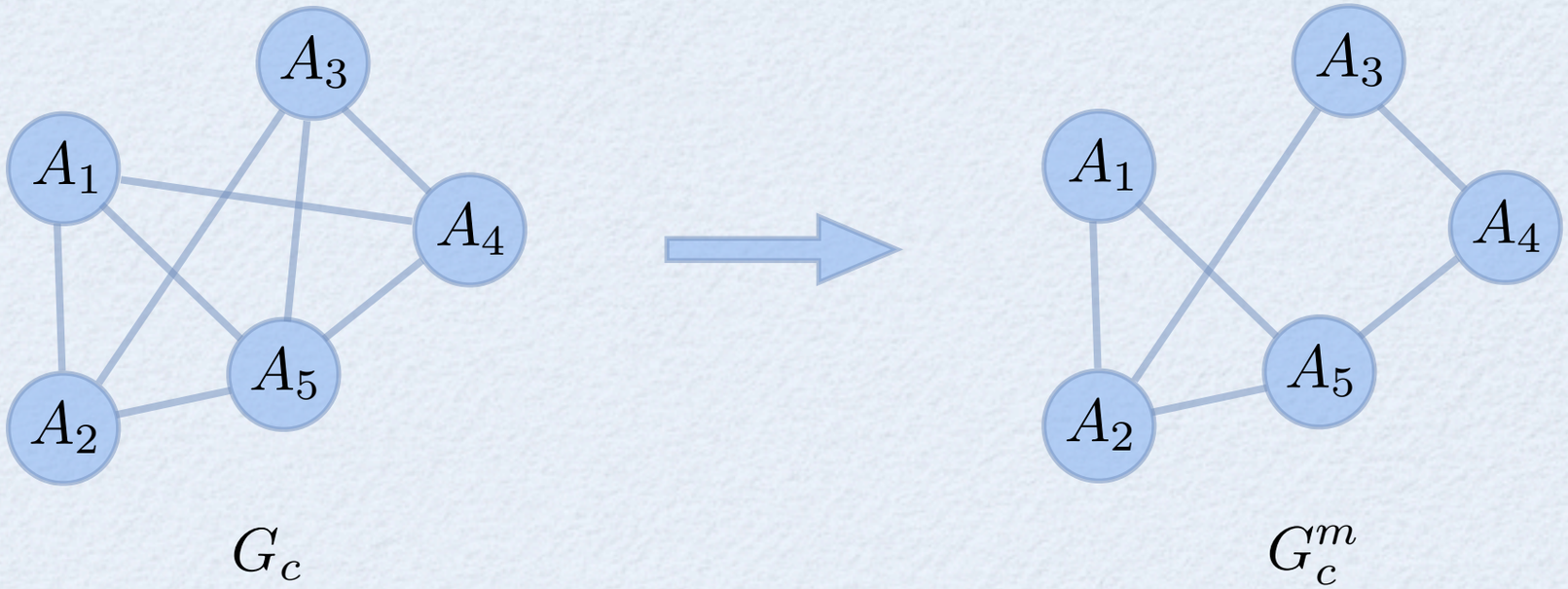
Troisième Heuristique : Poupées russes

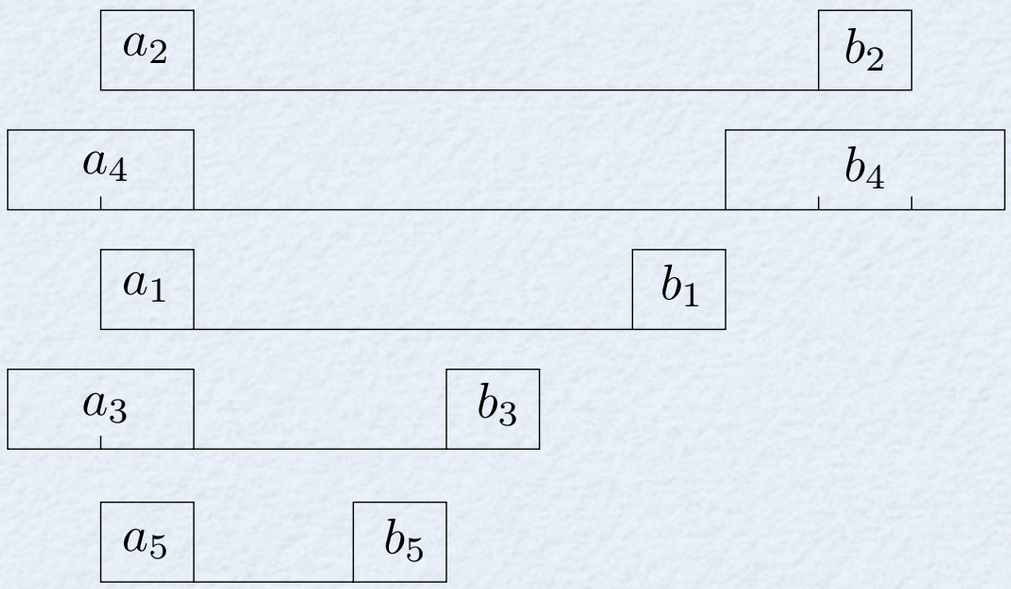


Troisième Heuristique : Poupées russes

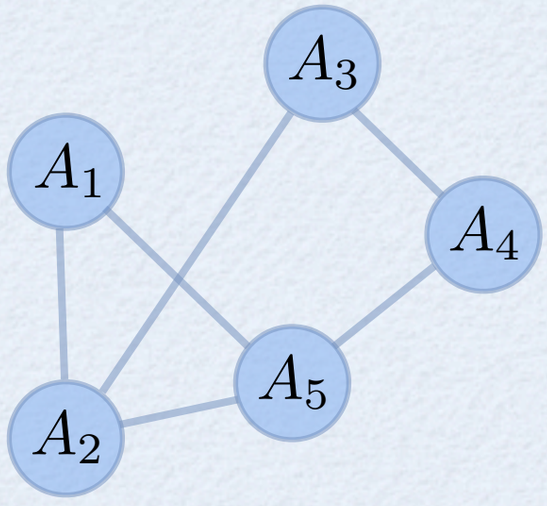


Troisième Heuristique : Poupées russes

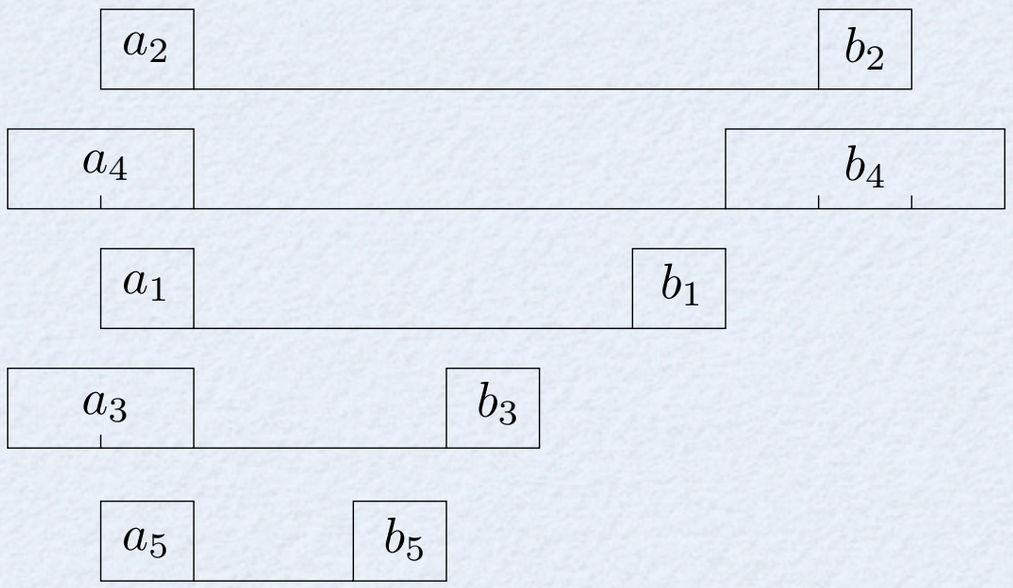




Troisième Heuristique : Poupées russes



G_c^m



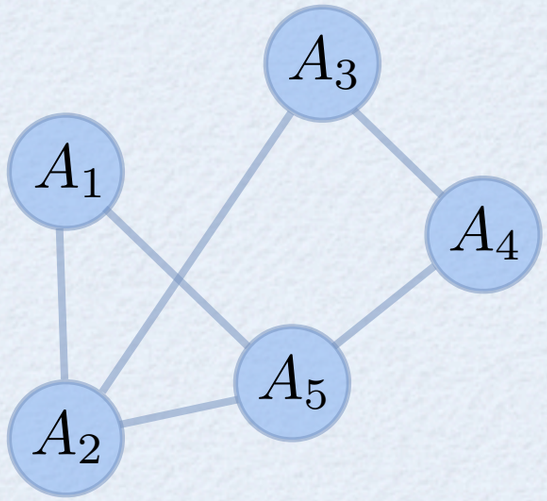
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c^m, G_p, \alpha \geq 1$

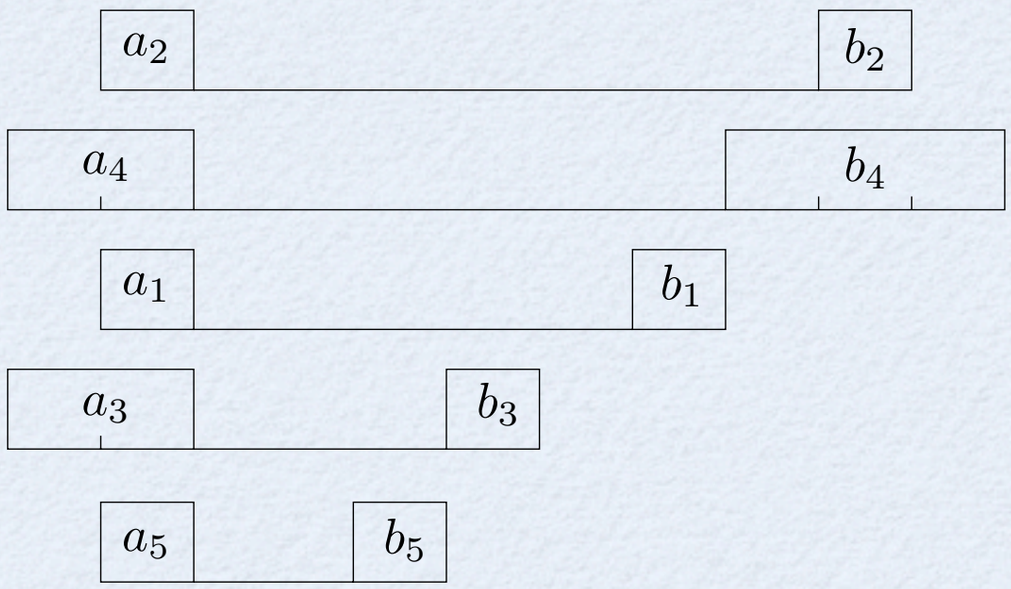
Résultat : C_{max}^h

Début

Troisième Heuristique : Poupées russes



G_c^m



Algorithme en temps polynomial :

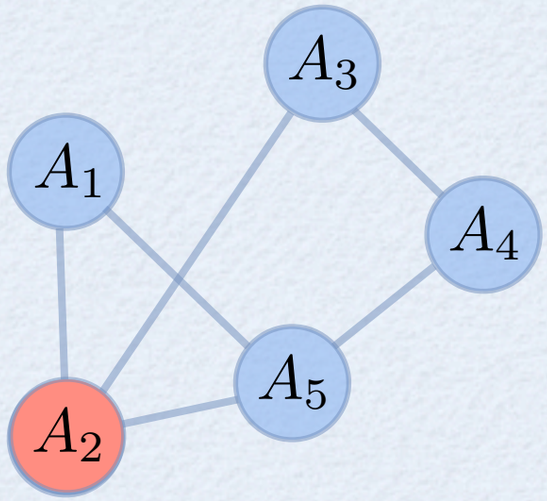
Données : $\mathcal{A}, \mathcal{T}, G_c^m, G_p, \alpha \geq 1$

Résultat : C_{max}^h

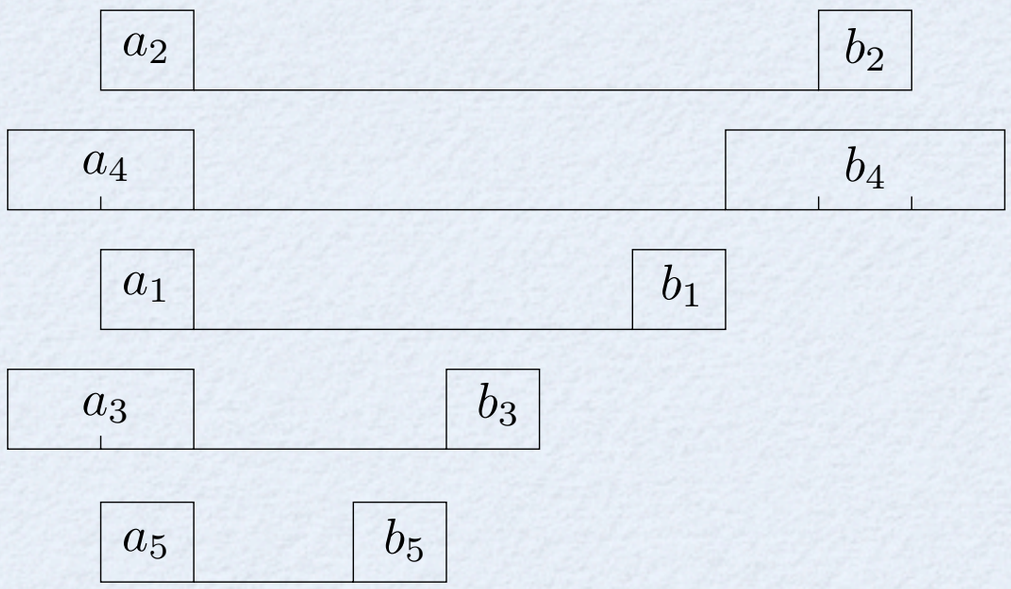
Début

- Tant qu'il reste des sommets non visités

Troisième Heuristique : Poupées russes



G_c^m



Algorithme en temps polynomial :

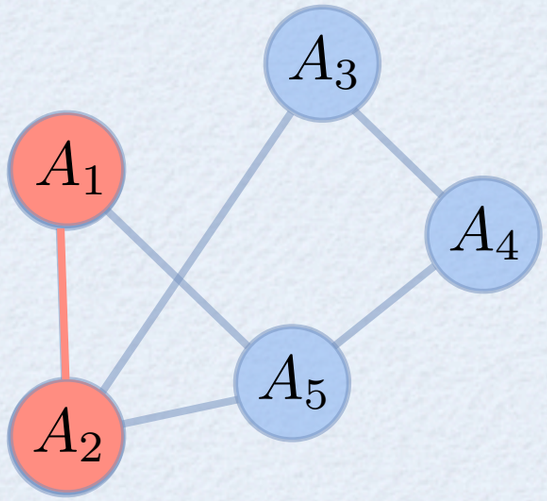
Données : $\mathcal{A}, \mathcal{T}, G_c^m, G_p, \alpha \geq 1$

Résultat : C_{max}^h

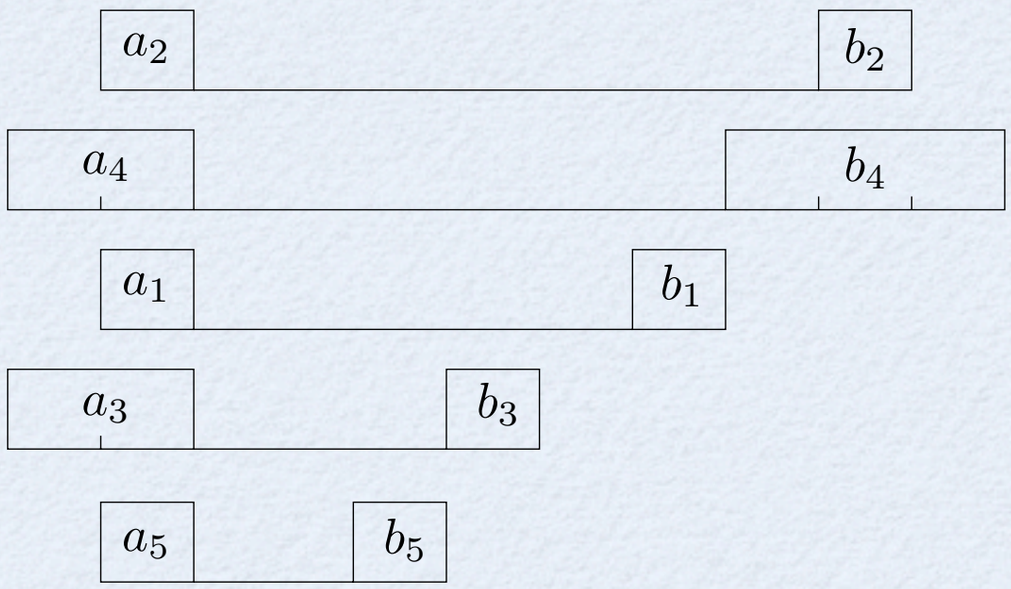
Début

- Tant qu'il reste des sommets non visités
  - Prendre un sommet non visité A_i
  - Chercher une clique maximale

Troisième Heuristique : Poupées russes



G_c^m



Algorithme en temps polynomial :

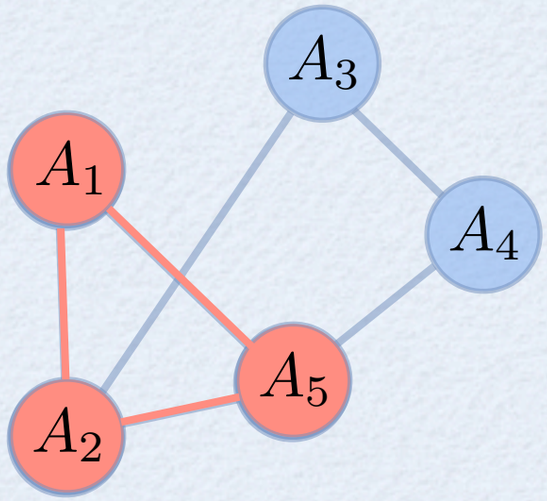
Données : $\mathcal{A}, \mathcal{T}, G_c^m, G_p, \alpha \geq 1$

Résultat : C_{max}^h

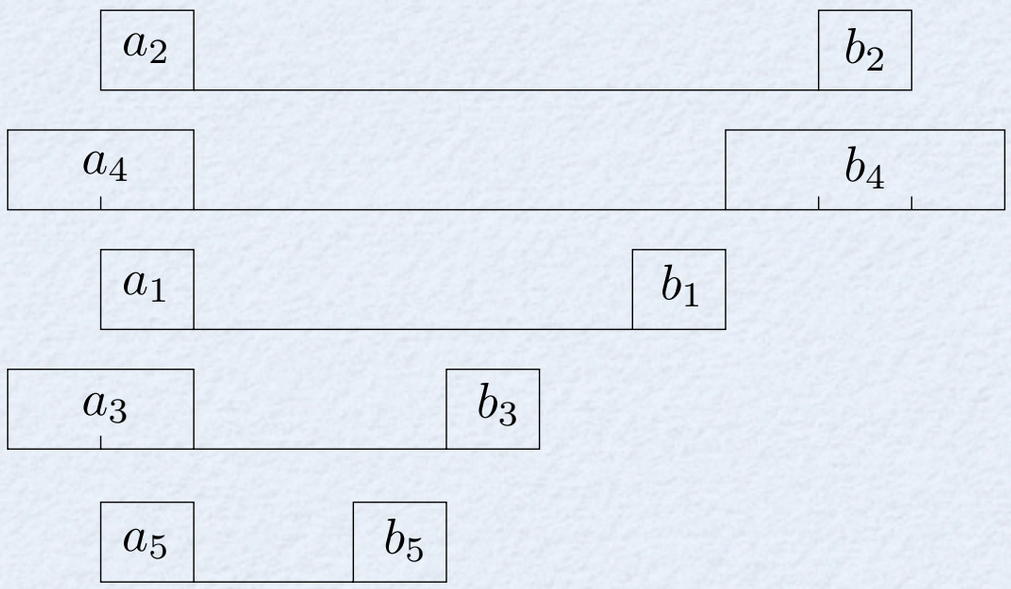
Début

- Tant qu'il reste des sommets non visités
  - Prendre un sommet non visité A_i
  - Chercher une clique maximale

Troisième Heuristique : Poupées russes



G_c^m



Algorithme en temps polynomial :

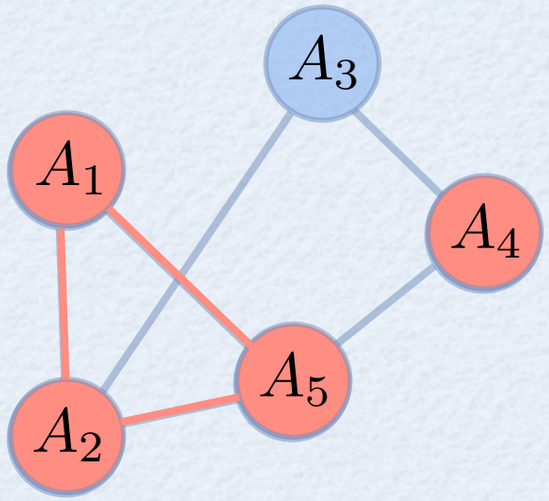
Données : $\mathcal{A}, \mathcal{T}, G_c^m, G_p, \alpha \geq 1$

Résultat : C_{max}^h

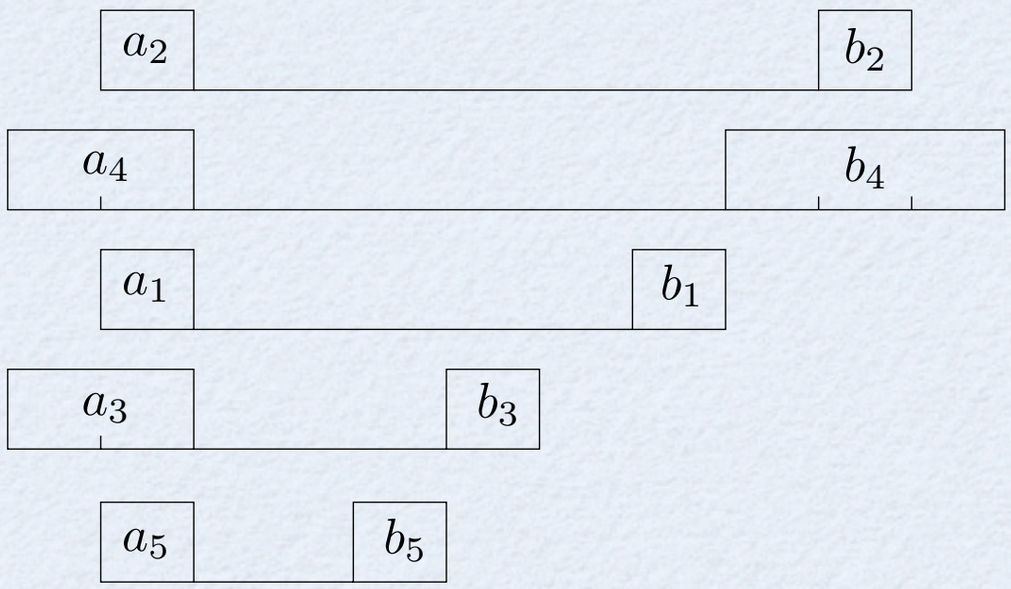
Début

- Tant qu'il reste des sommets non visités
  - Prendre un sommet non visité A_i
  - Chercher une clique maximale

Troisième Heuristique : Poupées russes



G_c^m



Algorithme en temps polynomial :

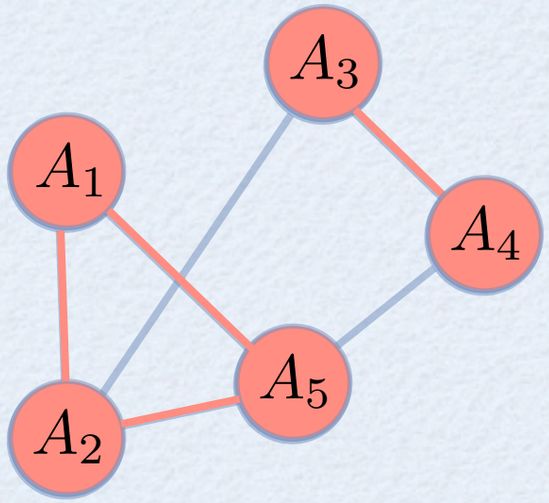
Données : $\mathcal{A}, \mathcal{T}, G_c^m, G_p, \alpha \geq 1$

Résultat : C_{max}^h

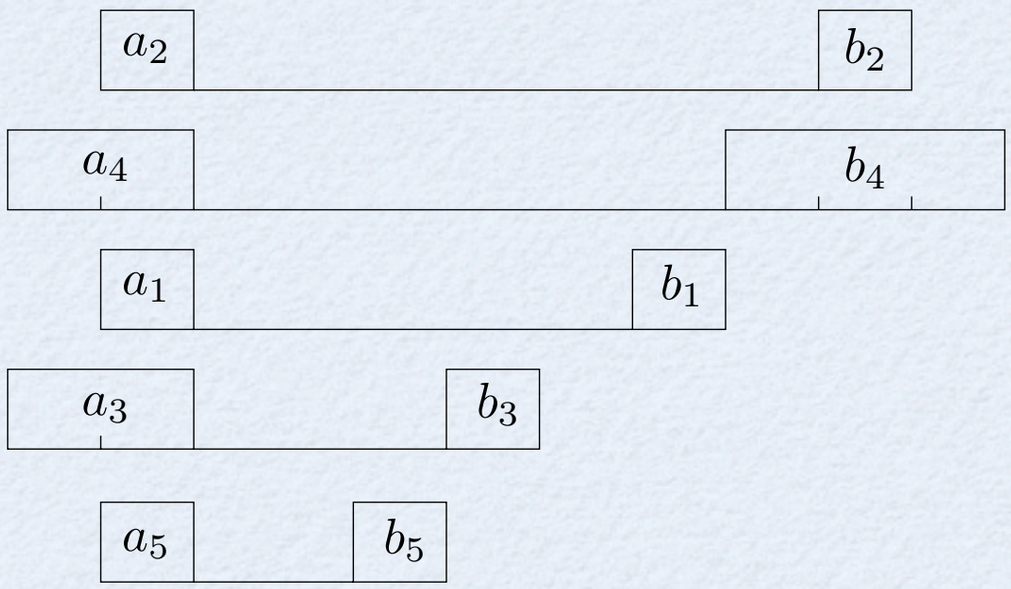
Début

- Tant qu'il reste des sommets non visités
  - Prendre un sommet non visité A_i
  - Chercher une clique maximale

Troisième Heuristique : Poupées russes



G_c^m



Algorithme en temps polynomial :

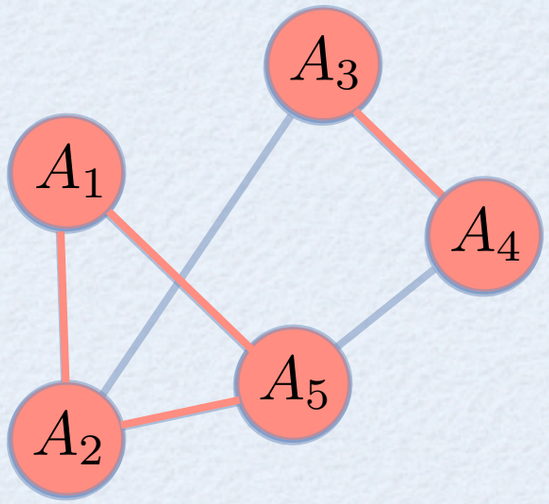
Données : $\mathcal{A}, \mathcal{T}, G_c^m, G_p, \alpha \geq 1$

Résultat : C_{max}^h

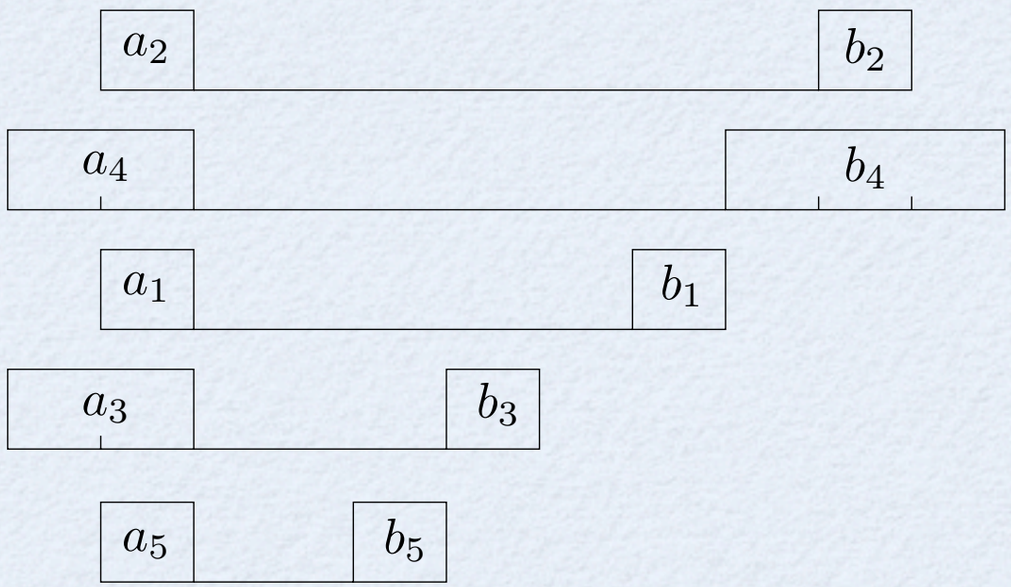
Début

- Tant qu'il reste des sommets non visités
  - Prendre un sommet non visité A_i
  - Chercher une clique maximale

Troisième Heuristique : Poupées russes



G_c^m



Algorithme en temps polynomial :

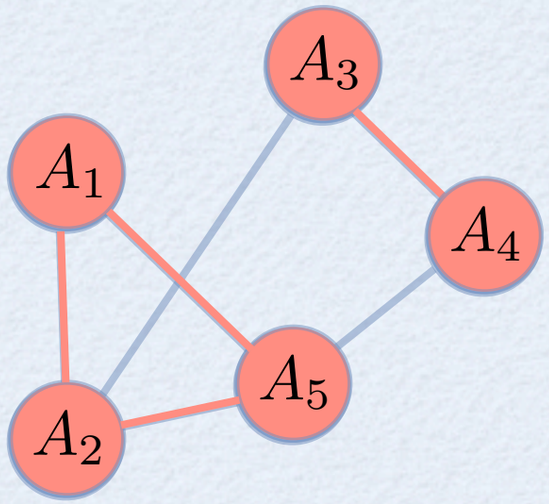
Données : $\mathcal{A}, \mathcal{T}, G_c^m, G_p, \alpha \geq 1$

Résultat : C_{max}^h

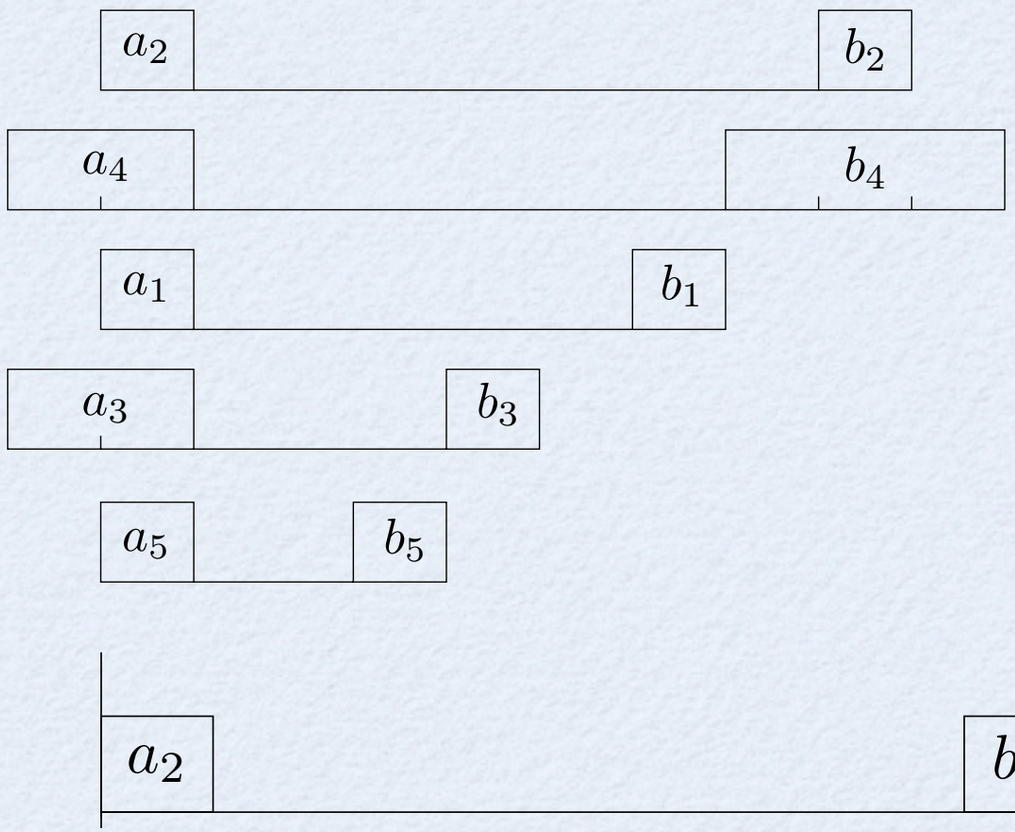
Début

- Tant qu'il reste des sommets non visités
  - Prendre un sommet non visité A_i
  - Chercher une clique maximale
- Ordonnancer les cliques selon l'ordre le plus judicieux

Troisième Heuristique : Poupées russes



G_c^m



Algorithme en temps polynomial :

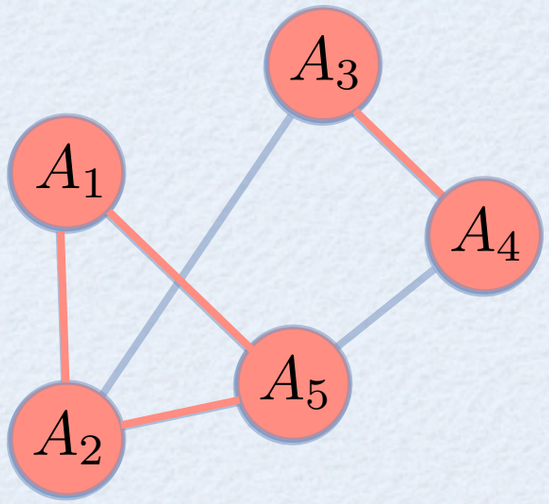
Données : $\mathcal{A}, \mathcal{T}, G_c^m, G_p, \alpha \geq 1$

Résultat : C_{max}^h

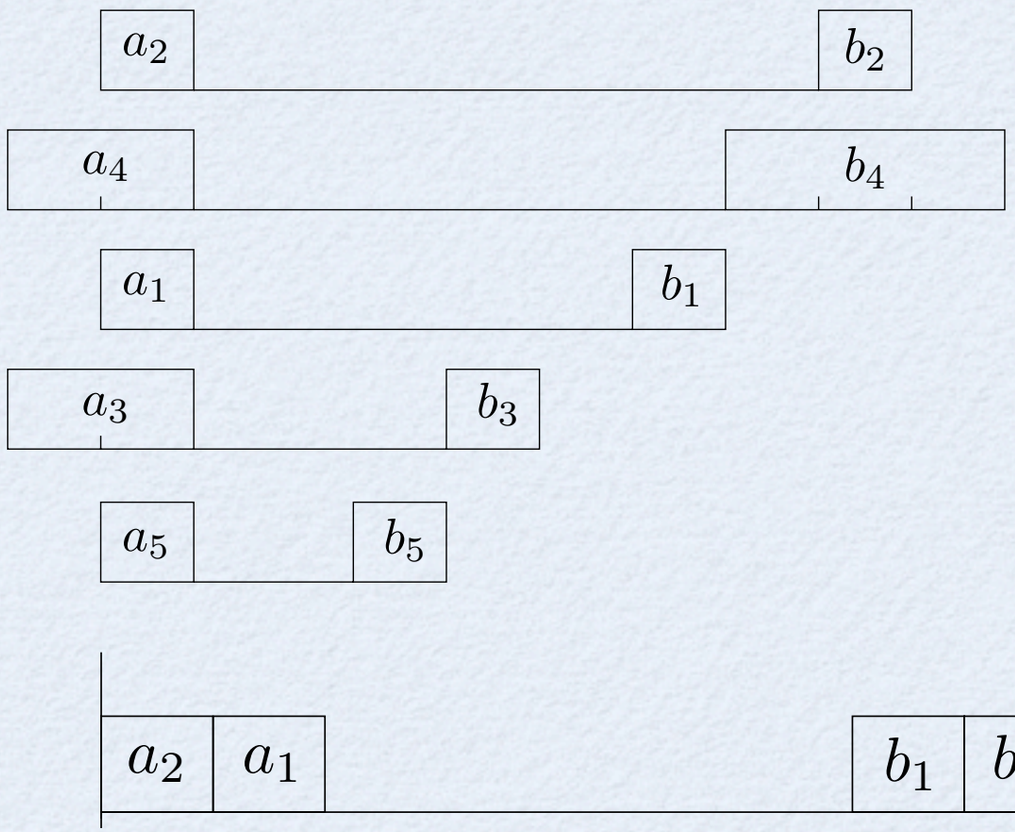
Début

- Tant qu'il reste des sommets non visités
  - Prendre un sommet non visité A_i
  - Chercher une clique maximale
- Ordonnancer les cliques selon l'ordre le plus judicieux

Troisième Heuristique : Poupées russes



G_c^m



Algorithme en temps polynomial :

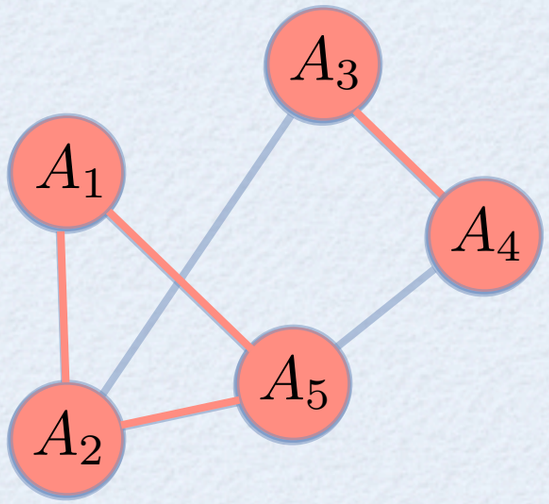
Données : $\mathcal{A}, \mathcal{T}, G_c^m, G_p, \alpha \geq 1$

Résultat : C_{max}^h

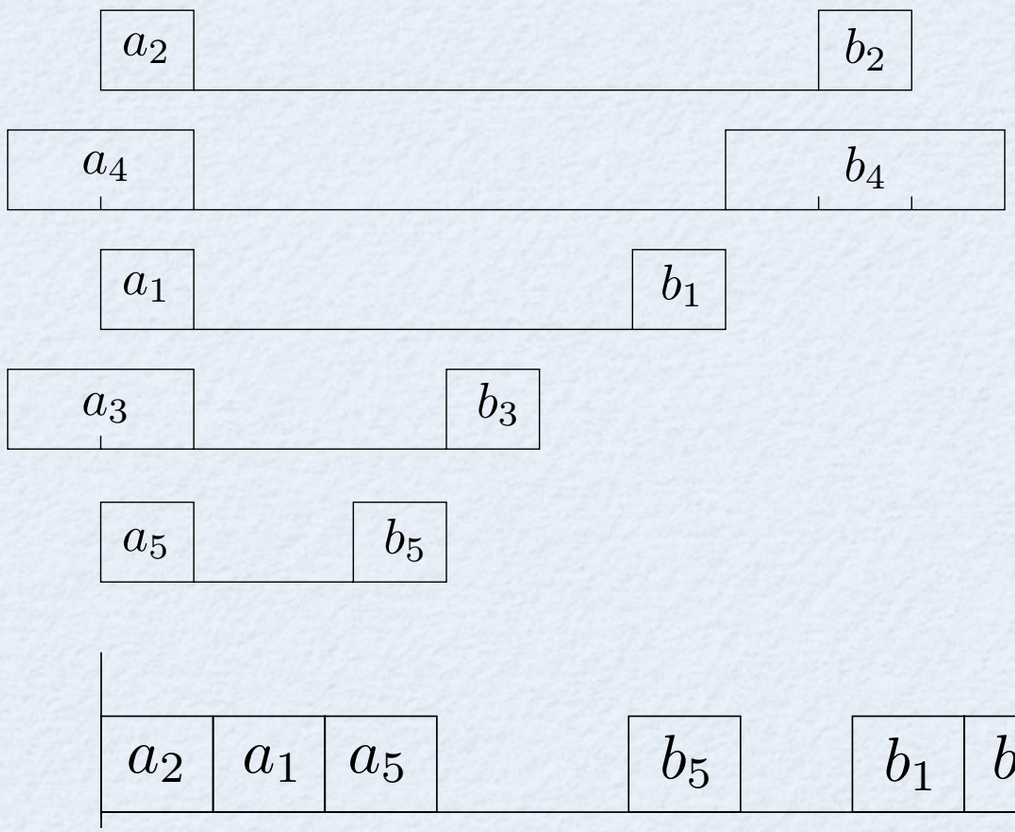
Début

- Tant qu'il reste des sommets non visités
  - Prendre un sommet non visité A_i
  - Chercher une clique maximale
- Ordonnancer les cliques selon l'ordre le plus judicieux

Troisième Heuristique : Poupées russes



G_c^m



Algorithme en temps polynomial :

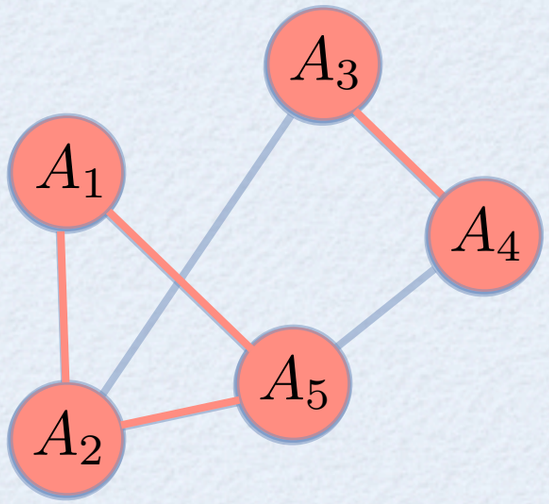
Données : $\mathcal{A}, \mathcal{T}, G_c^m, G_p, \alpha \geq 1$

Résultat : C_{max}^h

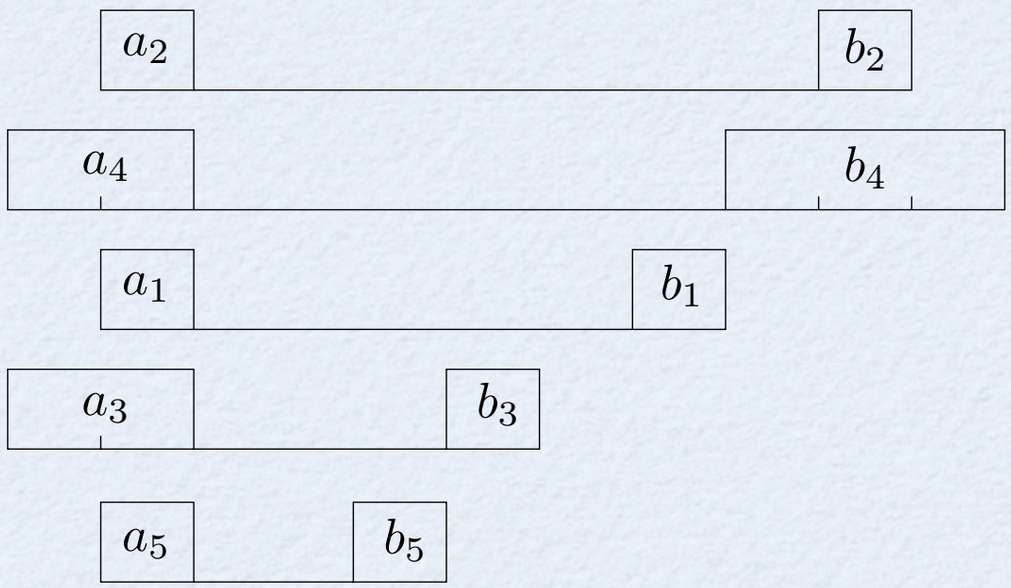
Début

- Tant qu'il reste des sommets non visités
  - Prendre un sommet non visité A_i
  - Chercher une clique maximale
- Ordonnancer les cliques selon l'ordre le plus judicieux

Troisième Heuristique : Poupées russes



G_c^m



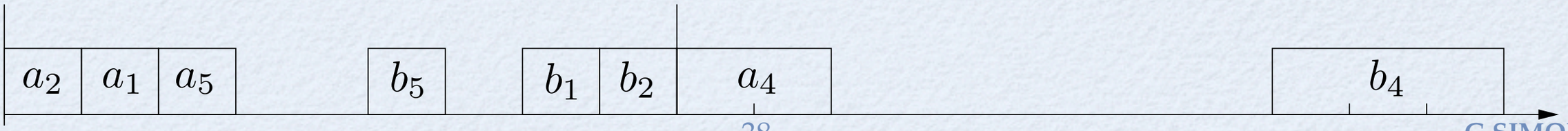
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c^m, G_p, \alpha \geq 1$

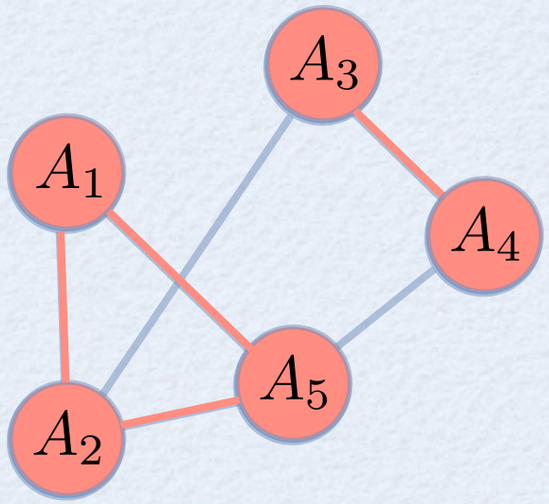
Résultat : C_{max}^h

Début

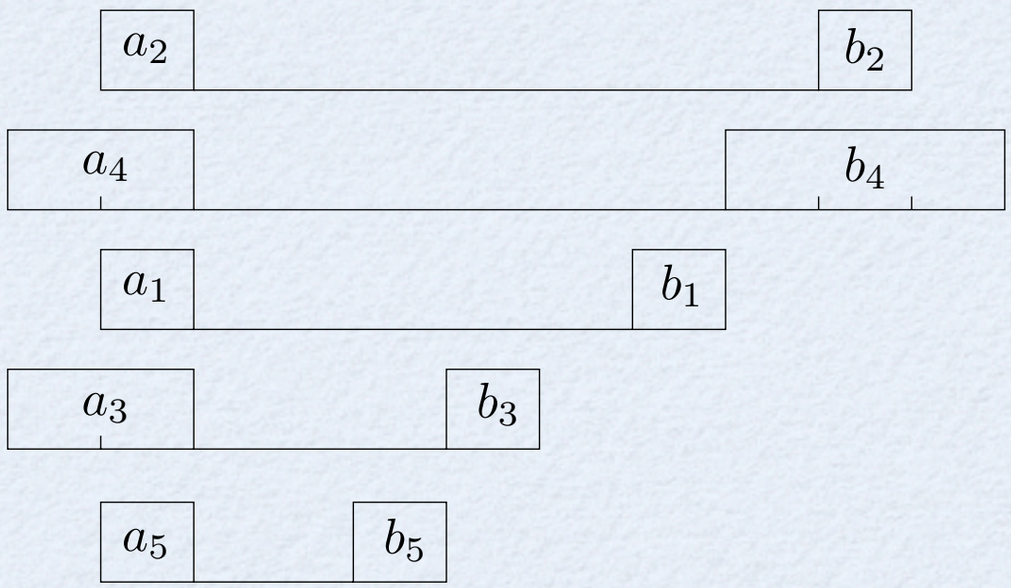
- Tant qu'il reste des sommets non visités
  - Prendre un sommet non visité A_i
  - Chercher une clique maximale
- Ordonnancer les cliques selon l'ordre le plus judicieux



Troisième Heuristique : Poupées russes



G_c^m



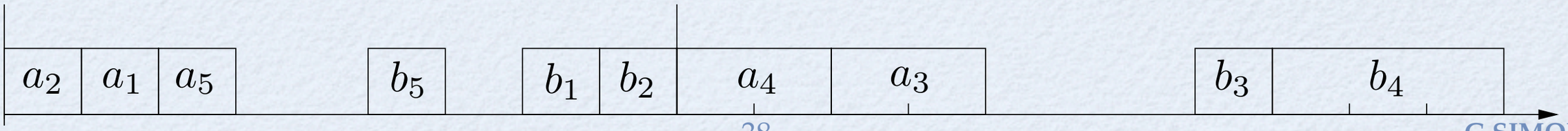
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c^m, G_p, \alpha \geq 1$

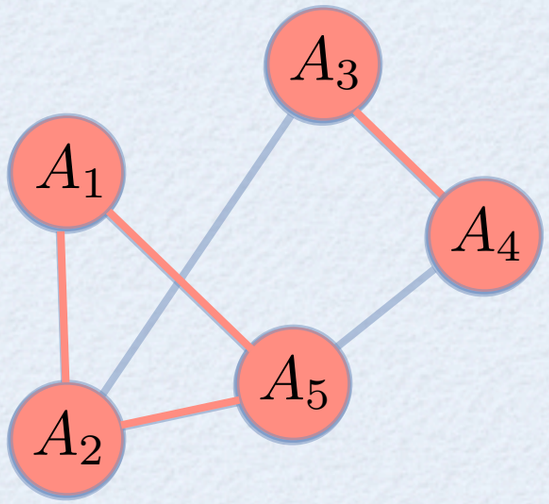
Résultat : C_{max}^h

Début

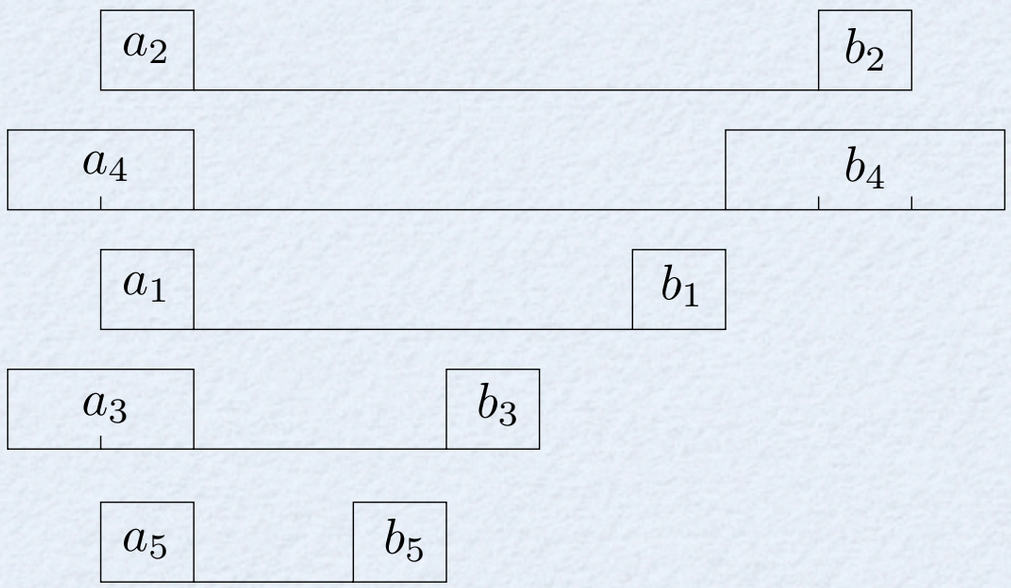
- Tant qu'il reste des sommets non visités
  - Prendre un sommet non visité A_i
  - Chercher une clique maximale
- Ordonnancer les cliques selon l'ordre le plus judicieux



Troisième Heuristique : Poupées russes



G_c^m



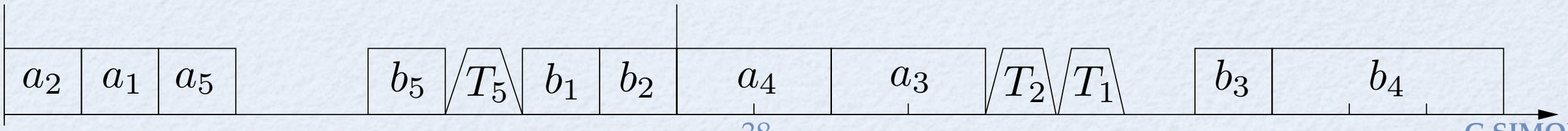
Algorithme en temps polynomial :

Données : $\mathcal{A}, \mathcal{T}, G_c^m, G_p, \alpha \geq 1$

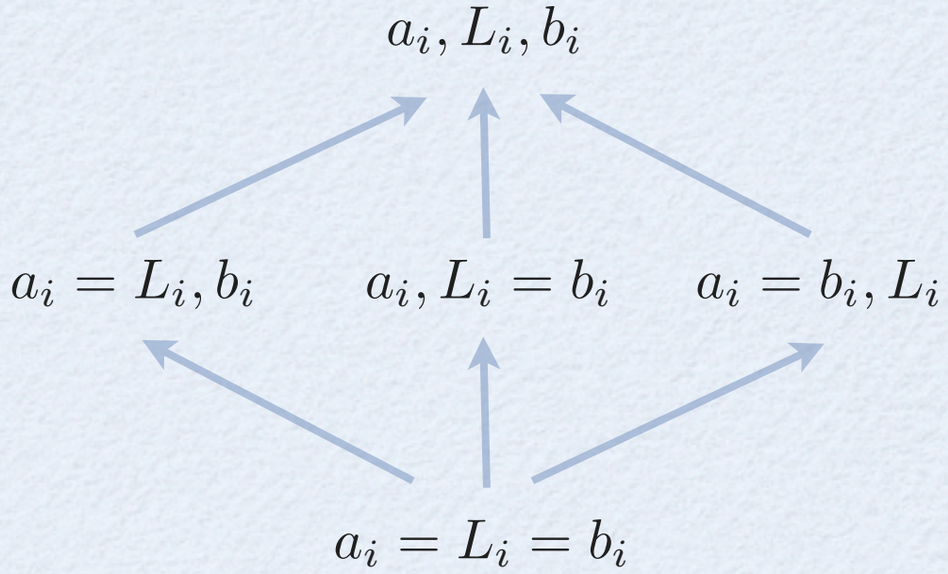
Résultat : C_{max}^h

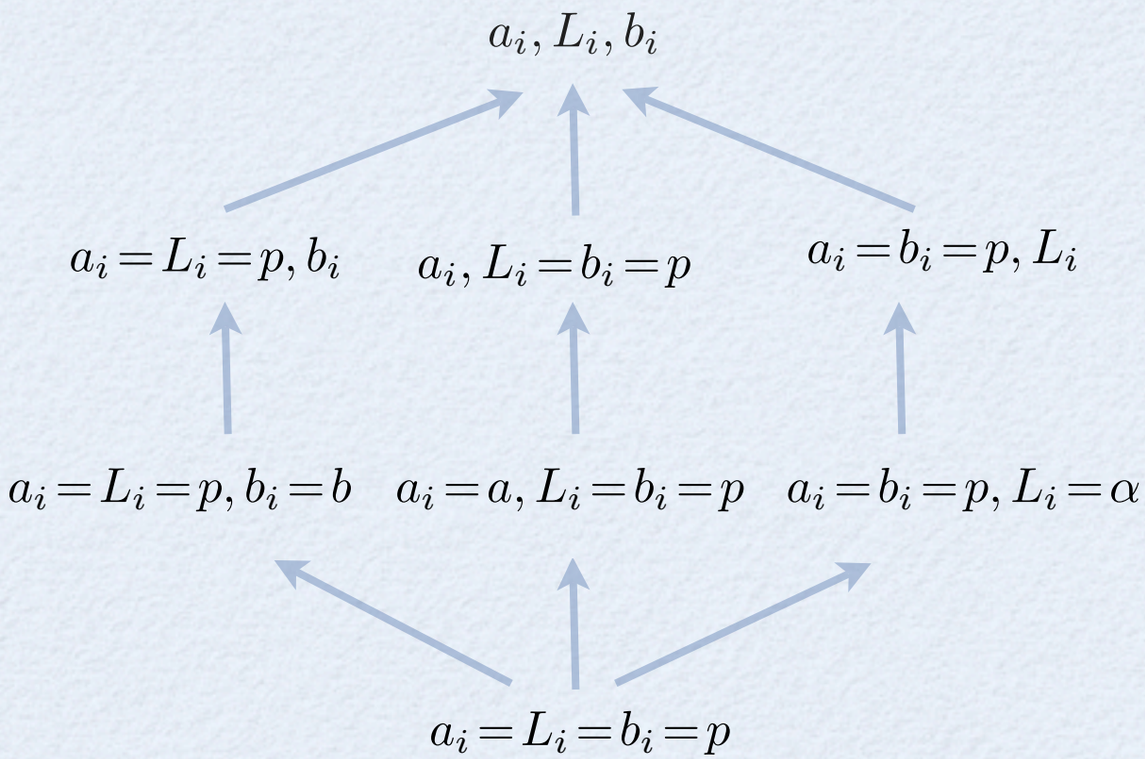
Début

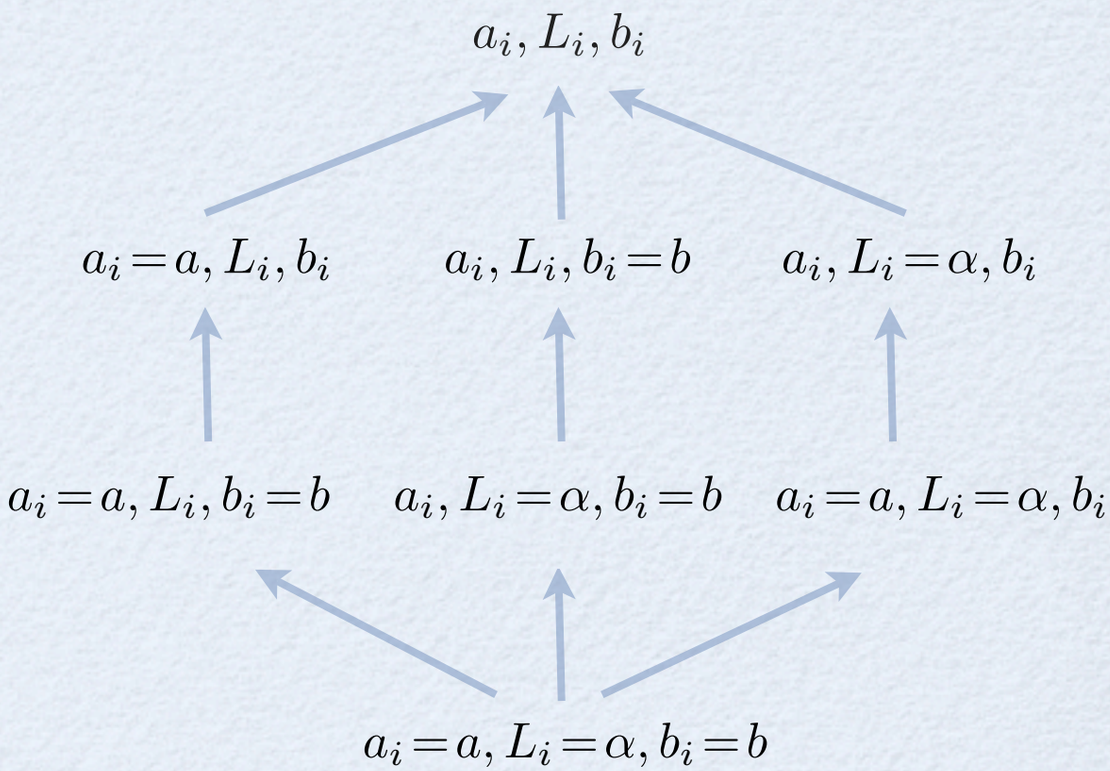
- Tant qu'il reste des sommets non visités
  - Prendre un sommet non visité A_i
  - Chercher une clique maximale
- Ordonnancer les cliques selon l'ordre le plus judicieux
- Exécutons les tâches de traitement aux premiers emplacements libres



Classification des heuristiques

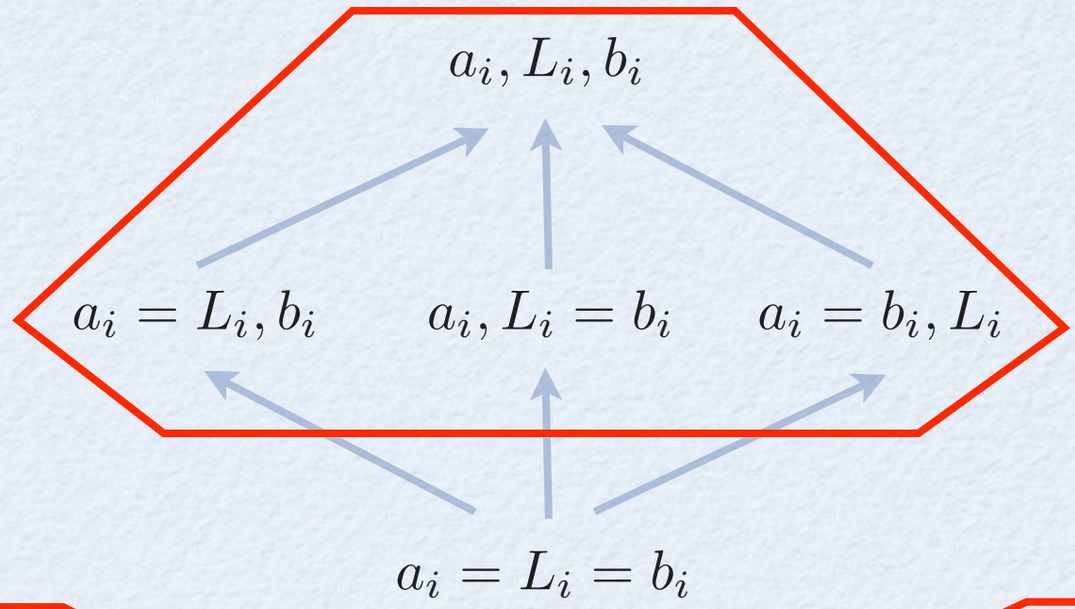


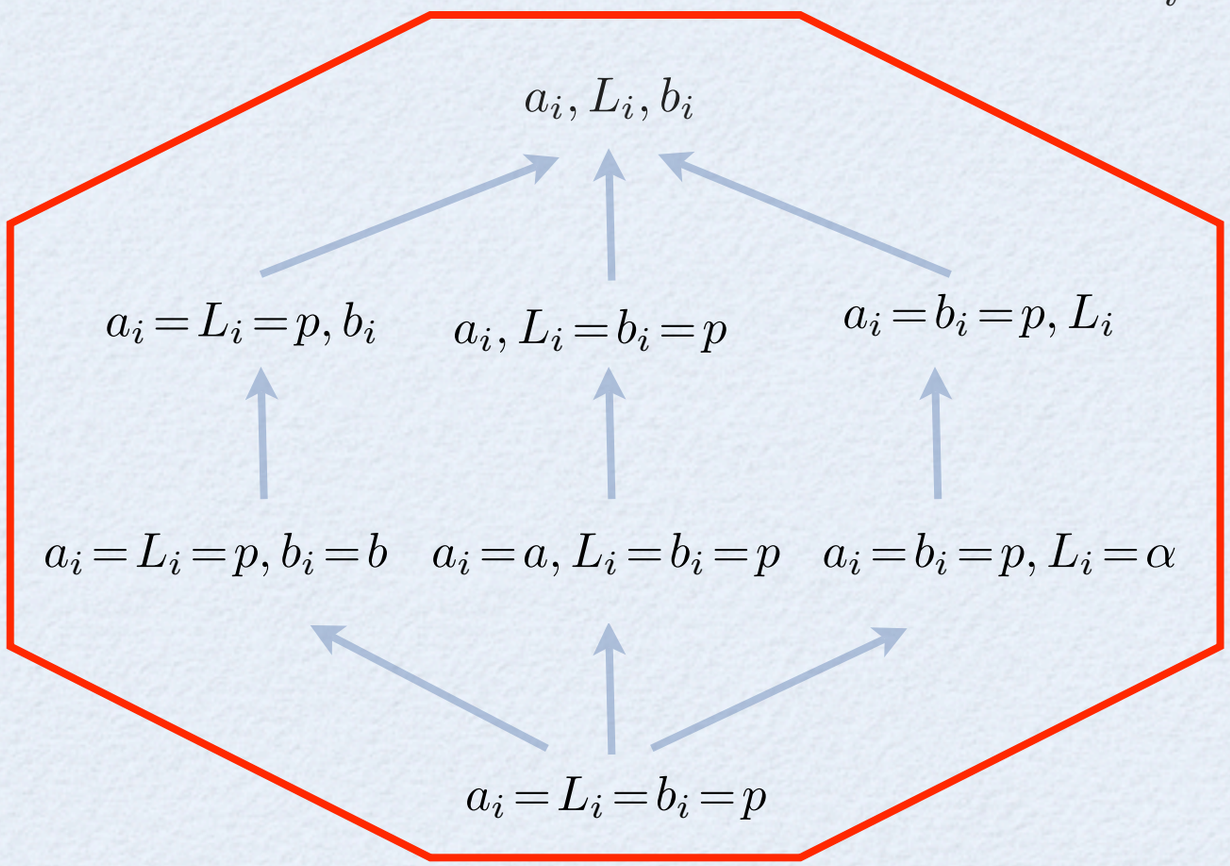


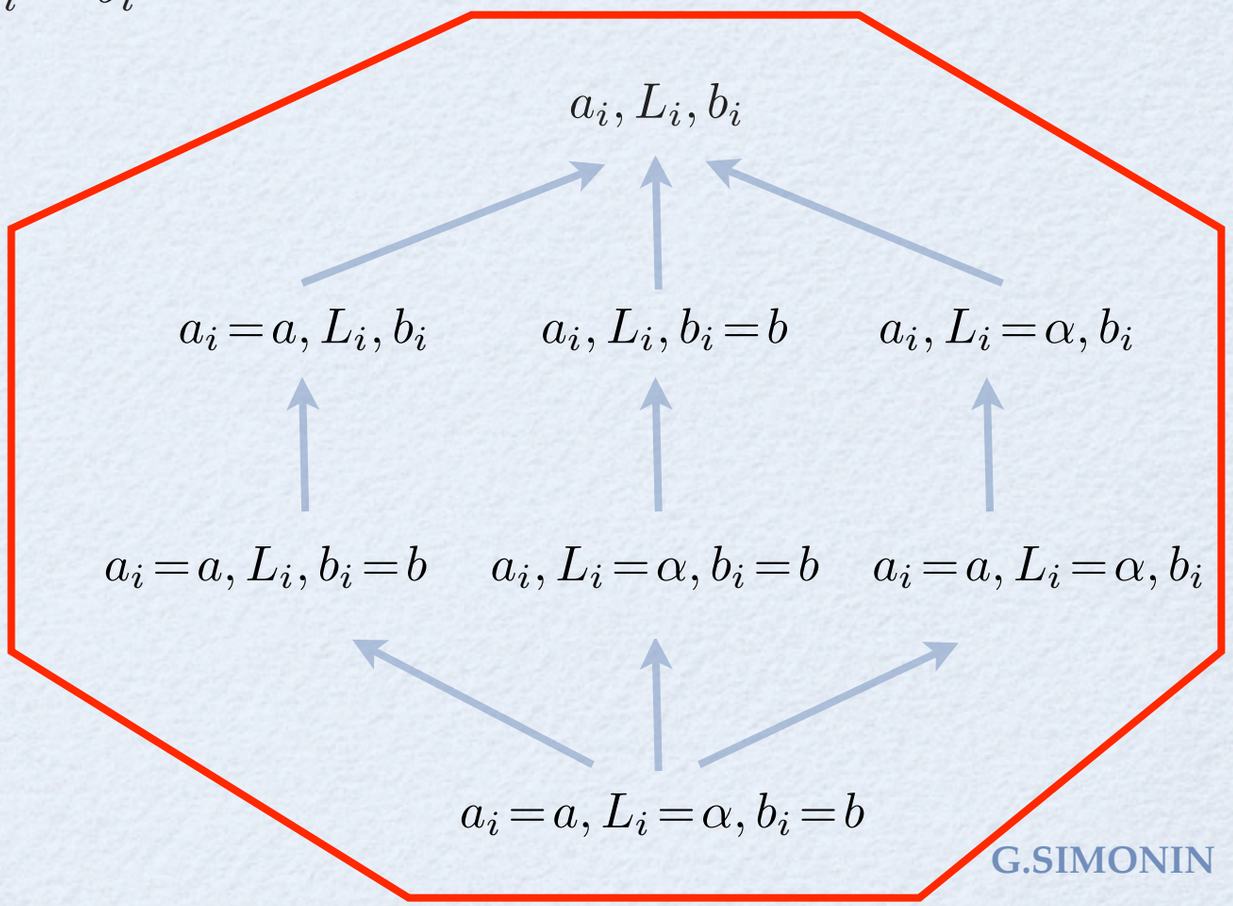


Classification des heuristiques

Couplage maximum



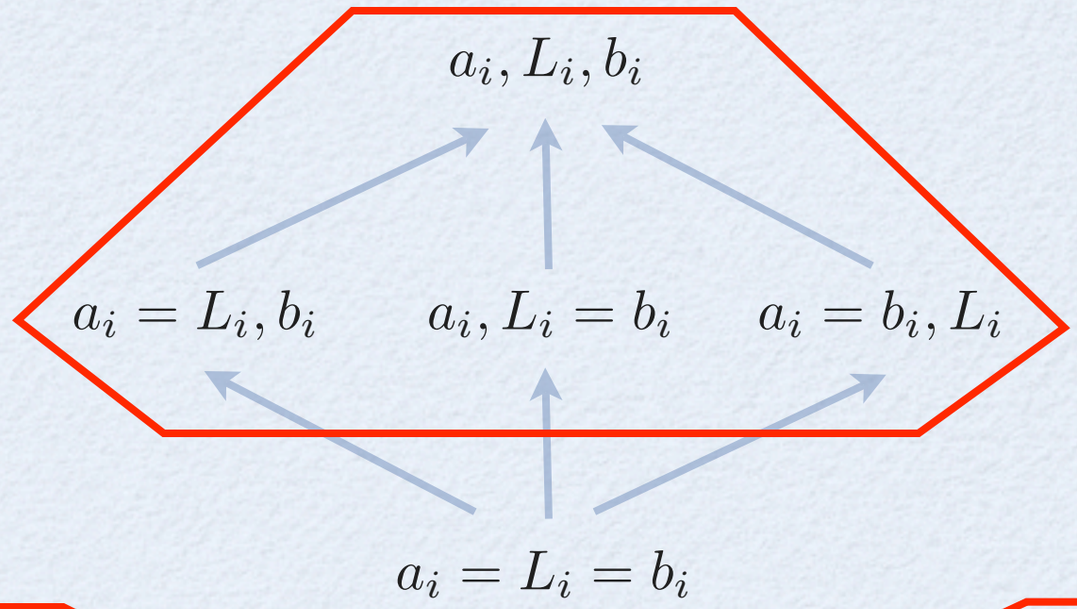


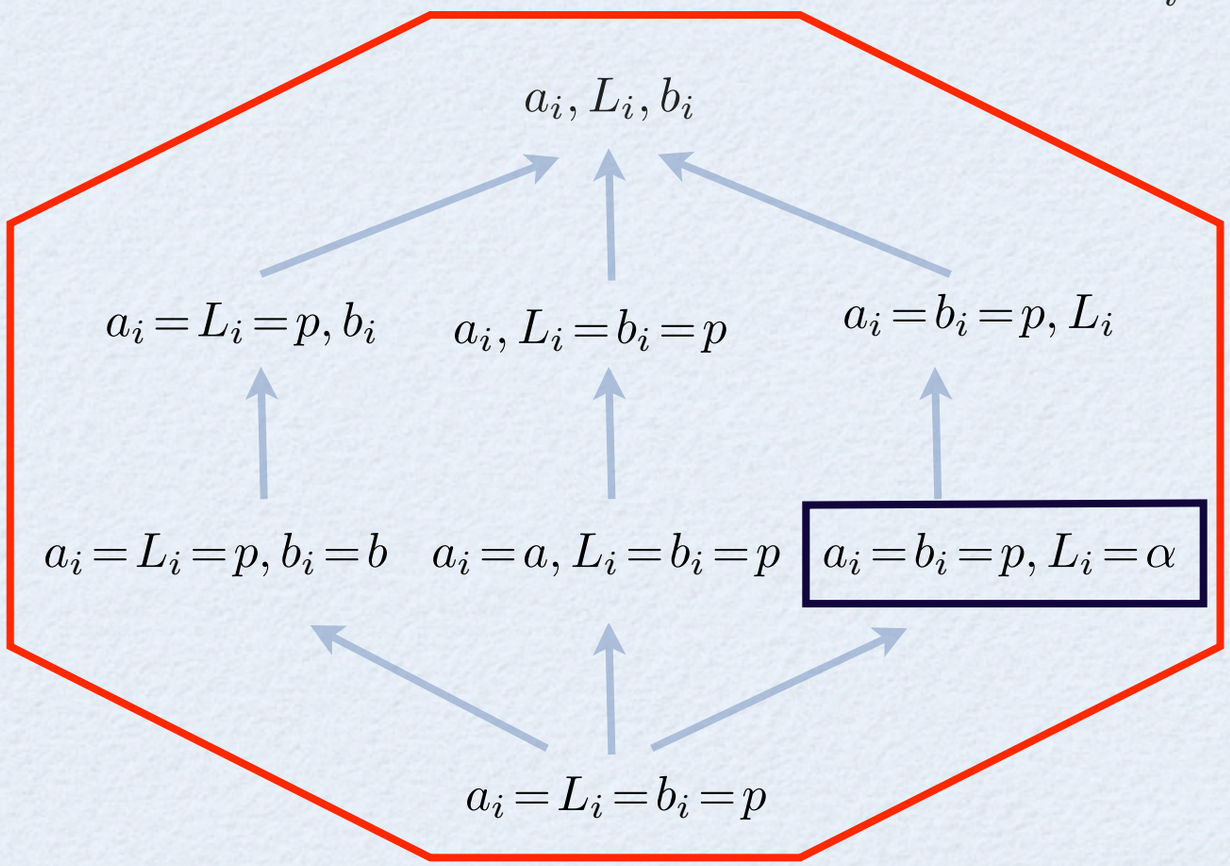


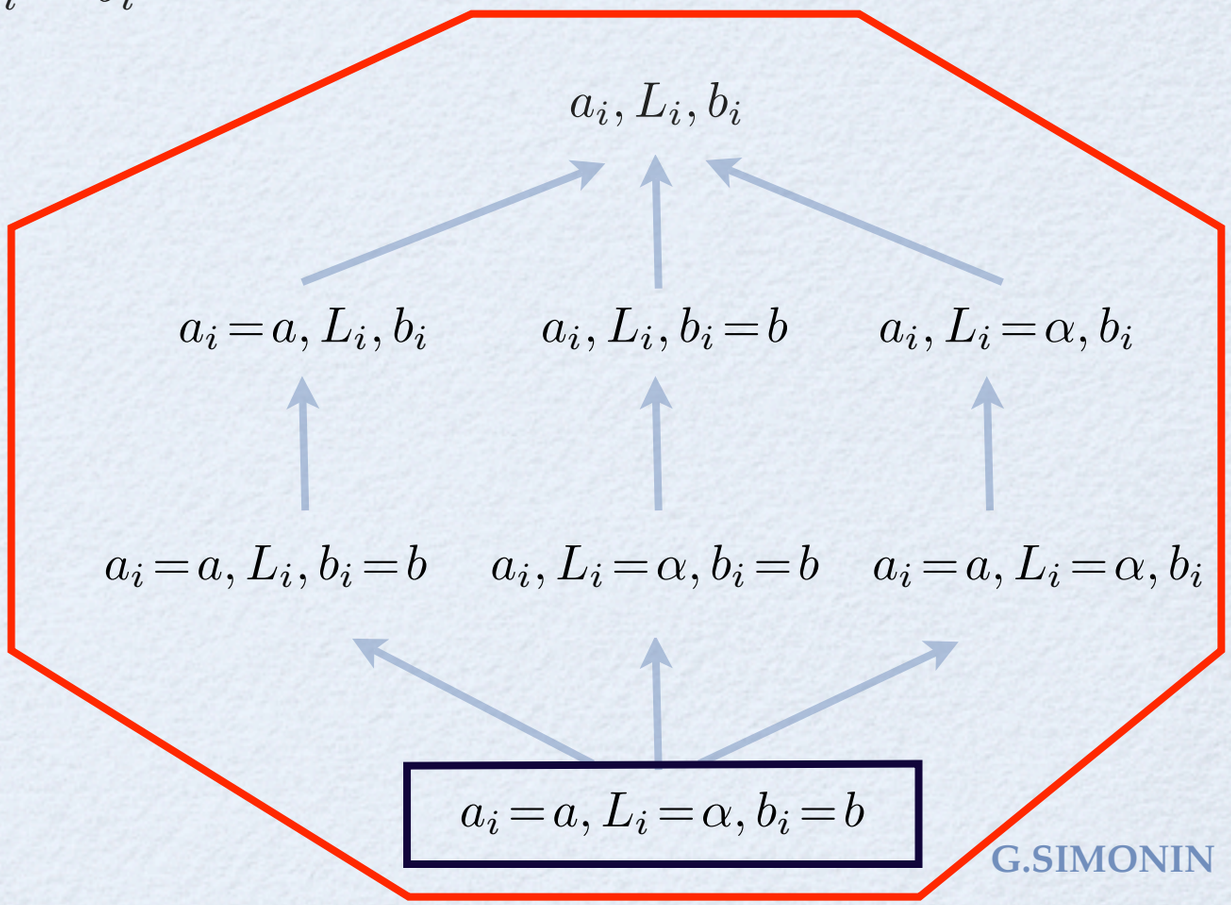
Classification des heuristiques

Couplage maximum

Cliques maximales





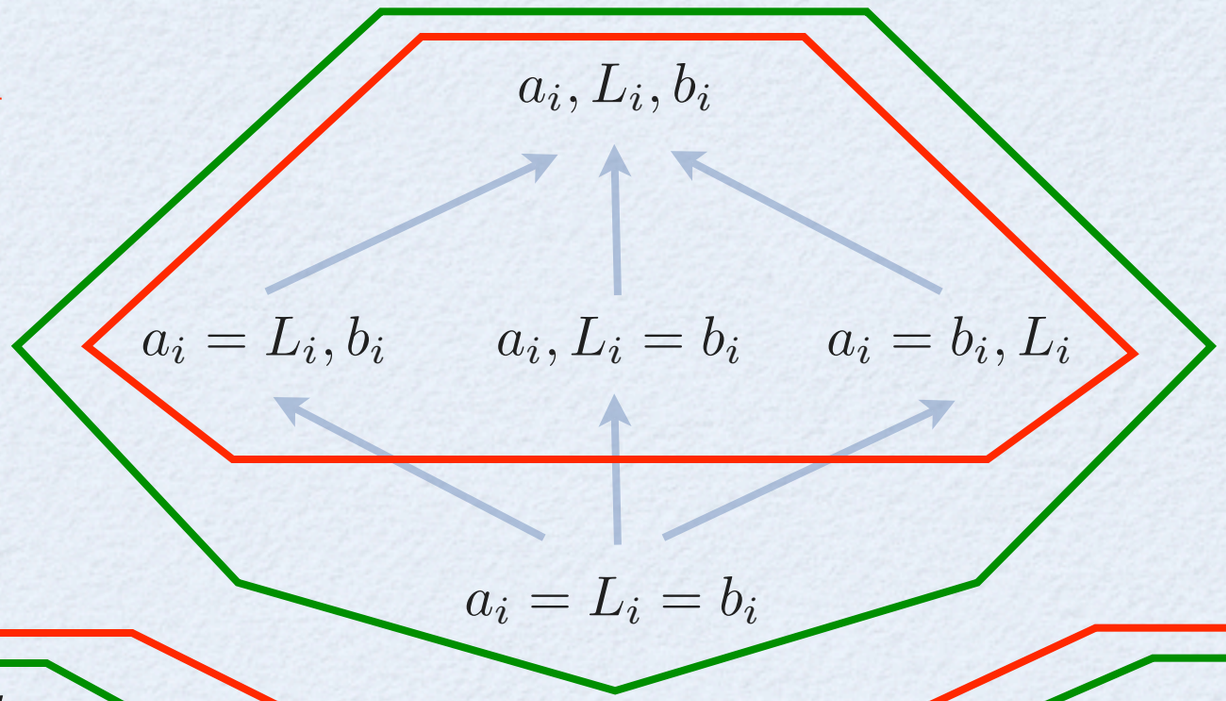


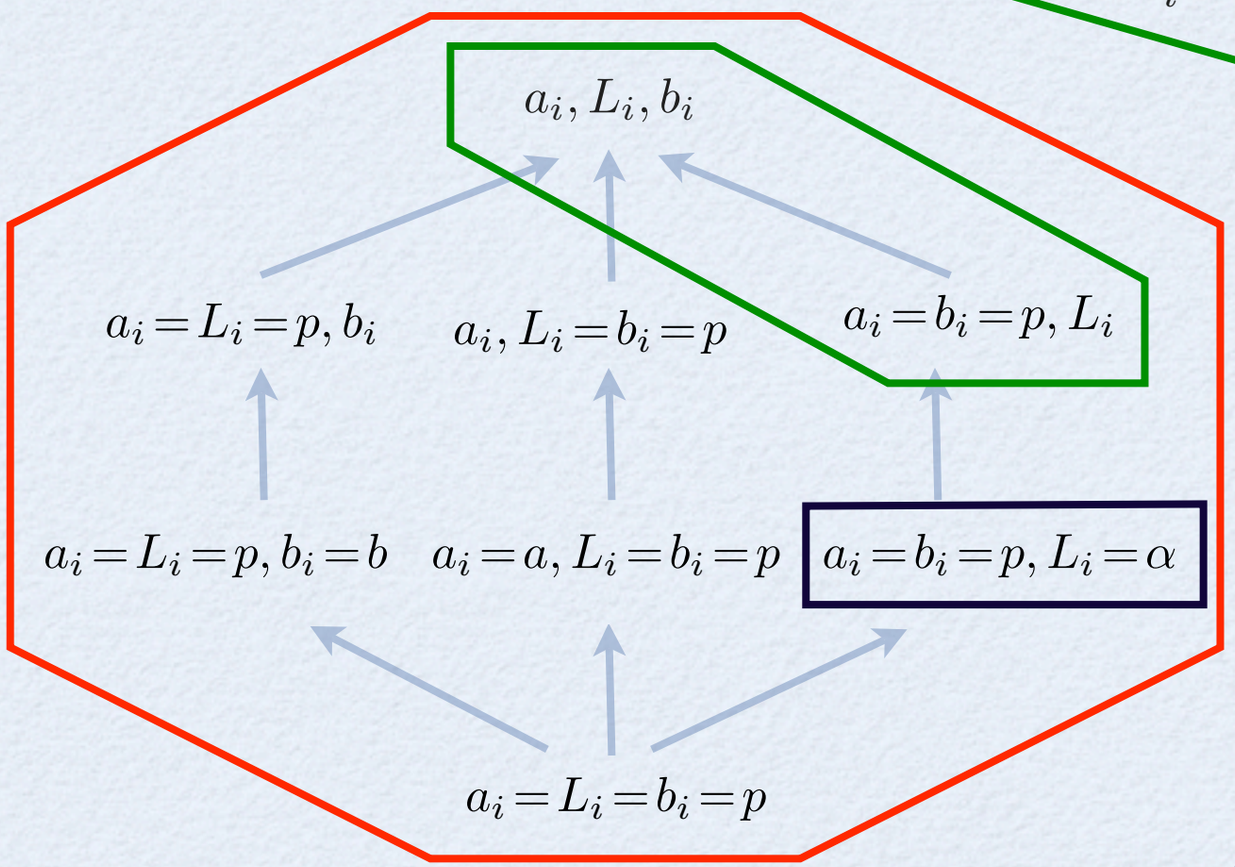
Classification des heuristiques

Couplage maximum

Cliques maximales

Poupées russes





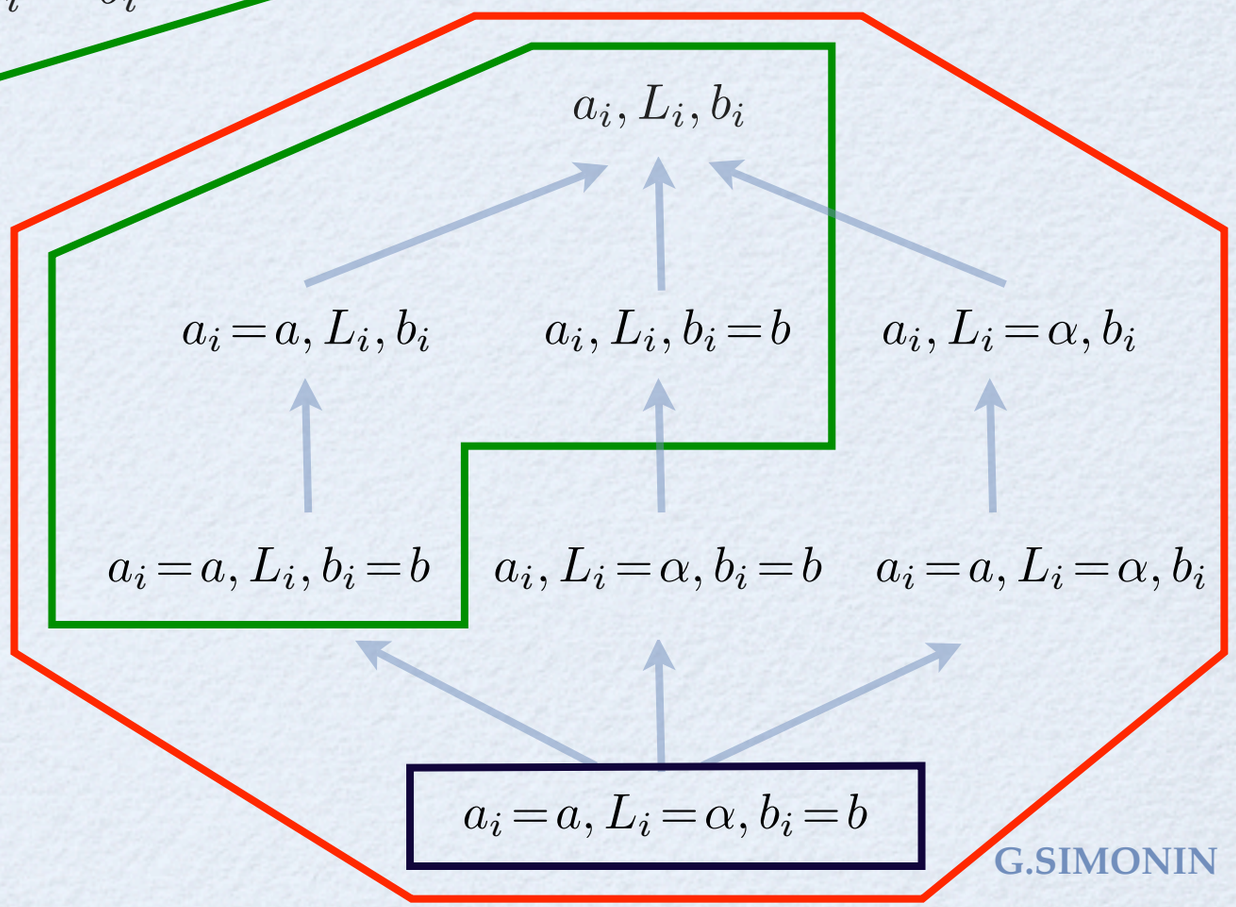


Table des matières

- Introduction
  - Présentation du problème
  - Modélisation
- Complexité de ces problèmes
  - Etat de l'art
  - Preuve de NP-complétude sur un cas particulier
- Les techniques d'approximation
  - Couplage maximum
  - Clique maximale
  - Poupées russes
- Les techniques de non approximation
  - Clique partition
  - Triangle packing
- Conclusion et perspectives

Bornes de non approximation

- Les différentes bornes sont obtenues par réductions faites à partir de ces deux problèmes NP-complets :
  - Clique partition
    - Donnée : Un graphe $G = (V, E)$
    - Résultat : Recouvrement de V par des sous graphes complets induits par V
  - Triangle packing
    - Donnée : Un graphe $G = (V, E)$
    - Résultat : Recouvrement de V par des triangles induits par V

Réduction faite à partir de Triangle packing

Exemple de réduction pour $1|prec;$ tâche-couplée$; (a_i, L_i, b_i); G_c|C_{max}$:

Réduction faite à partir de Triangle packing

Exemple de réduction pour $1|prec;$ tâche-couplée$; (a_i, L_i, b_i); G_c|C_{max}$:

- Soit I une instance de Triangle Packing.
- Soit $I^* = \{(a_j, l_j, b_j) : j \in J; G_c; G_p\}$ une instance de notre problème

Réduction faite à partir de Triangle packing

Exemple de réduction pour $1|prec;$ tâche-couplée$; (a_i, L_i, b_i); G_c|C_{max}$:

- Soit I une instance de Triangle Packing.
- Soit $I^* = \{(a_j, l_j, b_j) : j \in J; G_c; G_p\}$ une instance de notre problème
- Soit $J = \{1, \ldots, n+3\}$ et $\mathcal{Z} = \{A_j |\ j = 1, \ldots, n\}$:

Réduction faite à partir de Triangle packing

Exemple de réduction pour $1|prec;$ tâche-couplée$; (a_i, L_i, b_i); G_c|C_{max}$:

- Soit I une instance de Triangle Packing.
- Soit $I^* = \{(a_j, l_j, b_j) : j \in J; G_c; G_p\}$ une instance de notre problème
- Soit $J = \{1, \ldots, n+3\}$ et $\mathcal{Z} = \{A_j |\ j = 1, \ldots, n\}$:

$a_j = b_j = 1,\ L_j = 2$ pour $j = 1, \ldots, n$ où $n = 3q$

Réduction faite à partir de Triangle packing

Exemple de réduction pour $1|prec;$ tâche-couplée$; (a_i, L_i, b_i); G_c|C_{max}$:

- Soit I une instance de Triangle Packing.
- Soit $I^* = \{(a_j, l_j, b_j) : j \in J; G_c; G_p\}$ une instance de notre problème
- Soit $J = \{1, \ldots, n+3\}$ et $\mathcal{Z} = \{A_j |\ j = 1, \ldots, n\}$:

$a_j = b_j = 1,\ L_j = 2$ pour $j = 1, \ldots, n$ où $n = 3q$

$a_{n+1} = b_{n+1} = 1,\ L_{n+1} = B$

Réduction faite à partir de Triangle packing

Exemple de réduction pour $1|prec;$ tâche-couplée$; (a_i, L_i, b_i); G_c|C_{max}$:

- Soit I une instance de Triangle Packing.
- Soit $I^* = \{(a_j, l_j, b_j) : j \in J; G_c; G_p\}$ une instance de notre problème
- Soit $J = \{1, \ldots, n+3\}$ et $\mathcal{Z} = \{A_j |\ j = 1, \ldots, n\}$:

$a_j = b_j = 1,\ L_j = 2$ pour $j = 1, \ldots, n$ où $n = 3q$

$a_{n+1} = b_{n+1} = 1,\ L_{n+1} = B$

$a_{n+2} = B,\ b_{n+2} = 1,\ L_{n+2} = B + 2n + 1$

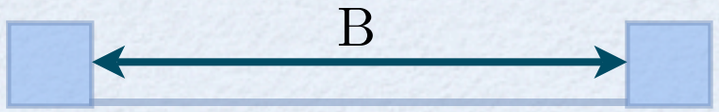
Réduction faite à partir de Triangle packing

Exemple de réduction pour $1|prec;$ tâche-couplée$; (a_i, L_i, b_i); G_c|C_{max}$:

- Soit I une instance de Triangle Packing.
- Soit $I^* = \{(a_j, l_j, b_j) : j \in J; G_c; G_p\}$ une instance de notre problème
- Soit $J = \{1, \ldots, n+3\}$ et $\mathcal{Z} = \{A_j |\ j = 1, \ldots, n\}$:

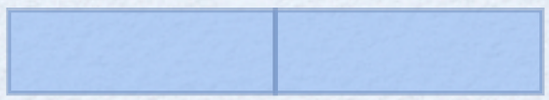
$a_j = b_j = 1,\ L_j = 2$ pour $j = 1, \ldots, n$ où $n = 3q$

$a_{n+1} = b_{n+1} = 1,\ L_{n+1} = B$



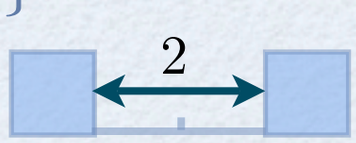
$a_{n+2} = B,\ b_{n+2} = 1,\ L_{n+2} = B + 2n + 1$

$a_{n+3} = b_{n+3} = \dfrac{B}{2},\ L_{n+3} = 0$



- Soit $J = \{1, \ldots, n+3\}$ et $\mathcal{Z} = \{A_j |\ j = 1, \ldots, n\}$:

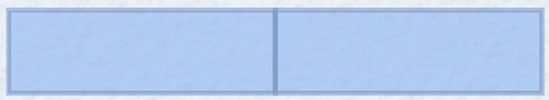
$a_j = b_j = 1,\ L_j = 2$ pour $j = 1, \ldots, n$ où $n = 3q$



$a_{n+1} = b_{n+1} = 1,\ L_{n+1} = B$

$a_{n+2} = B,\ b_{n+2} = 1,\ L_{n+2} = B + 2n + 1$

$a_{n+3} = b_{n+3} = \dfrac{B}{2},\ L_{n+3} = 0$



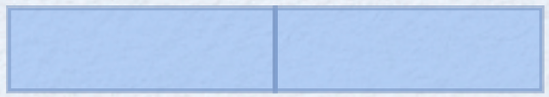
- Soit $J = \{1, \ldots, n+3\}$ et $\mathcal{Z} = \{A_j |\ j = 1, \ldots, n\}$:

$a_j = b_j = 1,\ L_j = 2$ pour $j = 1, \ldots, n$ où $n = 3q$

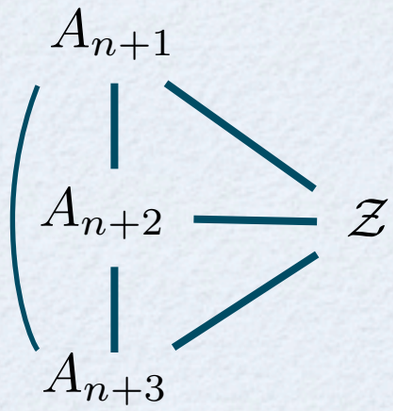
$a_{n+1} = b_{n+1} = 1,\ L_{n+1} = B$

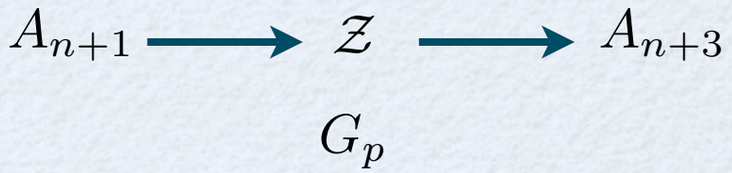
$a_{n+2} = B,\ b_{n+2} = 1,\ L_{n+2} = B + 2n + 1$

$a_{n+3} = b_{n+3} = \dfrac{B}{2},\ L_{n+3} = 0$



G_c





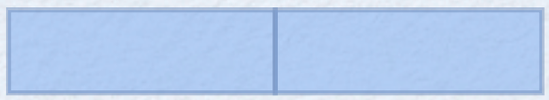
- Soit $J = \{1, \ldots, n+3\}$ et $\mathcal{Z} = \{A_j |\ j = 1, \ldots, n\}$:

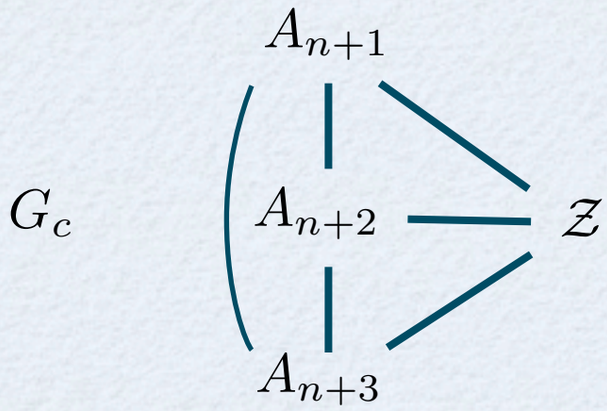
$a_j = b_j = 1,\ L_j = 2$ pour $j = 1, \ldots, n$ où $n = 3q$

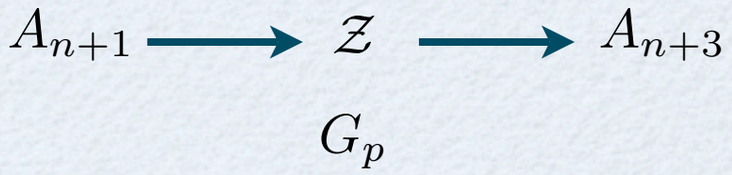
$a_{n+1} = b_{n+1} = 1,\ L_{n+1} = B$

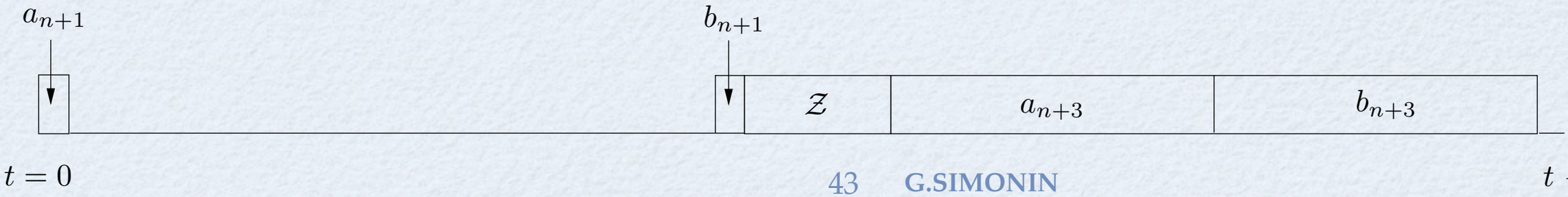
$a_{n+2} = B,\ b_{n+2} = 1,\ L_{n+2} = B + 2n + 1$

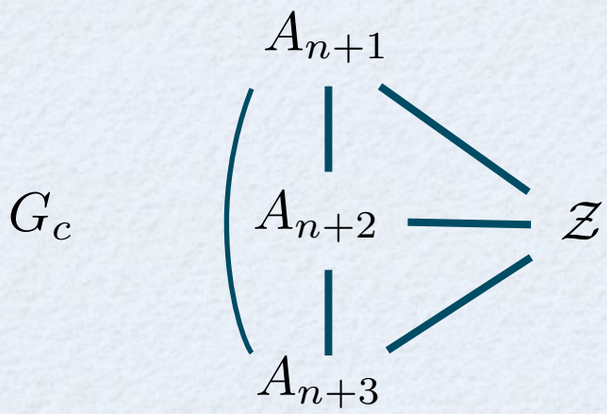
$a_{n+3} = b_{n+3} = \dfrac{B}{2},\ L_{n+3} = 0$

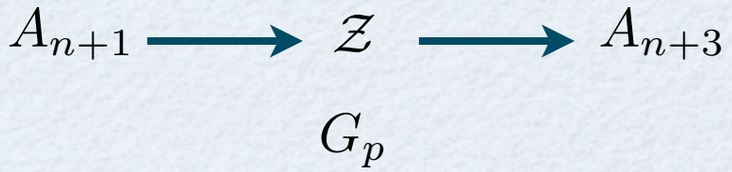


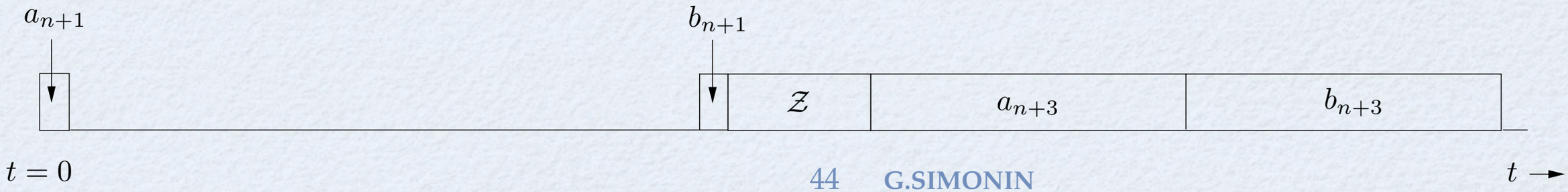


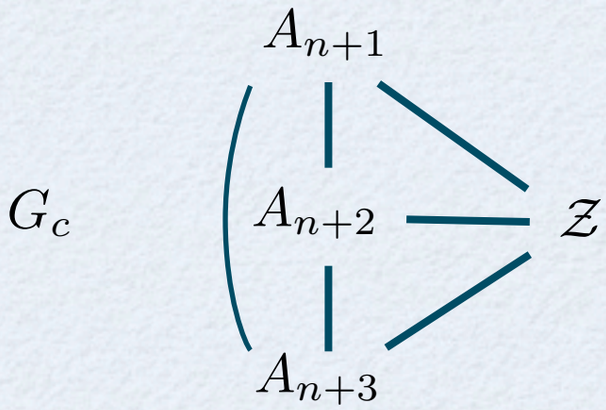


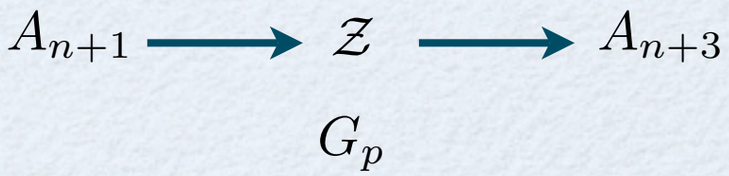



A_{n+1}
G_c
A_{n+2}
$\mathcal{Z}$
A_{n+3}

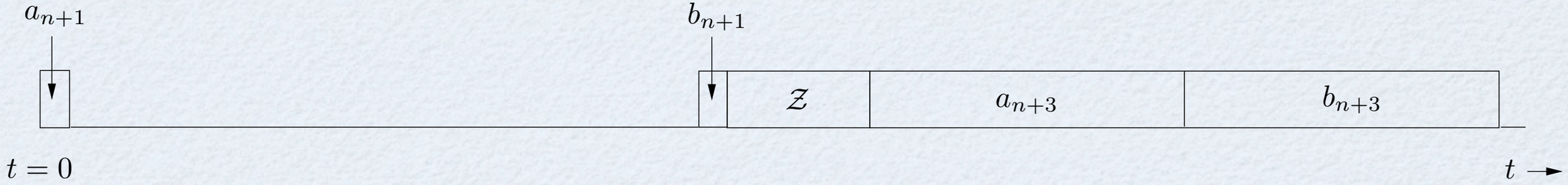

A_{n+1}
$\mathcal{Z}$
A_{n+3}
G_p

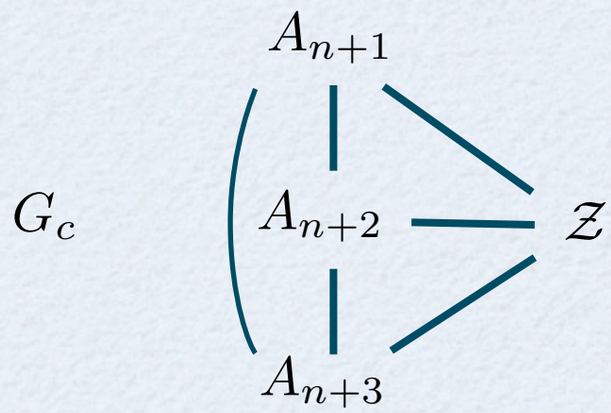

a_{n+1}
b_{n+1}
$\mathcal{Z}$
a_{n+3}
b_{n+3}
$t = 0$
$t \rightarrow$

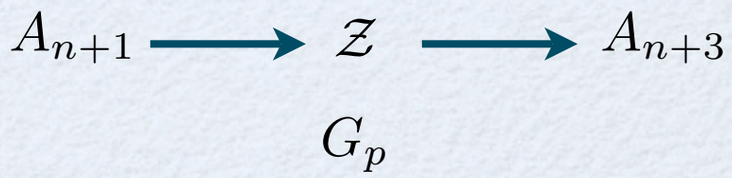




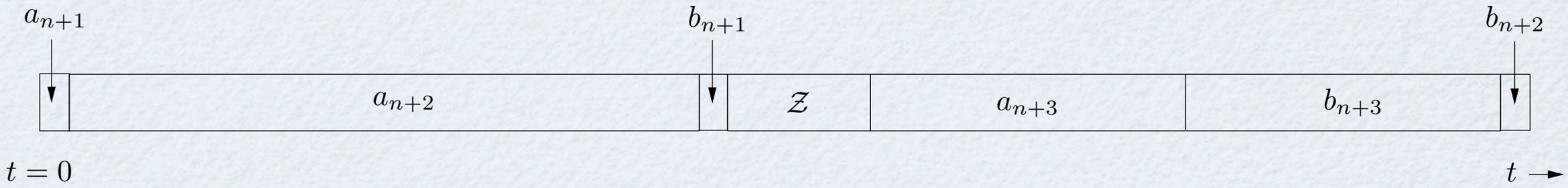
S'il existe un recouvrement par triangles des sommets de $\mathcal{Z}$

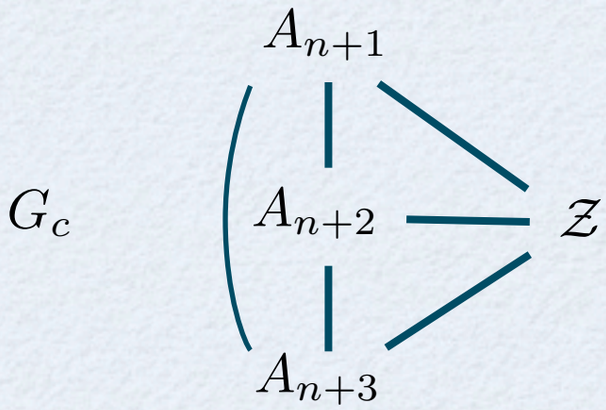


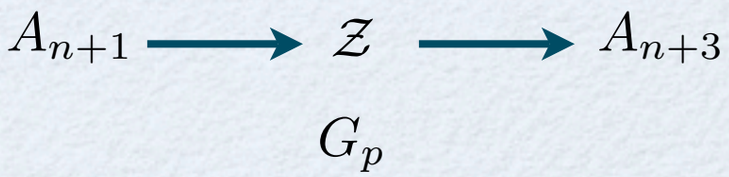




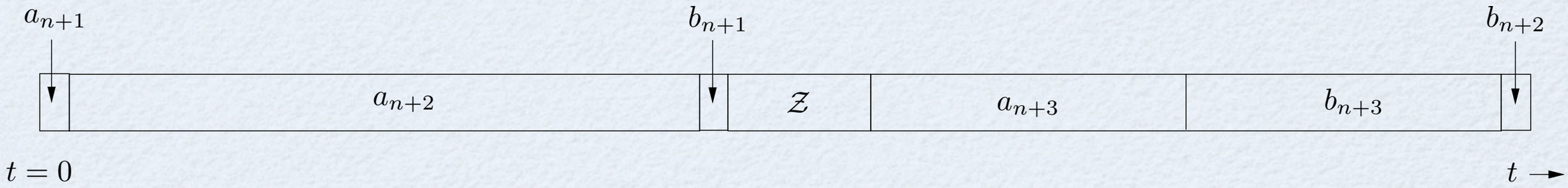
S'il existe un recouvrement par triangles des sommets de $\mathcal{Z}$



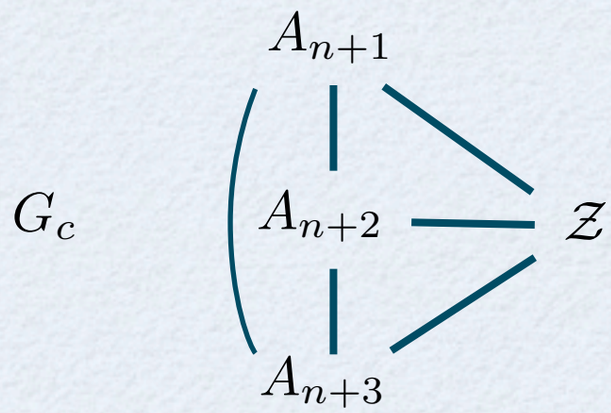


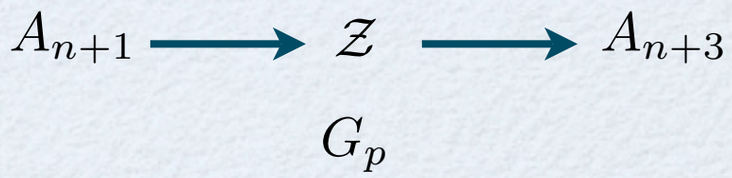


S'il existe un recouvrement par triangles des sommets de $\mathcal{Z}$

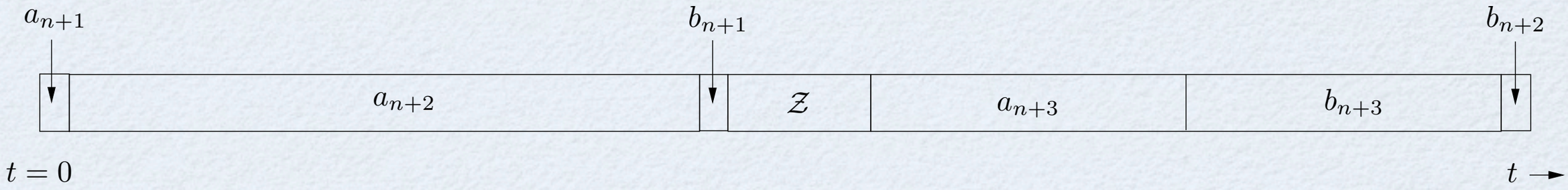


S'il n'existe pas de recouvrement par triangles des sommets de $\mathcal{Z}$

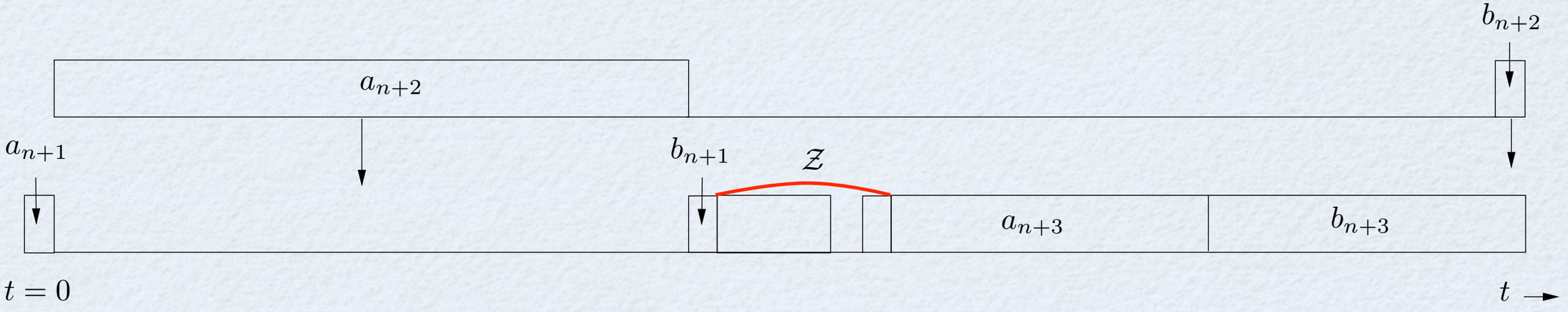




S'il existe un recouvrement par triangles des sommets de $\mathcal{Z}$



S'il n'existe pas de recouvrement par triangles des sommets de $\mathcal{Z}$



Réduction faite à partir de Clique partition

Exemple de réduction pour $1|prec;$ tâche-couplée; $(a_i = L_i = b_i); G_c|C_{max}$:

Réduction faite à partir de Clique partition

Exemple de réduction pour $1|prec;$ tâche-couplée; $(a_i = L_i = b_i); G_c|C_{max}$:

- Soit I une instance de Clique partition.
- Soit $I^* = \{(a_j, l_j, b_j) : j \in J; G_c; G_p\}$ une instance de notre problème

Réduction faite à partir de Clique partition

Exemple de réduction pour $1|prec;$ tâche-couplée; $(a_i = L_i = b_i); G_c|C_{max}$:

- Soit I une instance de Clique partition.
- Soit $I^* = \{(a_j, l_j, b_j) : j \in J; G_c; G_p\}$ une instance de notre problème
- Soit $J = \{1, \ldots, n+3\}$ et $\mathcal{Z} = \{A_j |\ j = 1, \ldots, n\}$:

Réduction faite à partir de Clique partition

Exemple de réduction pour $1|prec;$ tâche-couplée; $(a_i = L_i = b_i); G_c|C_{max}$:

- Soit I une instance de Clique partition.
- Soit $I^* = \{(a_j, l_j, b_j) : j \in J; G_c; G_p\}$ une instance de notre problème
- Soit $J = \{1, \ldots, n+3\}$ et $\mathcal{Z} = \{A_j |\ j = 1, \ldots, n\}$:

$a_j = L_j = b_j = w_j$ pour $j = 1, \ldots, n$ où chaque w_j est différent

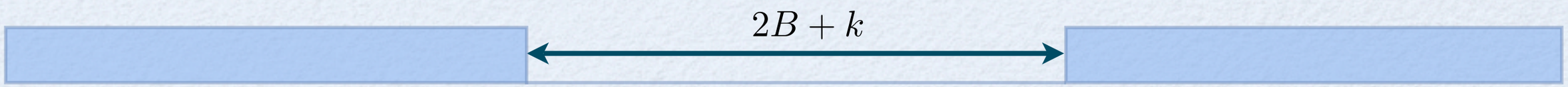
Réduction faite à partir de Clique partition

Exemple de réduction pour $1|prec;$ tâche-couplée; $(a_i = L_i = b_i); G_c|C_{max}$:

- Soit I une instance de Clique partition.
- Soit $I^* = \{(a_j, l_j, b_j) : j \in J; G_c; G_p\}$ une instance de notre problème
- Soit $J = \{1, \ldots, n+3\}$ et $\mathcal{Z} = \{A_j |\ j = 1, \ldots, n\}$:

$a_j = L_j = b_j = w_j$ pour $j = 1, \ldots, n$ où chaque w_j est différent

$a_{n+2} = L_{n+2} = b_{n+2} = 2B + k$ k le temps d'exécution de l'ensemble $\mathcal{Z}$



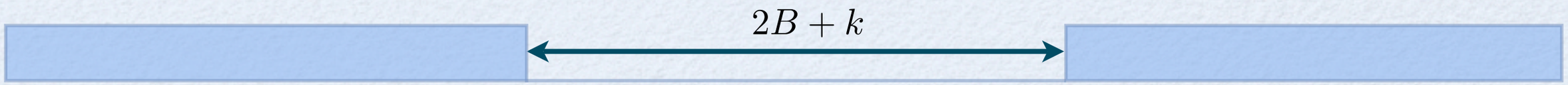
Réduction faite à partir de Clique partition

Exemple de réduction pour $1|prec;$ tâche-couplée$; (a_i = L_i = b_i); G_c|C_{max}$:

- Soit I une instance de Clique partition.
- Soit $I^* = \{(a_j, l_j, b_j) : j \in J; G_c; G_p\}$ une instance de notre problème
- Soit $J = \{1, \ldots, n+3\}$ et $\mathcal{Z} = \{A_j |\ j = 1, \ldots, n\}$:

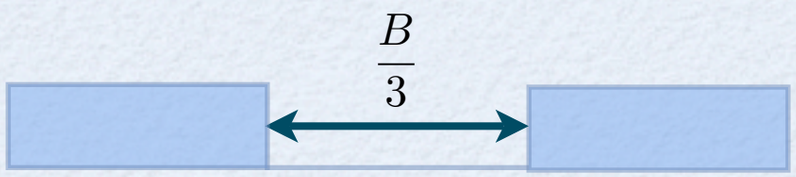
$a_j = L_j = b_j = w_j$ pour $j = 1, \ldots, n$ où chaque w_j est différent

$a_{n+2} = L_{n+2} = b_{n+2} = 2B + k$ k le temps d'exécution de l'ensemble $\mathcal{Z}$



$a_{n+1} = L_{n+1} = b_{n+1} = \frac{B}{3}$

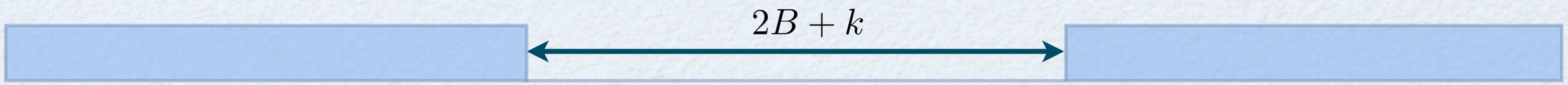
$a_{n+3} = L_{n+3} = b_{n+3} = \frac{B}{3}$



- Soit $J = \{1, \ldots, n+3\}$ et $\mathcal{Z} = \{A_j |\ j = 1, \ldots, n\}$:

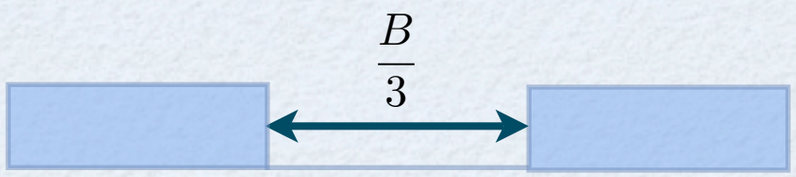
$a_j = L_j = b_j = w_j$ pour $j = 1, \ldots, n$ où chaque w_j est différent

$a_{n+2} = L_{n+2} = b_{n+2} = 2B + k$ k le temps d'exécution de l'ensemble $\mathcal{Z}$



$a_{n+1} = L_{n+1} = b_{n+1} = \frac{B}{3}$

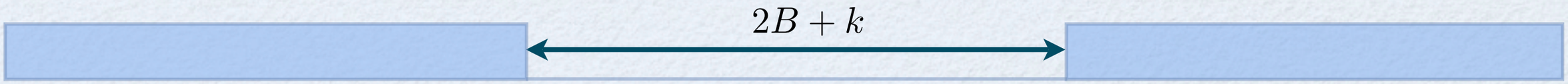
$a_{n+3} = L_{n+3} = b_{n+3} = \frac{B}{3}$



- Soit $J = \{1, \ldots, n+3\}$ et $\mathcal{Z} = \{A_j |\ j = 1, \ldots, n\}$:

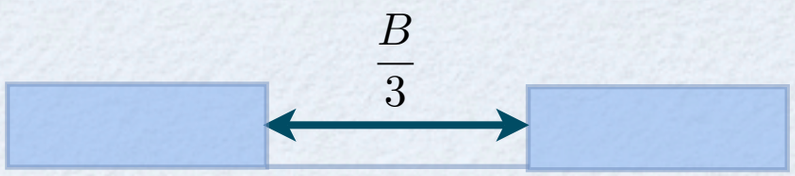
  $a_j = L_j = b_j = w_j$ pour $j = 1, \ldots, n$ où chaque w_j est différent

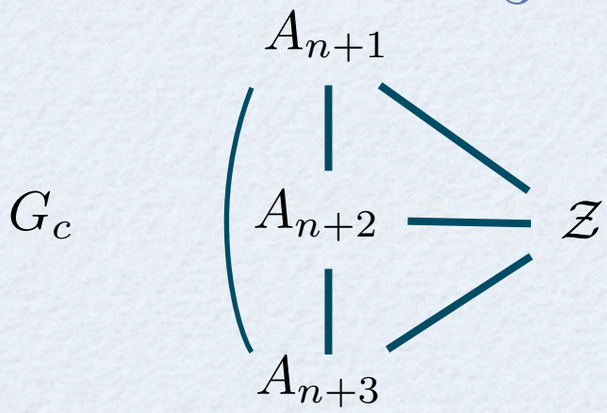
  $a_{n+2} = L_{n+2} = b_{n+2} = 2B + k$ k le temps d'exécution de l'ensemble $\mathcal{Z}$

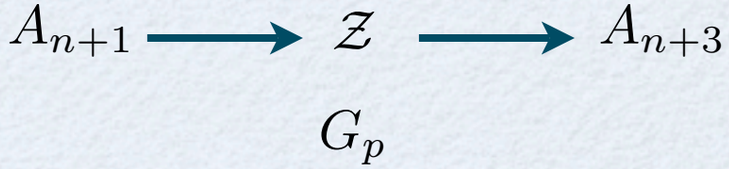


$$a_{n+1} = L_{n+1} = b_{n+1} = \frac{B}{3}$$

$$a_{n+3} = L_{n+3} = b_{n+3} = \frac{B}{3}$$



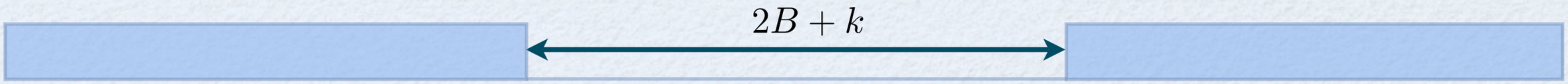




- Soit $J = \{1, \ldots, n+3\}$ et $\mathcal{Z} = \{A_j |\ j = 1, \ldots, n\}$:

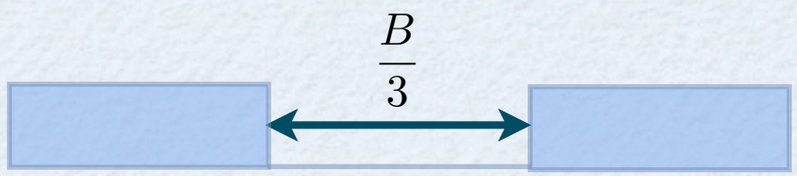
  $a_j = L_j = b_j = w_j$ pour $j = 1, \ldots, n$ où chaque w_j est différent

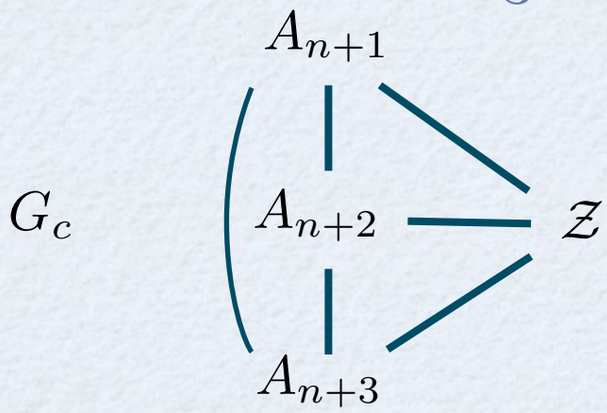
  $a_{n+2} = L_{n+2} = b_{n+2} = 2B + k$ k le temps d'exécution de l'ensemble $\mathcal{Z}$

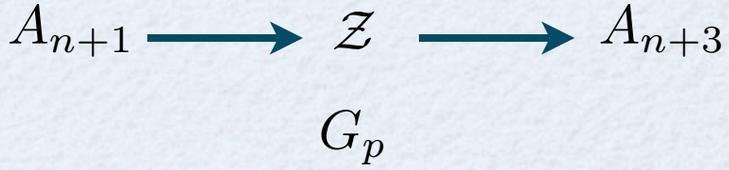


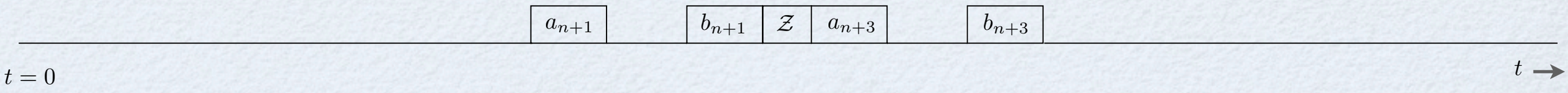
$$a_{n+1} = L_{n+1} = b_{n+1} = \frac{B}{3}$$

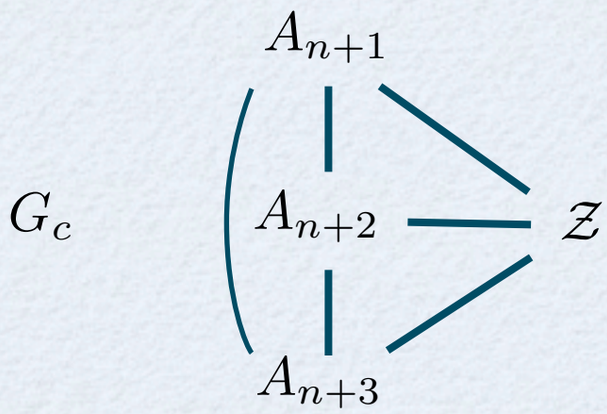
$$a_{n+3} = L_{n+3} = b_{n+3} = \frac{B}{3}$$

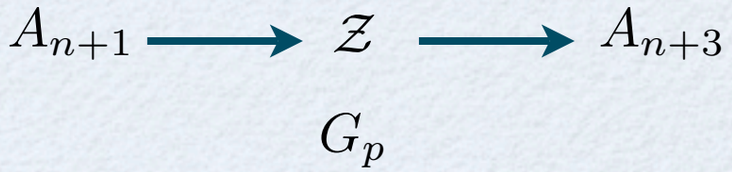


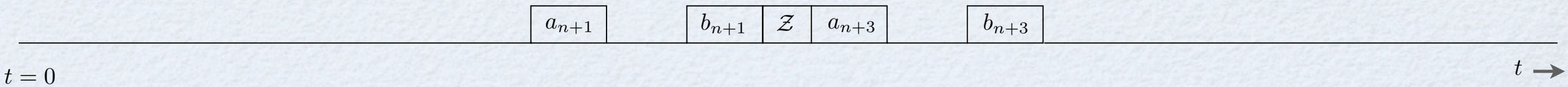


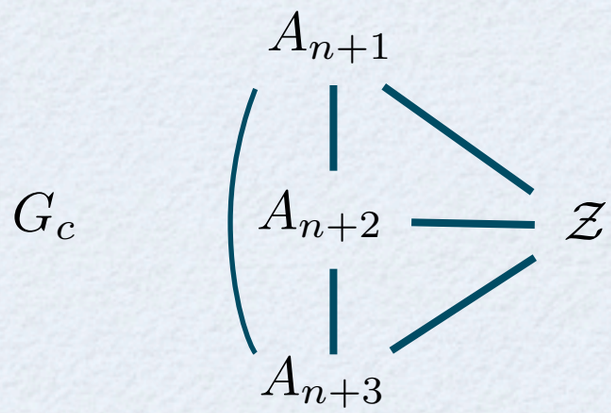


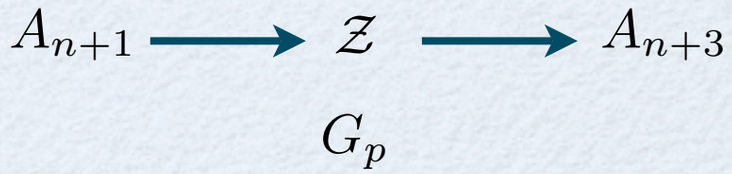



G_c
A_{n+1}
A_{n+2}
A_{n+3}
$\mathcal{Z}$

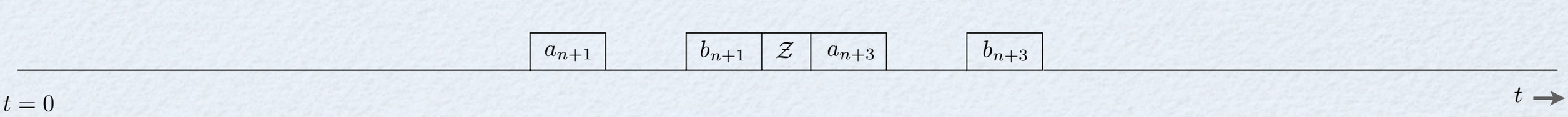

A_{n+1}
$\mathcal{Z}$
A_{n+3}
G_p

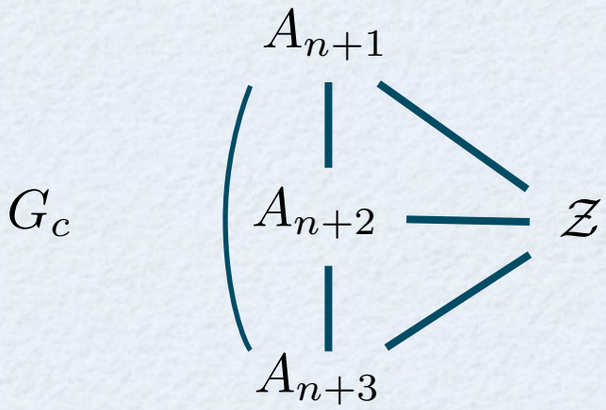

a_{n+1}
b_{n+1}
$\mathcal{Z}$
a_{n+3}
b_{n+3}
$t = 0$
t

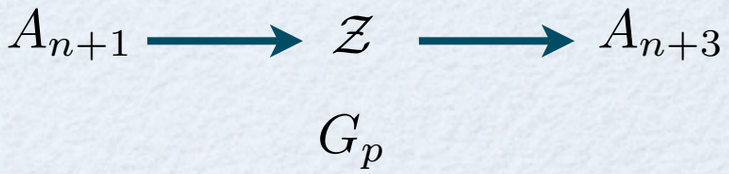




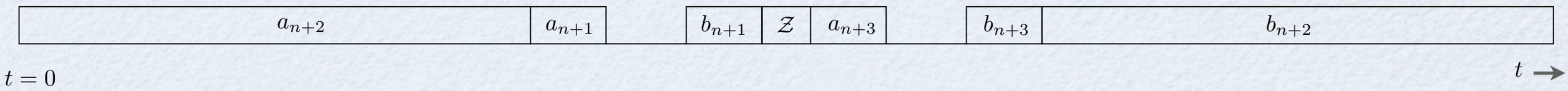
S'il existe un recouvrement par cliques des sommets de $\mathcal{Z}$

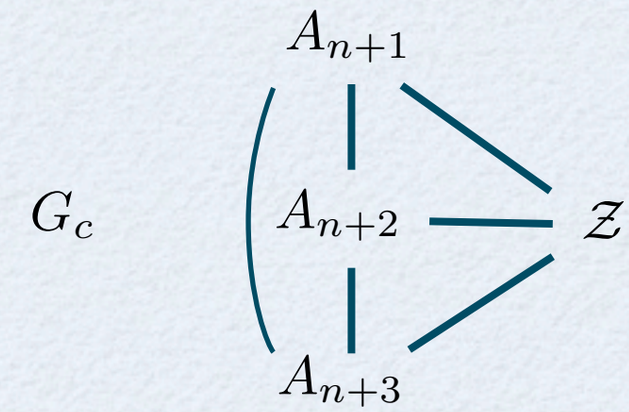


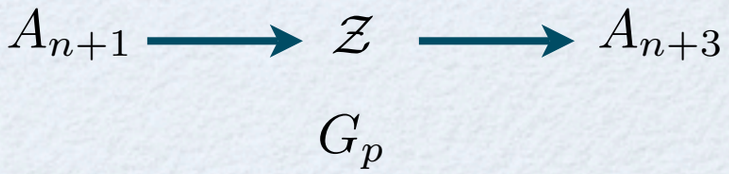




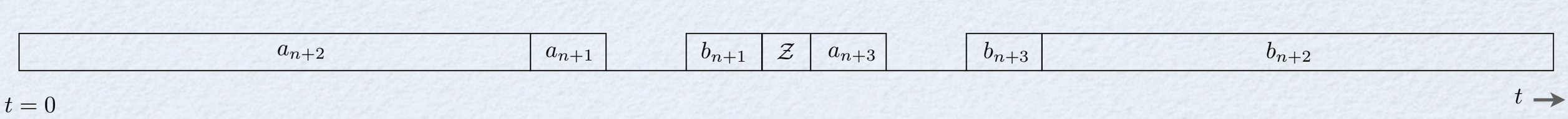
S'il existe un recouvrement par cliques des sommets de $\mathcal{Z}$



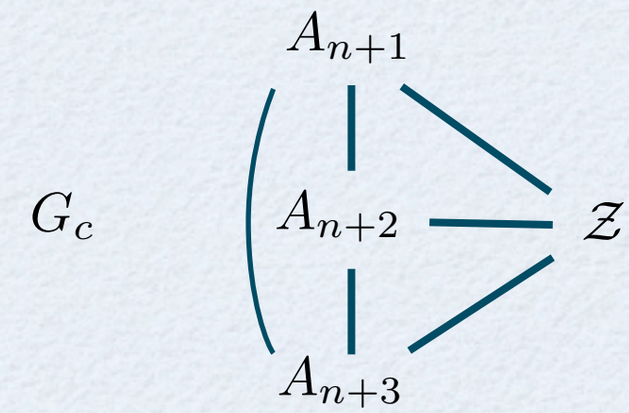


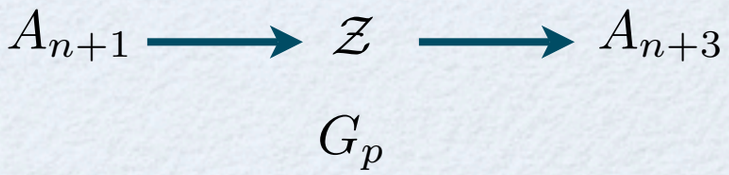


S'il existe un recouvrement par cliques des sommets de $\mathcal{Z}$

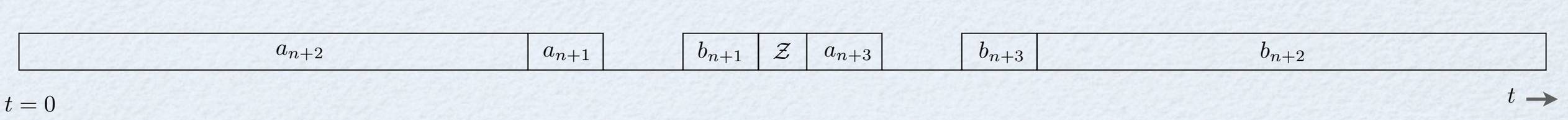


S'il n'existe pas de recouvrement par cliques des sommets de $\mathcal{Z}$

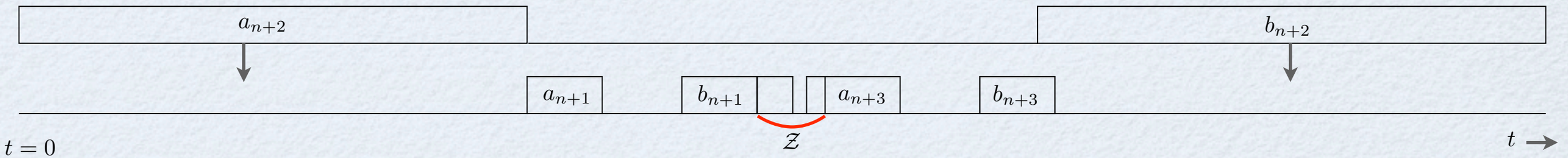




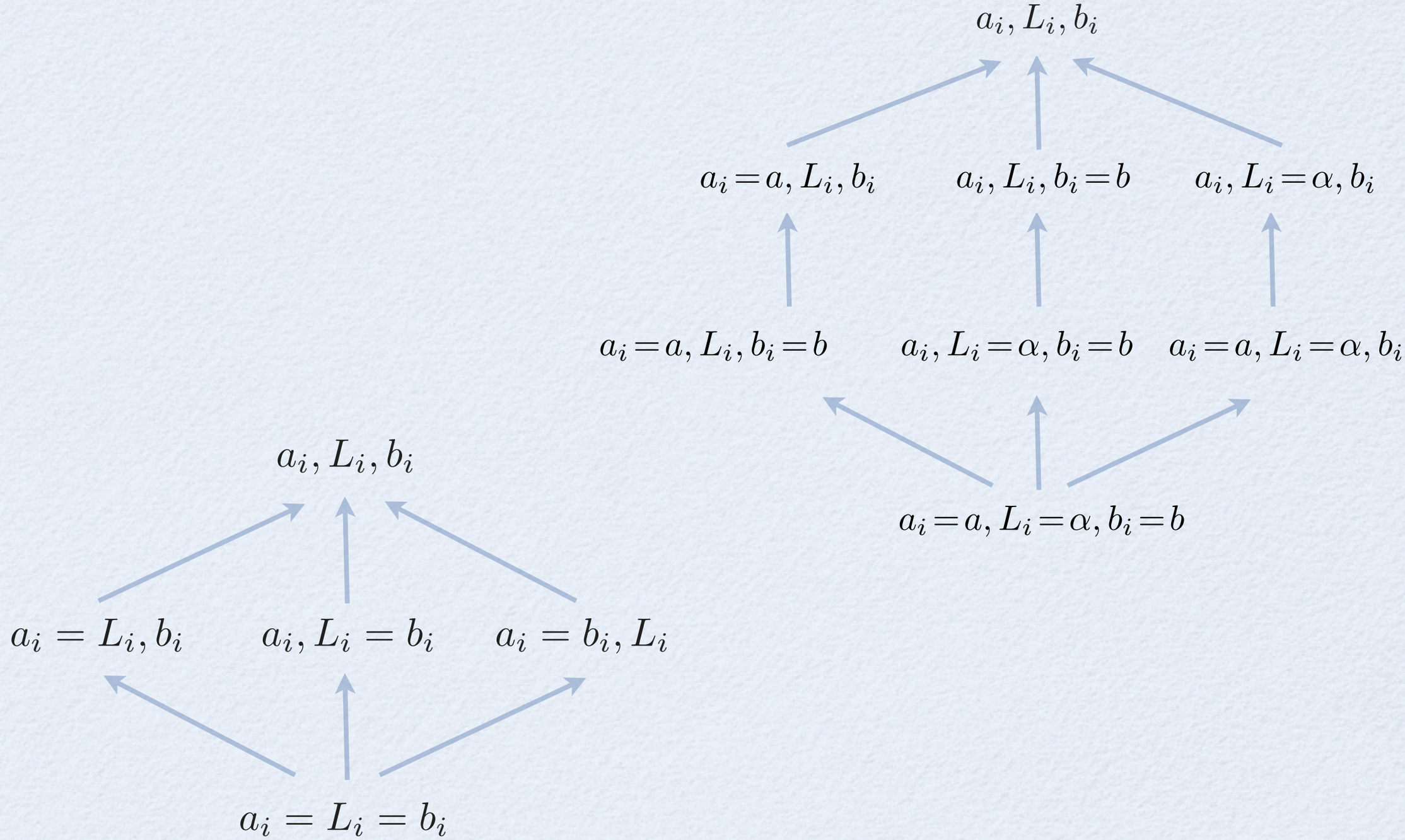
S'il existe un recouvrement par cliques des sommets de $\mathcal{Z}$



S'il n'existe pas de recouvrement par cliques des sommets de $\mathcal{Z}$

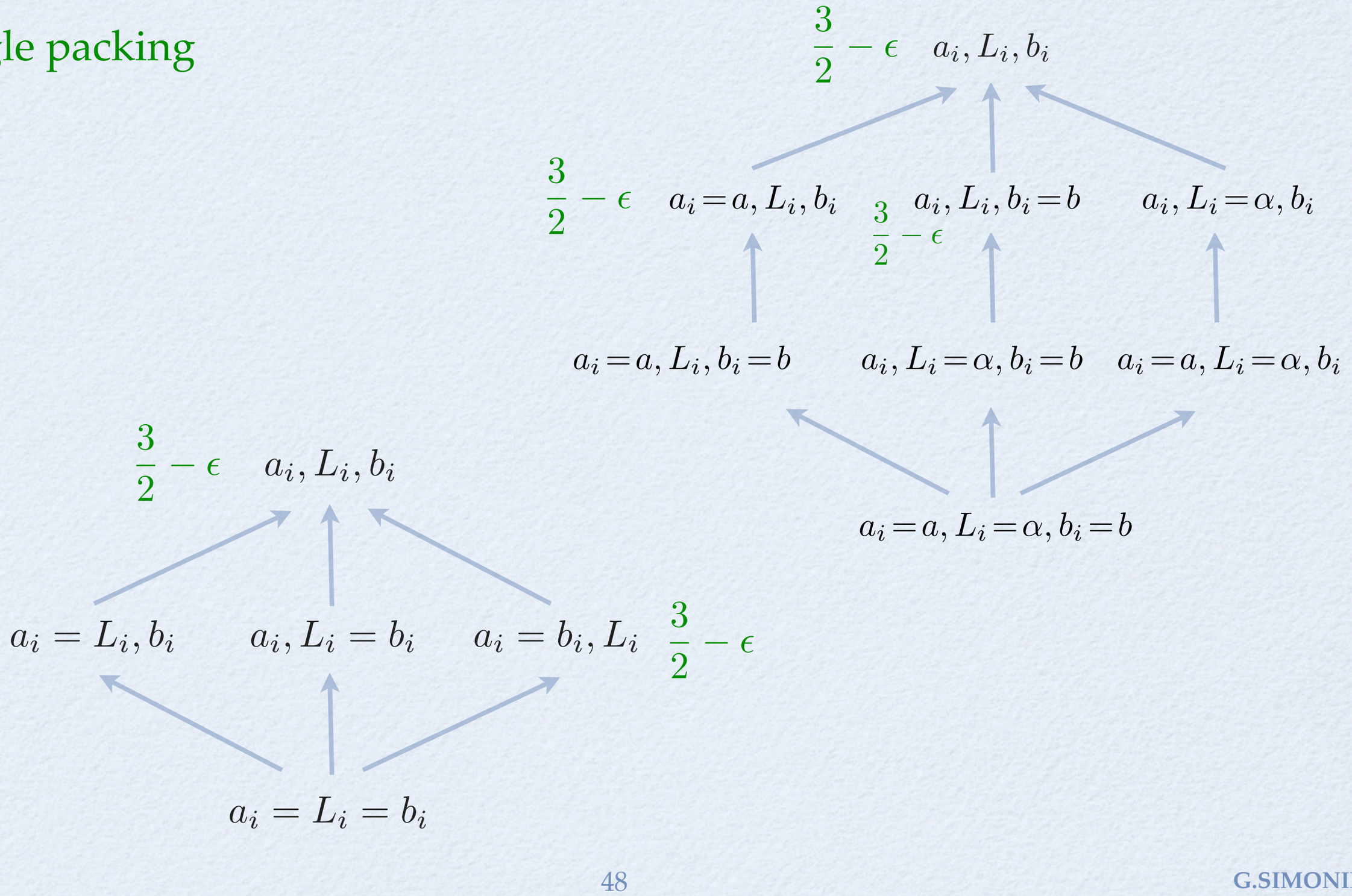


Vision globale des bornes obtenues



Vision globale des bornes obtenues

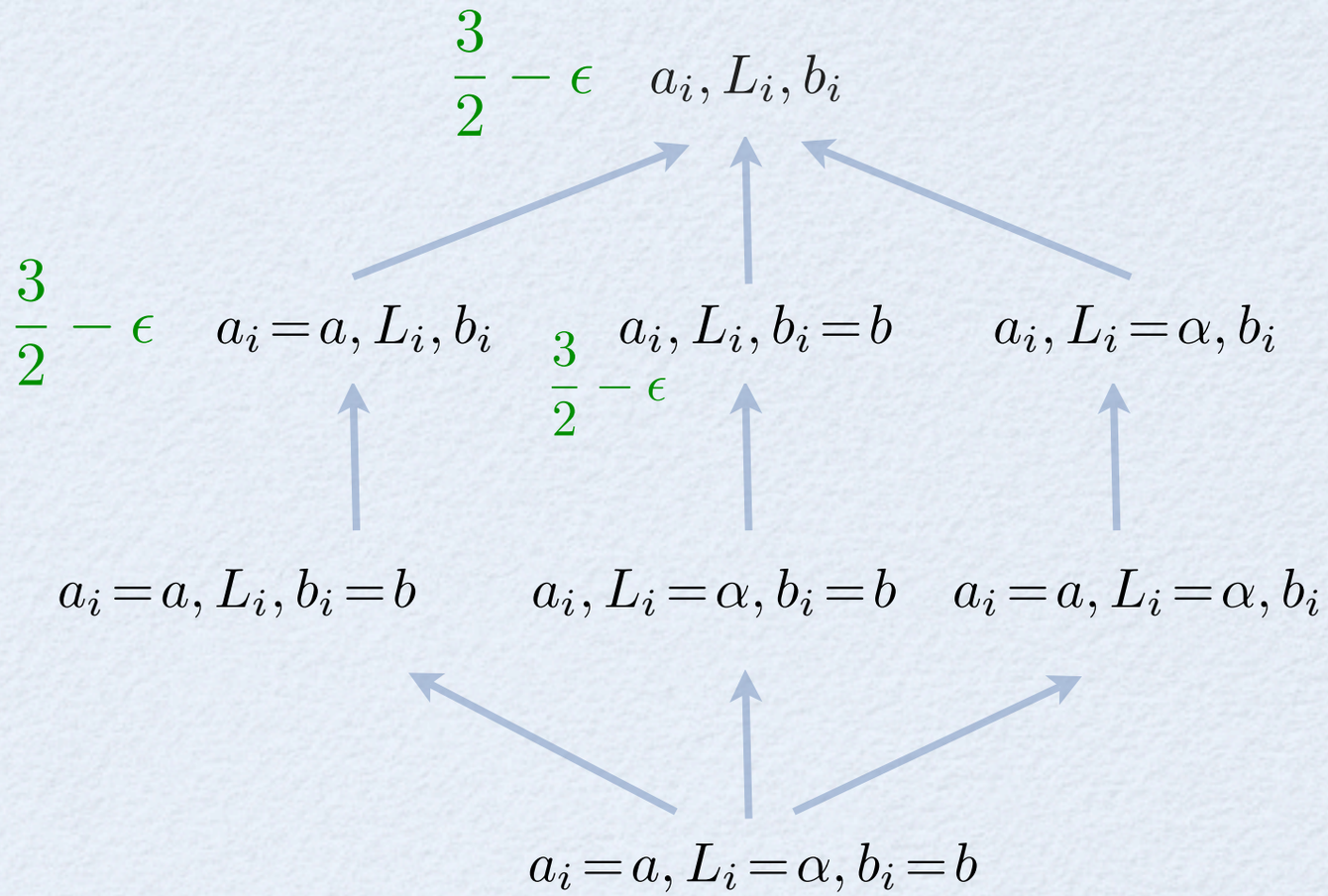
Triangle packing

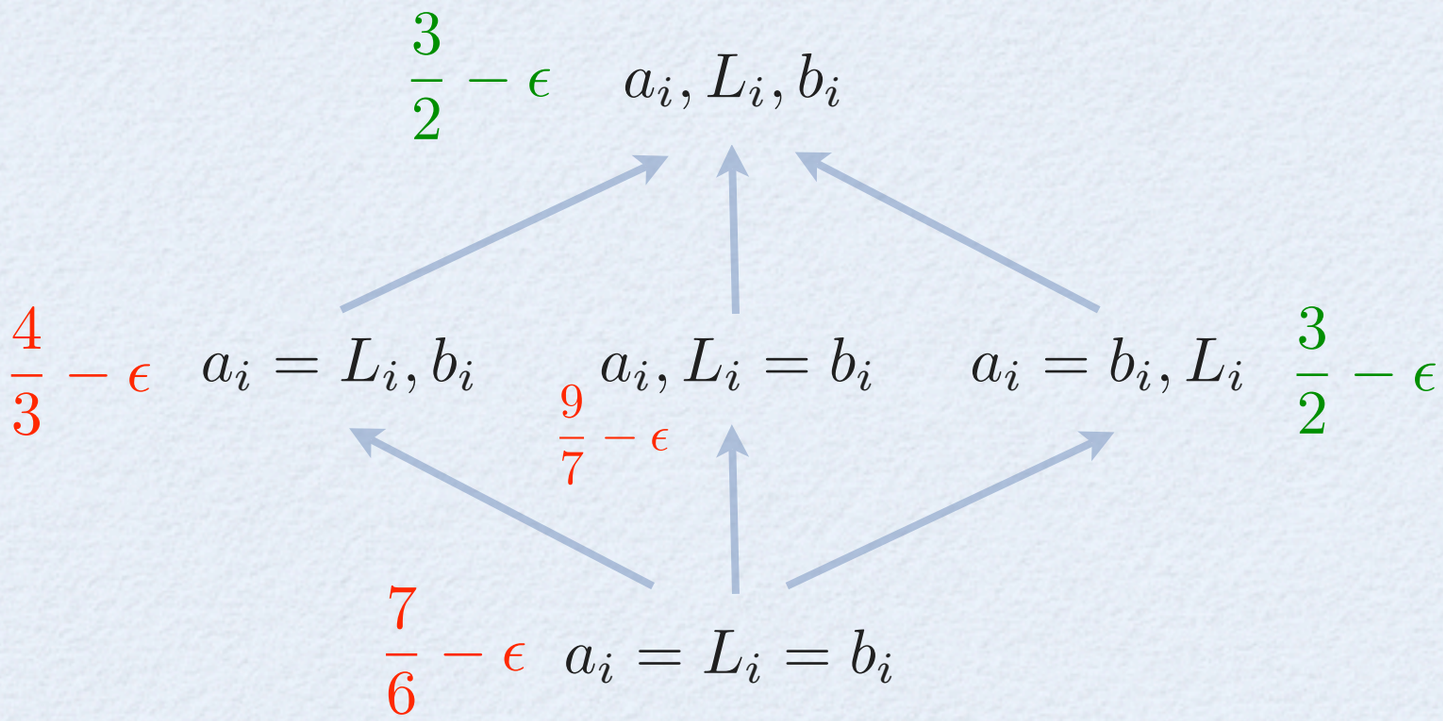


Vision globale des bornes obtenues

Triangle packing

Clique partition

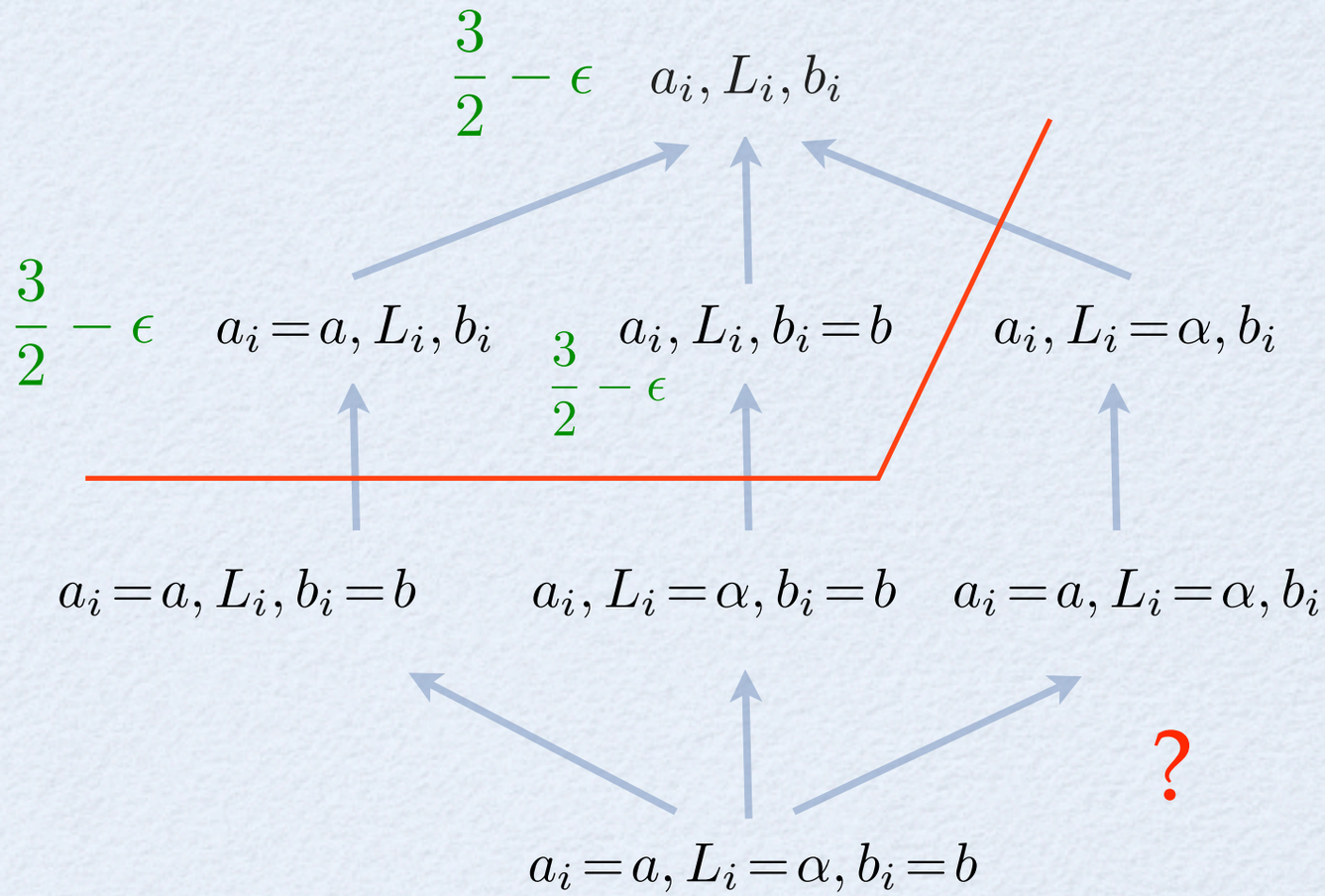




Vision globale des bornes obtenues

Triangle packing

Clique partition



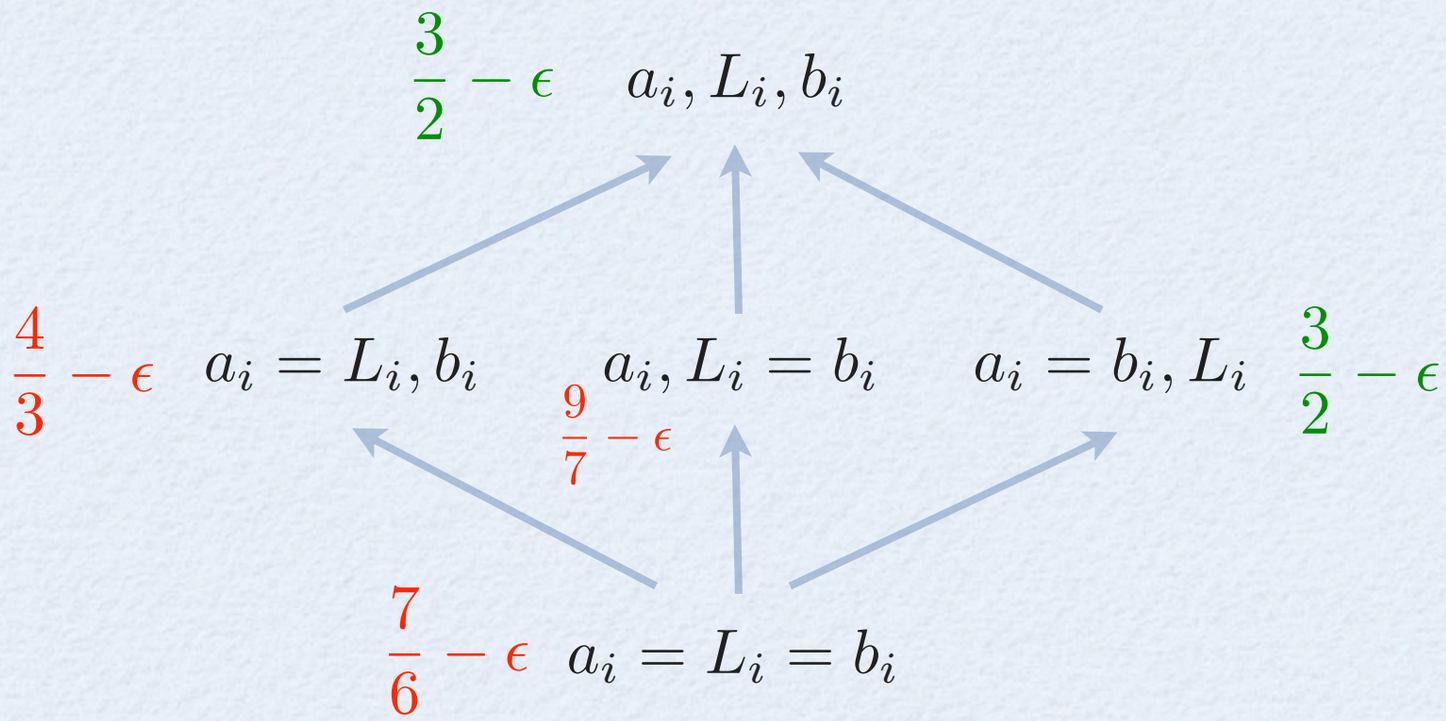


Table des matières

- Introduction
  - Présentation du problème
  - Modélisation
- Complexité de ces problèmes
  - Etat de l'art
  - Preuve de NP-complétude sur un cas particulier
- Les techniques d'approximation
  - Couplage maximum
  - Clique maximale
  - Poupées russes
- Les techniques de non approximation
  - Clique partition
  - Triangle packing
- Conclusion et perspectives

Conclusion

Conclusion

- Présentation d'un nouveau problème d'ordonnancement avec tâches-couplées

Conclusion

- Présentation d'un nouveau problème d'ordonnancement avec tâches-couplées
- Contrainte de compatibilité

Conclusion

- Présentation d'un nouveau problème d'ordonnancement avec tâches-couplées
- Contrainte de compatibilité
- Présentation de plusieurs résultats de $\mathcal{NP}$-complétude

Conclusion

- Présentation d'un nouveau problème d'ordonnancement avec tâches-couplées
- Contrainte de compatibilité
- Présentation de plusieurs résultats de $\mathcal{NP}$-complétude
- Présentation des trois heuristiques intéressantes :
  - Couplage maximum
  - Clique maximale
  - Poupées russes

Conclusion

- Présentation d'un nouveau problème d'ordonnancement avec tâches-couplées
- Contrainte de compatibilité
- Présentation de plusieurs résultats de $\mathcal{NP}$-complétude
- Présentation des trois heuristiques intéressantes :
  - Couplage maximum
  - Clique maximale
  - Poupées russes
- Présentation de deux méthodes de réduction pour avoir des bornes de non approximation :
  - Triangle Packing
  - Clique partition

Perspectives

Perspectives

- Trouver des bornes d'approximation à valeur constante

Perspectives

- Trouver des bornes d'approximation à valeur constante
- Continuer à étudier en rajoutant des paramètres :
  - Type de graphe de précédence particulier
  - des délais d'exécution

Perspectives

- Trouver des bornes d'approximation à valeur constante
- Continuer à étudier en rajoutant des paramètres :
  - Type de graphe de précédence particulier
  - des délais d'exécution
- Etudier ces problèmes en online

Perspectives

- Trouver des bornes d'approximation à valeur constante
- Continuer à étudier en rajoutant des paramètres :
  - Type de graphe de précédence particulier
  - des délais d'exécution
- Etudier ces problèmes en online
- Travailler sur des systèmes multiprocesseurs :
  - Rapprochement entre des clusters de machines et des flottilles

MERCI DE VOTRE

ATTENTION