

Minimisation du délai moyen : une contrainte globale

Camille Bonnin^{*,^o} Margaux Nattaf^{*} Arnaud Malapert^o
Marie-Laure Espinouse^{*}

^{*}Univ. Grenoble Alpes, CNRS, Grenoble INP, G-SCOP, 38000 Grenoble, France
{camille.bonnin, margaux.nattaf, marie-laure.espinouse}@grenoble-inp.fr

^oUniversité Côte d'Azur, CNRS, I3S, France

arnaud.malapert@univ-cotedazur.fr

24^e congrès de la ROADEF, 20-23 février 2023, Rennes

Objectif de la thèse

- Résolution de problèmes d'ordonnancement par la PPC
- Objectif général : améliorer la prise en compte en PPC de fonctions objectifs moins étudiées
- En particulier les fonctions objectifs de type " \sum "
- Travail actuel : porte sur le délai moyen ($\sum C_j$)

Objectif de la thèse

- Résolution de problèmes d'**ordonnement** par la **PPC**
- Objectif général : améliorer la prise en compte en PPC de **fonctions objectifs moins étudiées**
- En particulier les fonctions objectifs de type " \sum "
- Travail actuel : porte sur le **délai moyen** ($\sum C_j$)

Objectif de la thèse

- Résolution de problèmes d'**ordonnancement** par la PPC
- Objectif général : améliorer la prise en compte en PPC de **fonctions objectifs moins étudiées**
- En particulier les fonctions objectifs de type " \sum "
- Travail actuel : porte sur le **délai moyen** ($\sum C_j$)

Objectif de la thèse

- Résolution de problèmes d'**ordonnement** par la PPC
- Objectif général : améliorer la prise en compte en PPC de **fonctions objectifs moins étudiées**
- En particulier les fonctions objectifs de type " \sum "
- Travail actuel : porte sur le **délai moyen** ($\sum C_j$)

Objectif de la thèse

- Résolution de problèmes d'**ordonnement** par la **PPC**
- Objectif général : améliorer la prise en compte en PPC de **fonctions objectifs moins étudiées**
- En particulier les fonctions objectifs de type " \sum "
- Travail actuel : porte sur le **délai moyen** ($\sum C_j$)

Approche et contexte

- Approche choisie : définition d'une contrainte globale
- A notre connaissance : pas de contrainte globale spécifique à $\sum C_j$
- Une contrainte globale pour $\sum w_j C_j$ [Kovács et Beck, 2011]
- Motivation : leur travail utilise une **relaxation**, nous souhaitons en tester d'autres pour pouvoir comparer leurs performances pour notre contrainte

Approche et contexte

- Approche choisie : définition d'une contrainte globale
- A notre connaissance : pas de contrainte globale spécifique à $\sum C_j$
- Une contrainte globale pour $\sum w_j C_j$ [Kovács et Beck, 2011]
- Motivation : leur travail utilise une **relaxation**, nous souhaitons en tester d'autres pour pouvoir comparer leurs performances pour notre contrainte

Approche et contexte

- Approche choisie : définition d'une contrainte globale
- A notre connaissance : pas de contrainte globale spécifique à $\sum C_j$
- Une contrainte globale pour $\sum w_j C_j$ [Kovács et Beck, 2011]
- Motivation : leur travail utilise une **relaxation**, nous souhaitons en tester d'autres pour pouvoir comparer leurs performances pour notre contrainte

Approche et contexte

- Approche choisie : définition d'une contrainte globale
- A notre connaissance : pas de contrainte globale spécifique à $\sum C_j$
- Une contrainte globale pour $\sum w_j C_j$ [Kovács et Beck, 2011]
- Motivation : leur travail utilise une **relaxation**, nous souhaitons en tester d'autres pour pouvoir comparer leurs performances pour notre contrainte

Approche et contexte

- Approche choisie : définition d'une contrainte globale
- A notre connaissance : pas de contrainte globale spécifique à $\sum C_j$
- Une contrainte globale pour $\sum w_j C_j$ [Kovács et Beck, 2011]
- Motivation : leur travail utilise une **relaxation**, nous souhaitons en tester d'autres pour pouvoir comparer leurs performances pour notre contrainte

Notation de Graham

- Type de problème | contraintes | fonction objectif à minimiser
- 1 : problème à une machine
- p_j : durées (implicite)
- r_j : dates de disponibilité (release dates)
- d_j : dates butoires (deadlines)
- w_j : poids
- $pmtn$: autorisation de préempter les tâches
- $\sum C_j$: délai moyen (flowtime)
- $\sum w_j M_j$: temps d'occupation moyen pondéré (weighted mean busy time)

Notation de Graham

- Type de problème | contraintes | fonction objectif à minimiser
- 1 : problème à une machine
- p_j : durées (implicite)
- r_j : dates de disponibilité (release dates)
- d_j : dates butoires (deadlines)
- w_j : poids
- $pmtn$: autorisation de préempter les tâches
- $\sum C_j$: délai moyen (flowtime)
- $\sum w_j M_j$: temps d'occupation moyen pondéré (weighted mean busy time)

Notation de Graham

- Type de problème | contraintes | fonction objectif à minimiser
- 1 : problème à une machine
- p_j : durées (implicite)
- r_j : dates de disponibilité (release dates)
- d_j : dates butoires (deadlines)
- w_j : poids
- $pmtn$: autorisation de préempter les tâches
- $\sum C_j$: délai moyen (flowtime)
- $\sum w_j M_j$: temps d'occupation moyen pondéré (weighted mean busy time)

Notation de Graham

- Type de problème | contraintes | fonction objectif à minimiser
- 1 : problème à une machine
- p_j : durées (implicite)
- r_j : dates de disponibilité (release dates)
- d_j : dates butoires (deadlines)
- w_j : poids
- $pmtn$: autorisation de préempter les tâches
- $\sum C_j$: délai moyen (flowtime)
- $\sum w_j M_j$: temps d'occupation moyen pondéré (weighted mean busy time)

Notation de Graham

- Type de problème | contraintes | fonction objectif à minimiser
- 1 : problème à une machine
- p_j : durées (implicite)
- r_j : dates de disponibilité (release dates)
- d_j : dates butoires (deadlines)
- w_j : poids
- $pmtn$: autorisation de préempter les tâches
- $\sum C_j$: délai moyen (flowtime)
- $\sum w_j M_j$: temps d'occupation moyen pondéré (weighted mean busy time)

Notation de Graham

- Type de problème | contraintes | fonction objectif à minimiser
- 1 : problème à une machine
- p_j : durées (implicite)
- r_j : dates de disponibilité (release dates)
- d_j : dates butoires (deadlines)
- w_j : poids
- $pmtn$: autorisation de préempter les tâches
- $\sum C_j$: délai moyen (flowtime)
- $\sum w_j M_j$: temps d'occupation moyen pondéré (weighted mean busy time)

Notation de Graham

- Type de problème | contraintes | fonction objectif à minimiser
- 1 : problème à une machine
- p_j : durées (implicite)
- r_j : dates de disponibilité (release dates)
- d_j : dates butoires (deadlines)
- w_j : poids
- $pmtn$: autorisation de préempter les tâches
- $\sum C_j$: délai moyen (flowtime)
- $\sum w_j M_j$: temps d'occupation moyen pondéré (weighted mean busy time)

Notation de Graham

- Type de problème | contraintes | fonction objectif à minimiser
- 1 : problème à une machine
- p_j : durées (implicite)
- r_j : dates de disponibilité (release dates)
- d_j : dates butoires (deadlines)
- w_j : poids
- $pmtn$: autorisation de préempter les tâches
- $\sum C_j$: délai moyen (flowtime)
- $\sum w_j M_j$: temps d'occupation moyen pondéré (weighted mean busy time)

Notation de Graham

- Type de problème | contraintes | fonction objectif à minimiser
- 1 : problème à une machine
- p_j : durées (implicite)
- r_j : dates de disponibilité (release dates)
- d_j : dates butoires (deadlines)
- w_j : poids
- $pmtn$: autorisation de préempter les tâches
- $\sum C_j$: délai moyen (flowtime)
- $\sum w_j M_j$: temps d'occupation moyen pondéré (weighted mean busy time)

Notation de Graham

- Type de problème | contraintes | fonction objectif à minimiser
- 1 : problème à une machine
- p_j : durées (implicite)
- r_j : dates de disponibilité (release dates)
- d_j : dates butoires (deadlines)
- w_j : poids
- $pmtn$: autorisation de préempter les tâches
- $\sum C_j$: délai moyen (flowtime)
- $\sum w_j M_j$: temps d'occupation moyen pondéré (weighted mean busy time)

Choix du problème pour la contrainte globale

- $1|r_j; d_j| \sum C_j$
- Générique et adapté à la PPC
- \mathcal{NP} -difficile [Lenstra, Rinnooy Kan et Brucker, 1977]

Utilisation de bornes inférieures

- Filtrage basé sur les coûts : utilisation de bornes inf
- Ordonnancement : bornes inf \rightarrow relaxations
- Laquelle choisir ? Besoin de comparer \rightarrow schéma de contrainte qui permet de changer la relaxation

Choix du problème pour la contrainte globale

- $1|r_j; d_j| \sum C_j$
- Générique et adapté à la PPC
- \mathcal{NP} -difficile [Lenstra, Rinnooy Kan et Brucker, 1977]

Utilisation de bornes inférieures

- Filtrage basé sur les coûts : utilisation de bornes inf
- Ordonnancement : bornes inf \rightarrow relaxations
- Laquelle choisir ? Besoin de comparer \rightarrow schéma de contrainte qui permet de changer la relaxation

Choix du problème pour la contrainte globale

- $1|r_j; d_j| \sum C_j$
- Générique et adapté à la PPC
- \mathcal{NP} -difficile [Lenstra, Rinnooy Kan et Brucker, 1977]

Utilisation de bornes inférieures

- Filtrage basé sur les coûts : utilisation de bornes inf
- Ordonnancement : bornes inf \rightarrow relaxations
- Laquelle choisir ? Besoin de comparer \rightarrow schéma de contrainte qui permet de changer la relaxation

Choix du problème pour la contrainte globale

- $1|r_j; d_j| \sum C_j$
- Générique et adapté à la PPC
- \mathcal{NP} -difficile [Lenstra, Rinnooy Kan et Brucker, 1977]

Utilisation de bornes inférieures

- Filtrage basé sur les coûts : utilisation de bornes inf
- Ordonnancement : bornes inf \rightarrow relaxations
- Laquelle choisir ? Besoin de comparer \rightarrow schéma de contrainte qui permet de changer la relaxation

Choix du problème pour la contrainte globale

- $1|r_j; d_j| \sum C_j$
- Générique et adapté à la PPC
- \mathcal{NP} -difficile [Lenstra, Rinnooy Kan et Brucker, 1977]

Utilisation de bornes inférieures

- Filtrage basé sur les coûts : utilisation de bornes inf
- Ordonnement : bornes inf \rightarrow relaxations
- Laquelle choisir ? Besoin de comparer \rightarrow schéma de contrainte qui permet de changer la relaxation

Choix du problème pour la contrainte globale

- $1|r_j; d_j| \sum C_j$
- Générique et adapté à la PPC
- \mathcal{NP} -difficile [Lenstra, Rinnooy Kan et Brucker, 1977]

Utilisation de bornes inférieures

- Filtrage basé sur les coûts : utilisation de bornes inf
- Ordonnancement : bornes inf \rightarrow relaxations
- Laquelle choisir ? Besoin de comparer \rightarrow schéma de contrainte qui permet de changer la relaxation

Choix du problème pour la contrainte globale

- $1|r_j; d_j| \sum C_j$
- Générique et adapté à la PPC
- \mathcal{NP} -difficile [Lenstra, Rinnooy Kan et Brucker, 1977]

Utilisation de bornes inférieures

- Filtrage basé sur les coûts : utilisation de bornes inf
- Ordonnancement : bornes inf \rightarrow relaxations
- Laquelle choisir ? Besoin de comparer \rightarrow schéma de contrainte qui permet de changer la relaxation

Choix du problème pour la contrainte globale

- $1|r_j; d_j| \sum C_j$
- Générique et adapté à la PPC
- \mathcal{NP} -difficile [Lenstra, Rinnooy Kan et Brucker, 1977]

Utilisation de bornes inférieures

- Filtrage basé sur les coûts : utilisation de bornes inf
- Ordonnancement : bornes inf \rightarrow relaxations
- Laquelle choisir ? Besoin de comparer \rightarrow schéma de contrainte qui permet de changer la relaxation

Schéma de la contrainte (pour une relaxation fixée)

- Mise à jour de $\sum C_j$
- Filtrage des bornes des domaines des dates de début (S_j) des tâches

Schéma de la contrainte (pour une relaxation fixée)

- Mise à jour de $\sum C_j$
- Filtrage des bornes des domaines des dates de début (S_j) des tâches

Schéma de la contrainte (pour une relaxation fixée)

- Mise à jour de $\sum C_j$
- Filtrage des bornes des domaines des dates de début (S_j) des tâches

Relaxations polynomiales identifiées [ROADEF 2022]

- $1|r_j; pmtn| \sum C_j$ (résolue avec la règle de Smith modifiée [Brucker, 2006])
- $1|r_j; pmtn| \sum (w_j)M_j$ (utilisée par Kovács et Beck dans sa version pondérée) [Goemans, Queyranne, Schulz, Skutella et Wang, 2002]
- $1|r_j; pmtn; MP| \sum C_j$ [PMS, 2022]
- $1|r_j; pmtn; MP| \sum (w_j)M_j$
- $1|d_j| \sum C_j$ [Chen, Potts et Woeginger, 1998]
- $1|sp-graph| \sum w_j C_j$ [Lawler, 1978]

Relaxations polynomiales identifiées [ROADEF 2022]

- $1|r_j; pmtn| \sum C_j$ (résolue avec la règle de Smith modifiée [Brucker, 2006])
- $1|r_j; pmtn| \sum (w_j)M_j$ (utilisée par Kovács et Beck dans sa version pondérée) [Goemans, Queyranne, Schulz, Skutella et Wang, 2002]
- $1|r_j; pmtn; MP| \sum C_j$ [PMS, 2022]
- $1|r_j; pmtn; MP| \sum (w_j)M_j$
- $1|d_j| \sum C_j$ [Chen, Potts et Woeginger, 1998]
- $1|sp-graph| \sum w_j C_j$ [Lawler, 1978]

Relaxations polynomiales identifiées [ROADEF 2022]

- $1|r_j; pmtn| \sum C_j$ (résolue avec la règle de Smith modifiée [Brucker, 2006])
- $1|r_j; pmtn| \sum (w_j)M_j$ (utilisée par Kovács et Beck dans sa version pondérée) [Goemans, Queyranne, Schulz, Skutella et Wang, 2002]
- $1|r_j; pmtn; MP| \sum C_j$ [PMS, 2022]
- $1|r_j; pmtn; MP| \sum (w_j)M_j$
- $1|d_j| \sum C_j$ [Chen, Potts et Woeginger, 1998]
- $1|sp-graph| \sum w_j C_j$ [Lawler, 1978]

Relaxations polynomiales identifiées [ROADEF 2022]

- $1|r_j; pmtn| \sum C_j$ (résolue avec la règle de Smith modifiée [Brucker, 2006])
- $1|r_j; pmtn| \sum (w_j)M_j$ (utilisée par Kovács et Beck dans sa version pondérée) [Goemans, Queyranne, Schulz, Skutella et Wang, 2002]
- $1|r_j; pmtn; MP| \sum C_j$ [PMS, 2022]
- $1|r_j; pmtn; MP| \sum (w_j)M_j$
- $1|d_j| \sum C_j$ [Chen, Potts et Woeginger, 1998]
- $1|sp-graph| \sum w_j C_j$ [Lawler, 1978]

Relaxations polynomiales identifiées [ROADEF 2022]

- $1|r_j; pmtn| \sum C_j$ (résolue avec la règle de Smith modifiée [Brucker, 2006])
- $1|r_j; pmtn| \sum (w_j)M_j$ (utilisée par Kovács et Beck dans sa version pondérée) [Goemans, Queyranne, Schulz, Skutella et Wang, 2002]
- $1|r_j; pmtn; MP| \sum C_j$ [PMS, 2022]
- $1|r_j; pmtn; MP| \sum (w_j)M_j$
- $1|d_j| \sum C_j$ [Chen, Potts et Woeginger, 1998]
- $1|sp-graph| \sum w_j C_j$ [Lawler, 1978]

Relaxations polynomiales identifiées [ROADEF 2022]

- $1|r_j; pmtn| \sum C_j$ (résolue avec la règle de Smith modifiée [Brucker, 2006])
- $1|r_j; pmtn| \sum (w_j)M_j$ (utilisée par Kovács et Beck dans sa version pondérée) [Goemans, Queyranne, Schulz, Skutella et Wang, 2002]
- $1|r_j; pmtn; MP| \sum C_j$ [PMS, 2022]
- $1|r_j; pmtn; MP| \sum (w_j)M_j$
- $1|d_j| \sum C_j$ [Chen, Potts et Woeginger, 1998]
- $1|sp-graph| \sum w_j C_j$ [Lawler, 1978]

Relaxations polynomiales identifiées [ROADEF 2022]

- $1|r_j; pmtn| \sum C_j$ (résolue avec la règle de Smith modifiée [Brucker, 2006])
- $1|r_j; pmtn| \sum (w_j)M_j$ (utilisée par Kovács et Beck dans sa version pondérée) [Goemans, Queyranne, Schulz, Skutella et Wang, 2002]
- $1|r_j; pmtn; MP| \sum C_j$ [PMS, 2022]
- $1|r_j; pmtn; MP| \sum (w_j)M_j$
- $1|d_j| \sum C_j$ [Chen, Potts et Woeginger, 1998]
- $1|sp-graph| \sum w_j C_j$ [Lawler, 1978]

Algo 1 Mise à jour de la borne inf de $\sum C_j$

- 1: $relax \leftarrow$ valeur de l'objectif obtenue en résolvant la relaxation ;
 - 2: $\underline{\sum C_j} \leftarrow relax$;
-

NB : Si $\underline{\sum C_j} > \overline{\sum C_j}$, la contrainte échouera automatiquement.

Algo 2 Mise à jour de la borne inf de $\sum C_j$

- 1: $relax \leftarrow$ valeur de l'objectif obtenue en résolvant la relaxation ;
 - 2: $\underline{\sum C_j} \leftarrow relax$;
-

NB : Si $\underline{\sum C_j} > \overline{\sum C_j}$, la contrainte échouera automatiquement.

Algo 3 Mise à jour de la borne inf de $\sum C_j$

- 1: $relax \leftarrow$ valeur de l'objectif obtenue en résolvant la relaxation ;
 - 2: $\underline{\sum C_j} \leftarrow relax$;
-

NB : Si $\underline{\sum C_j} > \overline{\sum C_j}$, la contrainte échouera automatiquement.

Algo 4 Mise à jour de la borne inf de $\sum C_j$

- 1: $relax \leftarrow$ valeur de l'objectif obtenue en résolvant la relaxation ;
 - 2: $\underline{\sum C_j} \leftarrow relax$;
-

NB : Si $\underline{\sum C_j} > \overline{\sum C_j}$, la contrainte échouera automatiquement.

Algo 5 Filtrage des bornes inf des S_j

- 1: Pour toutes les tâches i , faire :
 - 2: $t \leftarrow \underline{S}_i$;
 - 3: Tant que $t \leq \overline{S}_i$, faire :
 - 4: $relax_i \leftarrow$ valeur de l'objectif obtenu en résolvant la
relaxation en fixant la tâche i dans $[t, t + p_i]$;
 - 5: Si $relax_i > \sum \overline{C}_j$, alors faire :
 - 6: $t \leftarrow t + 1$;
 - 7: Sinon, faire :
 - 8: Sortir du tant que ; (support trouvé)
 - 9: $\underline{S}_i \leftarrow t$;
-

L'algo de filtrage des bornes sup S_j est similaire.

Algo 6 Filtrage des bornes inf des S_j

- 1: **Pour toutes** les tâches i , **faire** :
 - 2: $t \leftarrow \underline{S}_i$;
 - 3: **Tant que** $t \leq \overline{S}_i$, **faire** :
 - 4: $relax_i \leftarrow$ valeur de l'objectif obtenu en résolvant la
relaxation en fixant la tâche i dans $[t, t + p_i]$;
 - 5: **Si** $relax_i > \sum C_j$, **alors faire** :
 - 6: $t \leftarrow t + 1$;
 - 7: **Sinon, faire** :
 - 8: **Sortir du tant que** ; (support trouvé)
 - 9: $\underline{S}_i \leftarrow t$;
-

L'algo de filtrage des bornes sup S_j est similaire.

Algo 7 Filtrage des bornes inf des S_j

- 1: **Pour toutes** les tâches i , **faire** :
 - 2: $t \leftarrow \underline{S}_i$;
 - 3: **Tant que** $t \leq \overline{S}_i$, **faire** :
 - 4: $relax_i \leftarrow$ valeur de l'objectif obtenu en résolvant la
relaxation en fixant la tâche i dans $[t, t + p_i]$;
 - 5: **Si** $relax_i > \sum \overline{C}_j$, **alors faire** :
 - 6: $t \leftarrow t + 1$;
 - 7: **Sinon, faire** :
 - 8: **Sortir du tant que** ; (support trouvé)
 - 9: $\underline{S}_i \leftarrow t$;
-

L'algo de filtrage des bornes sup S_j est similaire.

Algo 8 Filtrage des bornes inf des S_j

- 1: **Pour toutes** les taches i , **faire** :
 - 2: $t \leftarrow \underline{S}_i$;
 - 3: **Tant que** $t \leq \overline{S}_i$, **faire** :
 - 4: $relax_i \leftarrow$ valeur de l'objectif obtenu en résolvant la
relaxation en fixant la tâche i dans $[t, t + p_i]$;
 - 5: **Si** $relax_i > \sum \overline{C}_j$, **alors faire** :
 - 6: $t \leftarrow t + 1$;
 - 7: **Sinon, faire** :
 - 8: **Sortir du tant que** ; (support trouvé)
 - 9: $\underline{S}_i \leftarrow t$;
-

L'algo de filtrage des bornes sup S_j est similaire.

Algo 9 Filtrage des bornes inf des S_j

- 1: **Pour toutes** les tâches i , **faire** :
 - 2: $t \leftarrow \underline{S}_i$;
 - 3: **Tant que** $t \leq \overline{S}_i$, **faire** :
 - 4: $relax_i \leftarrow$ valeur de l'objectif obtenu en résolvant la
relaxation en fixant la tâche i dans $[t, t + p_i]$;
 - 5: **Si** $relax_i > \sum \overline{C}_j$, **alors faire** :
 - 6: $t \leftarrow t + 1$;
 - 7: **Sinon, faire** :
 - 8: **Sortir du tant que** ; (support trouvé)
 - 9: $\underline{S}_i \leftarrow t$;
-

L'algo de filtrage des bornes sup S_j est similaire.

Algo 10 Filtrage des bornes inf des S_j

- 1: **Pour toutes** les tâches i , **faire** :
 - 2: $t \leftarrow \underline{S}_i$;
 - 3: **Tant que** $t \leq \overline{S}_i$, **faire** :
 - 4: $relax_i \leftarrow$ valeur de l'objectif obtenu en résolvant la
relaxation en fixant la tâche i dans $[t, t + p_i]$;
 - 5: **Si** $relax_i > \sum C_j$, **alors faire** :
 - 6: $t \leftarrow t + 1$;
 - 7: **Sinon, faire** :
 - 8: **Sortir du tant que** ; (support trouvé)
 - 9: $\underline{S}_i \leftarrow t$;
-

L'algo de filtrage des bornes sup S_j est similaire.

Algo 11 Filtrage des bornes inf des S_j

- 1: **Pour toutes** les tâches i , **faire** :
 - 2: $t \leftarrow \underline{S}_i$;
 - 3: **Tant que** $t \leq \overline{S}_i$, **faire** :
 - 4: $relax_i \leftarrow$ valeur de l'objectif obtenu en résolvant la
relaxation en fixant la tâche i dans $[t, t + p_i]$;
 - 5: **Si** $relax_i > \sum C_j$, **alors faire** :
 - 6: $t \leftarrow t + 1$;
 - 7: **Sinon, faire** :
 - 8: **Sortir du tant que** ; (support trouvé)
 - 9: $\underline{S}_i \leftarrow t$;
-

L'algo de filtrage des bornes sup S_j est similaire.

Algo 12 Filtrage des bornes inf des S_j

- 1: **Pour toutes** les tâches i , **faire** :
 - 2: $t \leftarrow \underline{S}_i$;
 - 3: **Tant que** $t \leq \overline{S}_i$, **faire** :
 - 4: $relax_i \leftarrow$ valeur de l'objectif obtenu en résolvant la
relaxation en fixant la tâche i dans $[t, t + p_i]$;
 - 5: **Si** $relax_i > \sum C_j$, **alors faire** :
 - 6: $t \leftarrow t + 1$;
 - 7: **Sinon, faire** :
 - 8: Sortir du tant que ; (support trouvé)
 - 9: $\underline{S}_i \leftarrow t$;
-

L'algo de filtrage des bornes sup S_j est similaire.

Algo 13 Filtrage des bornes inf des S_j

- 1: **Pour toutes** les tâches i , **faire** :
 - 2: $t \leftarrow \underline{S}_i$;
 - 3: **Tant que** $t \leq \overline{S}_i$, **faire** :
 - 4: $relax_i \leftarrow$ valeur de l'objectif obtenu en résolvant la
relaxation en fixant la tâche i dans $[t, t + p_i]$;
 - 5: **Si** $relax_i > \sum C_j$, **alors faire** :
 - 6: $t \leftarrow t + 1$;
 - 7: **Sinon, faire** :
 - 8: **Sortir du tant que** ; (support trouvé)
 - 9: $\underline{S}_i \leftarrow t$;
-

L'algo de filtrage des bornes sup S_j est similaire.

Algo 14 Filtrage des bornes inf des S_j

- 1: **Pour toutes** les tâches i , **faire** :
 - 2: $t \leftarrow \underline{S}_i$;
 - 3: **Tant que** $t \leq \overline{S}_i$, **faire** :
 - 4: $relax_i \leftarrow$ valeur de l'objectif obtenu en résolvant la
relaxation en fixant la tâche i dans $[t, t + p_i]$;
 - 5: **Si** $relax_i > \sum C_j$, **alors faire** :
 - 6: $t \leftarrow t + 1$;
 - 7: **Sinon, faire** :
 - 8: **Sortir du tant que** ; (support trouvé)
 - 9: $\underline{S}_i \leftarrow t$;
-

L'algo de filtrage des bornes sup S_j est similaire.

Algo 15 Filtrage des bornes inf des S_j

- 1: **Pour toutes** les tâches i , **faire** :
 - 2: $t \leftarrow \underline{S}_i$;
 - 3: **Tant que** $t \leq \overline{S}_i$, **faire** :
 - 4: $relax_i \leftarrow$ valeur de l'objectif obtenu en résolvant la
relaxation en fixant la tâche i dans $[t, t + p_i]$;
 - 5: **Si** $relax_i > \sum C_j$, **alors faire** :
 - 6: $t \leftarrow t + 1$;
 - 7: **Sinon, faire** :
 - 8: **Sortir du tant que** ; (support trouvé)
 - 9: $\underline{S}_i \leftarrow t$;
-

L'algo de filtrage des bornes sup S_j est similaire.

Protocole

- CP Optimizer 22.1.0 et machine Linux
- Instances lancées sur 1 seul cœur
- Instances de Pan-Shi (pour $1|r_j; d_j| \sum C_j$) [Pan et Shi, 2008]
- 900 instances de 20 à 100 tâches
- Limite de temps : 10 min par instance

Protocole

- CP Optimizer 22.1.0 et machine Linux
- Instances lancées sur 1 seul cœur
- Instances de Pan-Shi (pour $1|r_j; d_j| \sum C_j$) [Pan et Shi, 2008]
- 900 instances de 20 à 100 tâches
- Limite de temps : 10 min par instance

Protocole

- CP Optimizer 22.1.0 et machine Linux
- Instances lancées sur 1 seul cœur
- Instances de Pan-Shi (pour $1|r_j; d_j| \sum C_j$) [Pan et Shi, 2008]
- 900 instances de 20 à 100 tâches
- Limite de temps : 10 min par instance

Protocole

- CP Optimizer 22.1.0 et machine Linux
- Instances lancées sur 1 seul cœur
- Instances de Pan-Shi (pour $1|r_j; d_j| \sum C_j$) [Pan et Shi, 2008]
- 900 instances de 20 à 100 tâches
- Limite de temps : 10 min par instance

Protocole

- CP Optimizer 22.1.0 et machine Linux
- Instances lancées sur 1 seul cœur
- Instances de Pan-Shi (pour $1|r_j; d_j| \sum C_j$) [Pan et Shi, 2008]
- 900 instances de 20 à 100 tâches
- Limite de temps : 10 min par instance

Protocole

- CP Optimizer 22.1.0 et machine Linux
- Instances lancées sur 1 seul cœur
- Instances de Pan-Shi (pour $1|r_j; d_j| \sum C_j$) [Pan et Shi, 2008]
- 900 instances de 20 à 100 tâches
- Limite de temps : 10 min par instance

Modèles testés (7 au total)

| Nom | Relaxation | Filtrage |
|-----------------|---------------------------------|----------|
| <i>none</i> | aucune | non |
| <i>pmtnFlow</i> | $1 r_j; pmtn \sum C_j$ | non |
| <i>pmtnWbmt</i> | $1 r_j; pmtn \sum w_j M_j$ | non |
| <i>mandFlow</i> | $1 r_j; pmtn; MP \sum C_j$ | non |
| <i>mandWbmt</i> | $1 r_j; pmtn; MP \sum w_j M_j$ | non |
| <i>filtFlow</i> | $1 r_j; pmtn \sum C_j$ | oui |
| <i>filtWbmt</i> | $1 r_j; pmtn \sum w_j M_j$ | oui |

Modèles testés (7 au total)

| Nom | Relaxation | Filtrage |
|-----------------|---------------------------------|----------|
| <i>none</i> | aucune | non |
| <i>pmtnFlow</i> | $1 r_j; pmtn \sum C_j$ | non |
| <i>pmtnWbmt</i> | $1 r_j; pmtn \sum w_j M_j$ | non |
| <i>mandFlow</i> | $1 r_j; pmtn; MP \sum C_j$ | non |
| <i>mandWbmt</i> | $1 r_j; pmtn; MP \sum w_j M_j$ | non |
| <i>filtFlow</i> | $1 r_j; pmtn \sum C_j$ | oui |
| <i>filtWbmt</i> | $1 r_j; pmtn \sum w_j M_j$ | oui |

Modèles testés (7 au total)

| Nom | Relaxation | Filtrage |
|-----------------|---------------------------------|----------|
| <i>none</i> | aucune | non |
| <i>pmtnFlow</i> | $1 r_j; pmtn \sum C_j$ | non |
| <i>pmtnWbmt</i> | $1 r_j; pmtn \sum w_j M_j$ | non |
| <i>mandFlow</i> | $1 r_j; pmtn; MP \sum C_j$ | non |
| <i>mandWbmt</i> | $1 r_j; pmtn; MP \sum w_j M_j$ | non |
| <i>filtFlow</i> | $1 r_j; pmtn \sum C_j$ | oui |
| <i>filtWbmt</i> | $1 r_j; pmtn \sum w_j M_j$ | oui |

Modèles testés (7 au total)

| Nom | Relaxation | Filtrage |
|-----------------|---------------------------------|----------|
| <i>none</i> | aucune | non |
| <i>pmtnFlow</i> | $1 r_j; pmtn \sum C_j$ | non |
| <i>pmtnWbmt</i> | $1 r_j; pmtn \sum w_j M_j$ | non |
| <i>mandFlow</i> | $1 r_j; pmtn; MP \sum C_j$ | non |
| <i>mandWbmt</i> | $1 r_j; pmtn; MP \sum w_j M_j$ | non |
| <i>filtFlow</i> | $1 r_j; pmtn \sum C_j$ | oui |
| <i>filtWbmt</i> | $1 r_j; pmtn \sum w_j M_j$ | oui |

Modèles testés (7 au total)

| Nom | Relaxation | Filtrage |
|-----------------|---------------------------------|----------|
| <i>none</i> | aucune | non |
| <i>pmtnFlow</i> | $1 r_j; pmtn \sum C_j$ | non |
| <i>pmtnWbmt</i> | $1 r_j; pmtn \sum w_j M_j$ | non |
| <i>mandFlow</i> | $1 r_j; pmtn; MP \sum C_j$ | non |
| <i>mandWbmt</i> | $1 r_j; pmtn; MP \sum w_j M_j$ | non |
| <i>filtFlow</i> | $1 r_j; pmtn \sum C_j$ | oui |
| <i>filtWbmt</i> | $1 r_j; pmtn \sum w_j M_j$ | oui |

Modèles testés (7 au total)

| Nom | Relaxation | Filtrage |
|-----------------|---------------------------------|----------|
| <i>none</i> | aucune | non |
| <i>pmtnFlow</i> | $1 r_j; pmtn \sum C_j$ | non |
| <i>pmtnWbmt</i> | $1 r_j; pmtn \sum w_j M_j$ | non |
| <i>mandFlow</i> | $1 r_j; pmtn; MP \sum C_j$ | non |
| <i>mandWbmt</i> | $1 r_j; pmtn; MP \sum w_j M_j$ | non |
| <i>filtFlow</i> | $1 r_j; pmtn \sum C_j$ | oui |
| <i>filtWbmt</i> | $1 r_j; pmtn \sum w_j M_j$ | oui |

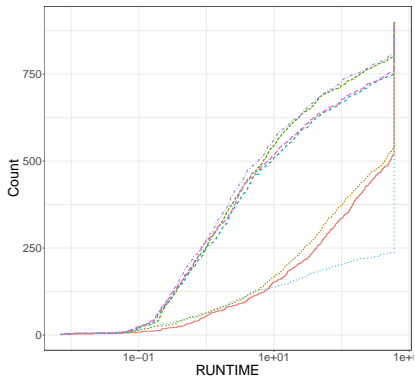
Modèles testés (7 au total)

| Nom | Relaxation | Filtrage |
|-----------------|---------------------------------|----------|
| <i>none</i> | aucune | non |
| <i>pmtnFlow</i> | $1 r_j; pmtn \sum C_j$ | non |
| <i>pmtnWbmt</i> | $1 r_j; pmtn \sum w_j M_j$ | non |
| <i>mandFlow</i> | $1 r_j; pmtn; MP \sum C_j$ | non |
| <i>mandWbmt</i> | $1 r_j; pmtn; MP \sum w_j M_j$ | non |
| <i>filtFlow</i> | $1 r_j; pmtn \sum C_j$ | oui |
| <i>filtWbmt</i> | $1 r_j; pmtn \sum w_j M_j$ | oui |

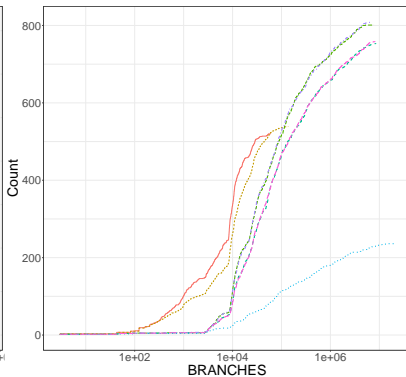
Modèles testés (7 au total)

| Nom | Relaxation | Filtrage |
|-----------------|---------------------------------|----------|
| <i>none</i> | aucune | non |
| <i>pmtnFlow</i> | $1 r_j; pmtn \sum C_j$ | non |
| <i>pmtnWbmt</i> | $1 r_j; pmtn \sum w_j M_j$ | non |
| <i>mandFlow</i> | $1 r_j; pmtn; MP \sum C_j$ | non |
| <i>mandWbmt</i> | $1 r_j; pmtn; MP \sum w_j M_j$ | non |
| <i>filtFlow</i> | $1 r_j; pmtn \sum C_j$ | oui |
| <i>filtWbmt</i> | $1 r_j; pmtn \sum w_j M_j$ | oui |

Résultats 1 (général)



a — *filtFlow* — *mandFlow* — *none* — *pmtnWbmt*
 — *filtWbmt* — *mandWbmt* — *pmtnFlow*

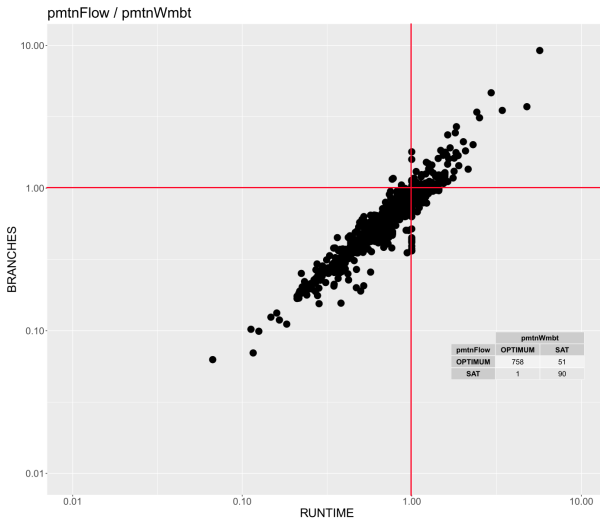


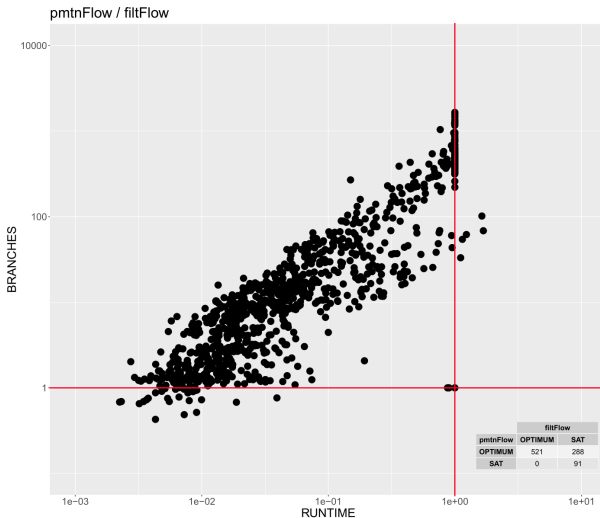
a — *filtFlow* — *mandFlow* — *none* — *pmtnWbmt*
 — *filtWbmt* — *mandWbmt* — *pmtnFlow*

Instances résolues à opt :

| <i>none</i> | <i>filtFlow</i> | <i>filtWbmt</i> | <i>mandFlow</i> | <i>mandWbmt</i> | <i>pmtnFlow</i> | <i>pmtnWbmt</i> |
|-------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 237 | 521 | 539 | 802 | 754 | 809 | 759 |

Résultats 2 (relaxations)





Conclusion

- 1^{ere} version d'une contrainte globale pour minimiser $\sum C_j$
- Schéma permettant de tester plusieurs relaxations
- Contrainte aide à la résolution
- *pmtnFlow* + rapide que *pmtnWmbt*, et a – de branches
- Actuellement, versions avec filtrage – rapide que celles avec uniquement la m-à-j de $\sum C_j$, mais ont – de branches

Conclusion

- 1^{ere} version d'une contrainte globale pour minimiser $\sum C_j$
- Schéma permettant de tester plusieurs relaxations
- Contrainte aide à la résolution
- *pmtnFlow* + rapide que *pmtnWmbt*, et a – de branches
- Actuellement, versions avec filtrage – rapide que celles avec uniquement la m-à-j de $\sum C_j$, mais ont – de branches

Conclusion

- 1^{ere} version d'une contrainte globale pour minimiser $\sum C_j$
- Schéma permettant de tester plusieurs relaxations
- Contrainte aide à la résolution
- *pmtnFlow* + rapide que *pmtnWmbt*, et a – de branches
- Actuellement, versions avec filtrage – rapide que celles avec uniquement la m-à-j de $\sum C_j$, mais ont – de branches

Conclusion

- 1^{ere} version d'une contrainte globale pour minimiser $\sum C_j$
- Schéma permettant de tester plusieurs relaxations
- Contrainte aide à la résolution
- *pmtnFlow* + rapide que *pmtnWmbt*, et a – de branches
- Actuellement, versions avec filtrage – rapide que celles avec uniquement la m-à-j de $\sum C_j$, mais ont – de branches

Conclusion

- 1^{ere} version d'une contrainte globale pour minimiser $\sum C_j$
- Schéma permettant de tester plusieurs relaxations
- Contrainte aide à la résolution
- *pmtnFlow* + rapide que *pmtnWmbt*, et a – de branches
- Actuellement, versions avec filtrage – rapide que celles avec uniquement la m-à-j de $\sum C_j$, mais ont – de branches

Conclusion

- 1^{ere} version d'une contrainte globale pour minimiser $\sum C_j$
- Schéma permettant de tester plusieurs relaxations
- Contrainte aide à la résolution
- *pmtnFlow* + rapide que *pmtnWmbt*, et a – de branches
- Actuellement, versions avec filtrage – rapide que celles avec uniquement la m-à-j de $\sum C_j$, mais ont – de branches

Perspectives

- Rendre la contrainte incrémentale (à la manière de ce qui est fait dans [Kovács et Beck, 2011])
- Intégrer d'autres relaxations ($1|sp-graph| \sum w_j C_j, \dots$)
- Intégrer la contraintes à des problèmes + variés (flowshop, ...)

Perspectives

- Rendre la contrainte incrémentale (à la manière de ce qui est fait dans [Kovács et Beck, 2011])
- Intégrer d'autres relaxations ($1|sp-graph| \sum w_j C_j, \dots$)
- Intégrer la contraintes à des problèmes + variés (flowshop, ...)

Perspectives

- Rendre la contrainte incrémentale (à la manière de ce qui est fait dans [Kovács et Beck, 2011])
- Intégrer d'autres relaxations ($1|sp-graph| \sum w_j C_j, \dots$)
- Intégrer la contraintes à des problèmes + variés (flowshop, ...)

Perspectives

- Rendre la contrainte incrémentale (à la manière de ce qui est fait dans [Kovács et Beck, 2011])
- Intégrer d'autres relaxations ($1|sp-graph| \sum w_j C_j, \dots$)
- Intégrer la contraintes à des problèmes + variés (flowshop, ...)