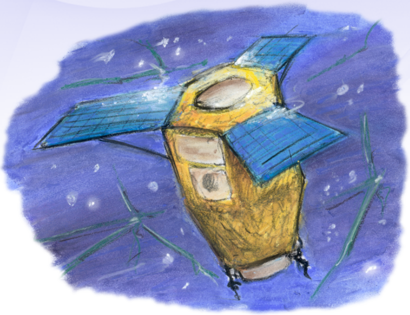
Allocation et partage équitables de ressources indivisibles : modélisation, complexité et algorithmique

Les scènes coupées au montage



Soutenance de thèse, Le 16 novembre 2007.

Sylvain Bouveret

devant le jury composé de :

Christian BESSIÈRE, Ulle ENDRISS, Thibault GAJDOS, Jean-Michel LACHIVER (co-directeur de thèse), Jérôme LANG (co-directeur de thèse), Michel LEMAÎTRE (co-directeur de thèse), Patrice PERNY, Thomas SCHIEX

Rapporteurs : **Boi FALTINGS, Patrice PERNY**

Les scènes coupées au montage

1. **Droits inégaux**
   - Détail des propriétés
   - CUF à droits inégaux
   - Propriétés des CUF introduites
2. **Problème EEF**
   - Détails du modèle logique
   - Quelques rappels de complexité
   - Détails des résultats de complexité
3. **Optimisation leximin**
   - La contrainte leximin (multiset ordering)
   - Fonctionnement de l'algorithme atleast
   - La méta-contrainte avec des contraintes linéaires
   - Référence réseau de comparaisons
   - Sous-ensembles saturés
4. **Résultats en vrac**

Les scènes coupées au montage

1. Droits inégaux
   - Détail des propriétés
   - CUF à droits inégaux
   - Propriétés des CUF introduites
2. Problème EEF
   - Détails du modèle logique
   - Quelques rappels de complexité
   - Détails des résultats de complexité
3. Optimisation leximin
   - La contrainte leximin (multiset ordering)
   - Fonctionnement de l'algorithme atleast
   - La méta-contrainte avec des contraintes linéaires
   - Référence réseau de comparaisons
   - Sous-ensembles saturés
4. Résultats en vrac

Droits inégaux et propriétés

Objectifs :

1. Proposer une extension des propriétés classiques pour permettre la prise en compte des droits inégaux :
   - unanimité, anonymat, IUA, ...
   - propriétés d'équité : absence d'envie, juste part, réduction des inégalités, indices d'inégalité, courbe de Lorenz, ...
2. Introduire de nouvelles propriétés pour caractériser l'effet des droits inégaux sur la prise de décision.

Droits inégaux et propriétés

Objectifs :

1. Proposer une extension des propriétés classiques pour permettre la prise en compte des droits inégaux :
   - unanimité, anonymat, IUA, ...
   - propriétés d'équité : absence d'envie, juste part, réduction des inégalités, indices d'inégalité, courbe de Lorenz, ...
2. Introduire de nouvelles propriétés pour caractériser l'effet des droits inégaux sur la prise de décision.

Unanimité

(Propriété inchangée)

Droits inégaux et propriétés

Objectifs :

1. Proposer une extension des propriétés classiques pour permettre la prise en compte des droits inégaux :
   - unanimité, anonymat, IUA, ...
   - propriétés d'équité : absence d'envie, juste part, réduction des inégalités, indices d'inégalité, courbe de Lorenz, ...
2. Introduire de nouvelles propriétés pour caractériser l'effet des droits inégaux sur la prise de décision.

Anonymat

Insensibilité à une permutation commune des composantes du profil d'utilité ET du vecteur de droits exogènes.

Droits inégaux et propriétés

Objectifs :

1. Proposer une extension des propriétés classiques pour permettre la prise en compte des droits inégaux :
   - unanimité, anonymat, IUA, ...
   - propriétés d'équité : absence d'envie, juste part, réduction des inégalités, indices d'inégalité, courbe de Lorenz, ...
2. Introduire de nouvelles propriétés pour caractériser l'effet des droits inégaux sur la prise de décision.

Équité généralisée

- Extension fondée sur le principe de duplication avec division des utilités (idée évoquée dans [Steinhaus, 1948]).
- $\rightsquigarrow$ Profil parfaitement égalitaire proportionnel au vecteur $\overrightarrow{e}$.
- Rapprochement avec la théorie de la décision en présence de risque.

Steinhaus, H. (1948).
The problem of fair division.
Econometrica, 16(1) :101–104.

Droits inégaux et propriétés

Objectifs :

1. Proposer une extension des propriétés classiques pour permettre la prise en compte des droits inégaux :
    - unanimité, anonymat, IUA, . . .
    - propriétés d'équité : absence d'envie, juste part, réduction des inégalités, indices d'inégalité, courbe de Lorenz, . . .
2. Introduire de nouvelles propriétés pour caractériser l'effet des droits inégaux sur la prise de décision.

Conformité

«Monotonie de l'ordre de bien-être social ou de la fonction d'utilité collective vis-à-vis d'une augmentation conjointe des droits et de l'utilité»

Droits inégaux et propriétés

Objectifs :

1. Proposer une extension des propriétés classiques pour permettre la prise en compte des droits inégaux :
   - unanimité, anonymat, IUA, ...
   - propriétés d'équité : absence d'envie, juste part, réduction des inégalités, indices d'inégalité, courbe de Lorenz, ...
2. Introduire de nouvelles propriétés pour caractériser l'effet des droits inégaux sur la prise de décision.

Avantage aux droits élevés

«Une augmentation d'utilité sur un agent à droit élevé est meilleure pour la collectivité».

Lien avec la propriété de conformité ?

Droits inégaux et propriétés

Objectifs :

1. Proposer une extension des propriétés classiques pour permettre la prise en compte des droits inégaux :
   - unanimité, anonymat, IUA, . . .
   - propriétés d'équité : absence d'envie, juste part, réduction des inégalités, indices d'inégalité, courbe de Lorenz, . . .
2. Introduire de nouvelles propriétés pour caractériser l'effet des droits inégaux sur la prise de décision.

Insensibilité à une Dilatation Commune des Droits

Équivalence des ordres de bien-être collectif $\succeq_{k.\overrightarrow{e}}$, pour tout k.

Fonctions d'utilités collectives à droits inégaux

	$u \div e \stackrel{def}{=} u/e$ (division)	$u \div e \stackrel{def}{=} u$ (réplication)
$g \stackrel{def}{=} \sum^{(m)}$ (utilitarisme cl.)	$\sum_{i \in \mathcal{N}} u_i$	$\sum_{i \in \mathcal{N}} (e_i \cdot u_i)$
$g \stackrel{def}{=} \min^{(m)}$ (égalitarisme)	$\min_{i \in \mathcal{N}} (u_i/e_i)$	$\min_{i \in \mathcal{N}} u_i$

Proposition d'extension des familles de fonctions d'utilité : somme des puissances et moyennes pondérées ordonnées.

Propriétés vérifiées par les CUF classiques

	una.	ano.	IUA	Juste P.	Réd. In.	conf.	ADE	IDCD
$\sum_i e_i \cdot u_i$	oui	oui	oui	**non**	**non**	oui	oui	oui
$\sum_i u_i$	oui	oui	oui	**non**	**non**	oui	**non**	oui
$\text{leximin}_i\, u_i/e_i$	oui	oui	oui	oui	oui	oui	oui	oui
$\text{leximin}_i\, u_i$	oui	oui	oui	**non**	**non**	oui	**non**	oui
$g^{(p)}_{\overrightarrow{e},div}$	oui	oui	?	$p < 1$	oui	oui	oui	oui
$g^{(p)}_{\overrightarrow{e},rep}$	oui	oui	?	**non**	oui	oui	oui	oui

Les scènes coupées au montage

1. Droits inégaux
   - Détail des propriétés
   - CUF à droits inégaux
   - Propriétés des CUF introduites
2. Problème EEF
   - Détails du modèle logique
   - Quelques rappels de complexité
   - Détails des résultats de complexité
3. Optimisation leximin
   - La contrainte leximin (multiset ordering)
   - Fonctionnement de l'algorithme atleast
   - La méta-contrainte avec des contraintes linéaires
   - Référence réseau de comparaisons
   - Sous-ensembles saturés
4. Résultats en vrac

La contrainte de préemption

Lorsque toutes les préférences sont dichotomiques, une instance du problème de partage peut être représenté par l'ensemble des formules des agents :

$$\mathcal{P} = \langle \varphi_1, \ldots, \varphi_n \rangle$$

$\Rightarrow$ Un partage valide est donc une interprétation des o_i, satisfaisant la contrainte de préemption (exprimée comme une formule de $L_{\mathscr{O}}^{alloc}$) :

$$\Gamma_{\mathcal{P}} = \bigwedge_{o \in \mathscr{O}} \bigwedge_{i \neq j} \neg(al(o,i) \wedge al(o,j))$$

Exemple (suite) :

$\Gamma_{\mathcal{P}} = \neg(al(o_1,1) \wedge al(o_1,2)) \wedge \neg(al(o_2,1) \wedge al(o_2,2)) \wedge \neg(al(o_3,1) \wedge al(o_3,2))$.

La contrainte de préemption

Lorsque toutes les préférences sont dichotomiques, une instance du problème de partage peut être représenté par l'ensemble des formules des agents :

$$\mathcal{P} = \langle \varphi_1, \ldots, \varphi_n \rangle$$

⇒ Un partage valide est donc une interprétation des o_i, satisfaisant la contrainte de préemption (exprimée comme une formule de $L_{\mathscr{O}}^{alloc}$) :

$$\Gamma_{\mathcal{P}} = \bigwedge_{o \in \mathscr{O}} \bigwedge_{i \neq j} \neg(al(o,i) \wedge al(o,j))$$

Exemple (suite) :

$$\Gamma_{\mathcal{P}} = \neg(al(o_1,1) \wedge al(o_1,2)) \wedge \neg(al(o_2,1) \wedge al(o_2,2)) \wedge \neg(al(o_3,1) \wedge al(o_3,2)).$$

Préférences dichotomiques et absence d'envie

L'absence d'envie s'exprime simplement comme une formule de $L_{\mathscr{O}}^{alloc}$.
Nous introduisons deux nouvelles notations :

- $\varphi_i^* = \varphi_i(o \leftarrow al(o, i))$,
- $\varphi_{j|i}^* = \varphi_i^*(al(o, i) \leftarrow al(o, j))$

Exemple (suite) :

		
$Good_i$	$\{\{o_1, o_2\}, \{o_2, o_3\}\}$	$\{\{o_2\}\{o_2, o_3\}\}$
φ_i	$(o_1 \land o_2 \land \neg o_3) \lor (\neg o_1 \land o_2 \land o_3)$	$o_2 \land \neg o_1$
φ_i^*	$(al(o_1, 1) \land al(o_2, 1) \land \neg al(o_2, 1)) \lor$ $(\neg al(o_1, 1) \land al(o_2, 1) \land al(o_3, 1))$	$al(o_2, 2) \land \neg al(o_1, 2)$
$\varphi_{j\|i}^*$	$(al(o_1, 2) \land al(o_2, 2) \land \neg al(o_2, 2)) \lor$ $(\neg al(o_1, 2) \land al(o_2, 2) \land al(o_3, 2))$	$al(o_2, 1) \land \neg al(o_1, 1)$

Préférences dichotomiques et absence d'envie

L'absence d'envie s'exprime simplement de la manière suivante :

$$\Lambda_{\mathcal{P}} = \bigwedge_{i=1,\ldots,N} \left[\varphi_i^* \vee \left(\bigwedge_{j \neq i} \neg \varphi_{j|i}^* \right) \right]$$

Proposition

$\overrightarrow{\pi}$ est sans envie si et seulement si $F(\overrightarrow{\pi}) \models \Lambda_{\mathcal{P}}$.

Préférences dichotomiques et absence d'envie

Exemple (suite) :

$$\Lambda_{\mathcal{P}} = \begin{array}{c} \overbrace{(((al(o_1,1) \wedge al(o_2,1) \wedge \neg al(o_3,1)) \vee (\neg al(o_1,1) \wedge al(o_2,1) \wedge al(o_3,1)))}^{\text{est satisfait avec sa part}} \\ \vee \overbrace{\neg((al(o_1,2) \wedge al(o_2,2) \wedge \neg al(o_3,2)) \vee (\neg al(o_1,2) \wedge al(o_2,2) \wedge al(o_3,2))))}^{\text{ne serait pas satisfait avec la part de}} \\ \bigwedge \\ (\underbrace{(al(o_2,2) \wedge \neg al(o_1,2))}_{\text{est satisfait avec sa part}} \vee \underbrace{\neg(al(o_2,1) \wedge \neg al(o_1,1))}_{\text{ne serait pas satisfait avec la part de}}) \end{array}$$

Préférences dichotomiques et Pareto-efficacité

La Pareto-efficacité requiert que l'allocation satisfasse un nombre **maximal** (pour l'inclusion) d'agents (tout en étant valide).

Proposition

$\overrightarrow{\pi}$ est une allocation valide et Pareto-efficace si et seulement si $\{\varphi_i^* \mid F(\overrightarrow{\pi}) \models \varphi_i^*\}$ est un sous-ensemble maximal $\Gamma_{\mathcal{P}}$-consistant de $\{\varphi_1^*, \ldots, \varphi_n^*\}$.

Préférences dichotomiques et Pareto-efficacité

Exemple (suite) :

		
$Good_i$	$\{\{o_1, o_2\}, \{o_2, o_3\}\}$	$\{\{o_2\}\{o_2, o_3\}\}$
φ_i	$(o_1 \wedge o_2 \wedge \neg o_3) \vee (\neg o_1 \wedge o_2 \wedge o_3)$	$o_2 \wedge \neg o_1$
φ_i^*	$(al(o_1,1) \wedge al(o_2,1) \wedge \neg al(o_2,1)) \vee$ $(\neg al(o_1,1) \wedge al(o_2,1) \wedge al(o_3,1))$	$al(o_2,2) \wedge \neg al(o_1,2)$

$$\Gamma_{\mathcal{P}} = \neg(al(o_1,1) \wedge al(o_1,2)) \wedge \neg(al(o_2,1) \wedge al(o_2,2)) \wedge \neg(al(o_3,1) \wedge al(o_3,2)).$$

Les deux sous-ensembles maximaux $\Gamma_{\mathcal{P}}$-consistants de $\{\varphi_1^*, \varphi_2^*\}$ sont $\{\varphi_1^*\}$ et $\{\varphi_2^*\}$ (à cause de o_2).

Un problème d'inférence sceptique

Il s'agit d'un problème connu dans le domaine du raisonnement non monotone : *inférence sceptique avec des défauts normaux sans prérequis* [Reiter, 1980].

Conséquence sceptique

Soient Δ un ensemble de formules, β (un *fait*) et ψ (la *conséquence*) des formules. ψ est une **conséquence sceptique** de $\langle \beta, \Delta \rangle$ (noté $\langle \beta, \Delta \rangle \mid\!\sim^{\forall} \psi$) si et seulement si $\forall \mathscr{S} \in \mathit{MaxCons}(\Delta, \beta)$ (extension) $\bigwedge \mathscr{S} \wedge \beta \models \psi$.

Donc on peut réduire le problème d'existence d'un partage Pareto-efficace et sans envie à :

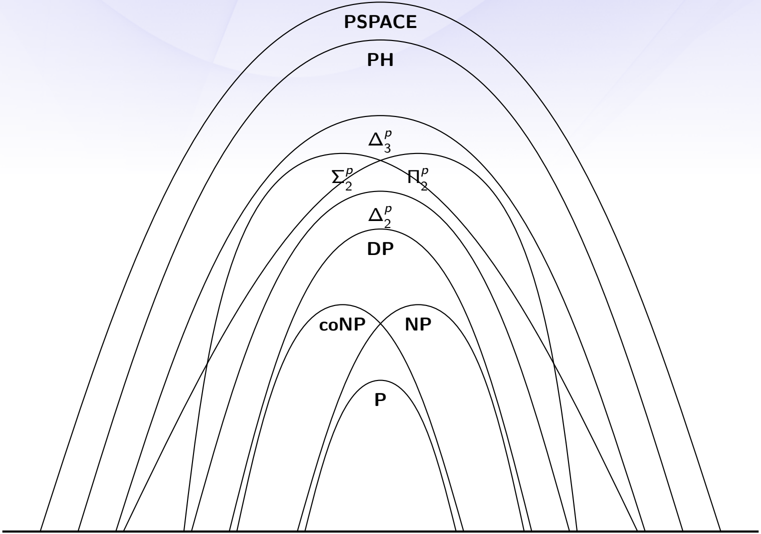
$$\langle \Gamma_{\mathcal{P}}, \{\varphi_1^*, \ldots, \varphi_n^*\} \rangle \not\mid\!\sim^{\forall} \neg \Lambda_{\mathcal{P}}$$

Reiter, R. (1980).
A logic for default reasoning.
Artificial Intelligence, 13 :81–132.

Des classes de complexité bien utiles

- **P**, **NP** : *a priori* vous les connaissez déjà...
- $\mathbf{BH}_2 = \mathbf{DP} = \{L_1 \cap L_2 \mid L_1 \in \mathbf{NP} \text{ and } L_2 \in \mathbf{coNP}\}$
- $\Delta_2^p = \mathbf{P}^{\mathbf{NP}}$ (langages décidables en temps polynomial par une machine de Turing déterministe à oracles [SAT]).
- $\Sigma_2^p = \mathbf{NP}^{\mathbf{NP}}$
- $\Theta_2^p = \Delta_2^p[\mathcal{O}(\log n)]$ (seulement un nombre logarithmique d'oracles).

Le schéma de complexité



Restrictions et autres critères d'efficacité

- **Restrictions :**
  - préférences identiques,
  - nombre d'agents,
  - le langage propositionnel.
- **Critère d'efficacité alternatif :**
  - complétude,
  - nombre maximal d'agents satisfaits.

Restrictions et autres critères d'efficacité

- **Restrictions :**
  - préférences identiques,
  - nombre d'agents,
  - le langage propositionnel.
- **Critère d'efficacité alternatif :**
  - complétude,
  - nombre maximal d'agents satisfaits.

Préférences identiques

- Préférences dichotomiques identiques et **monotones** $\leadsto$ **NP**-complet (même pour un n fixé).
- Préférences dichotomiques identiques et **monotones** $\leadsto$ **coBH**$_2$-complet (même pour un n fixé).

Restrictions et autres critères d'efficacité

- **Restrictions :**
  - préférences identiques,
  - nombre d'agents,
  - le langage propositionnel.
- **Critère d'efficacité alternatif :**
  - complétude,
  - nombre maximal d'agents satisfaits.

Deux agents

- Préférences dichotomiques **monotones**, 2 agents $\rightsquigarrow$ **NP**-complet.
- Préférences dichotomiques **monotones**, 2 agents $\rightsquigarrow$ **coBH**$_2$-complet.

Restrictions et autres critères d'efficacité

- **Restrictions :**
  - préférences identiques,
  - nombre d'agents,
  - le langage propositionnel.
- **Critère d'efficacité alternatif :**
  - complétude,
  - nombre maximal d'agents satisfaits.

Restrictions sur le langage propositionnel

- Clauses $\rightsquigarrow$ Polynomial.
- Cubes $\rightsquigarrow$ **NP**-complet.
- Classe $\mathcal{C}$ de formules telle que :
  - $[SAT[\mathcal{C}]]$ est polynomial, et
  - $\mathcal{C}$ est clos par conjonction.

(*e.g.* 2-CNF) $\rightsquigarrow$ **NP**-complet.

Restrictions et autres critères d'efficacité

- **Restrictions :**
  - préférences identiques,
  - nombre d'agents,
  - le langage propositionnel.
- **Critère d'efficacité alternatif :**
  - complétude,
  - nombre maximal d'agents satisfaits.

Complétude

- Allocation sans envie complète $\rightsquigarrow$ **NP**-complet (même pour 2 agents avec préférences identiques).
- Est-ce toujours vrai si les préférences sont **monotones** ?

Restrictions et autres critères d'efficacité

- **Restrictions :**
  - préférences identiques,
  - nombre d'agents,
  - le langage propositionnel.
- **Critère d'efficacité alternatif :**
  - complétude,
  - nombre maximal d'agents satisfaits.

Nombre maximal d'agents

- Existence d'une allocation sans envie parmi celles qui satisfont un nombre maximal d'agents $\rightsquigarrow$ Θ_2^p-complet (même pour des préférences monotones).

Préférences sous forme logique pondérée

D'après le résultat précédent, le problème [EEF EXISTENCE] est aussi Σ_2^p-complet pour des préférences sous forme logique pondérée.
Cependant, l'introduction d'utilités permet des critères d'efficacité alternatifs, fondés sur la maximisation de l'utilité collective **utilitariste classique** et **égalitariste**.

Dans les deux cas, il s'avère que le problème [EEF EXISTENCE] est Δ_2^p-complet.

Préférences sous forme logique pondérée

D'après le résultat précédent, le problème [EEF EXISTENCE] est aussi Σ_2^p-complet pour des préférences sous forme logique pondérée. Cependant, l'introduction d'utilités permet des critères d'efficacité alternatifs, fondés sur la maximisation de l'utilité collective **utilitariste classique** et **égalitariste**.

Dans les deux cas, il s'avère que le problème [EEF EXISTENCE] est Δ_2^p-complet.

Préférences sous forme logique pondérée

D'après le résultat précédent, le problème [EEF EXISTENCE] est aussi Σ_2^p-complet pour des préférences sous forme logique pondérée. Cependant, l'introduction d'utilités permet des critères d'efficacité alternatifs, fondés sur la maximisation de l'utilité collective **utilitariste classique** et **égalitariste**.

Dans les deux cas, il s'avère que le problème [EEF EXISTENCE] est Δ_2^p-complet.

Préférences additives (demandes atomiques)

On sait déjà (d'après [Lipton et al., 2004]) que :

Proposition

Le problème d'existence d'une allocation **complète** et sans envie pour des agents ayant des préférences additives est **NP**-complet.

Mais… Quelle est la complexité exacte du problème d'existence d'une allocation **Pareto-efficace** et sans envie pour des agents ayant des préférences additives ?

Tout ce que nous pouvons dire est que ce problème est dans Σ_2^p et **NP**-difficile…

Lipton, R., Markakis, E., Mossel, E., and Saberi, A. (2004).
On approximately fair allocations of divisible goods.
In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC-04)*, New York, NY. ACM.

Préférences additives (demandes atomiques)

On sait déjà (d'après [Lipton et al., 2004]) que :

Proposition

Le problème d'existence d'une allocation **complète** et sans envie pour des agents ayant des préférences additives est **NP**-complet.

Mais... Quelle est la complexité exacte du problème d'existence d'une allocation **Pareto-efficace** et sans envie pour des agents ayant des préférences additives ?

Tout ce que nous pouvons dire est que ce problème est dans Σ_2^p et **NP**-difficile...

Lipton, R., Markakis, E., Mossel, E., and Saberi, A. (2004).
On approximately fair allocations of divisible goods.
In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC-04)*, New York, NY. ACM.

Préférences additives (demandes atomiques)

On sait déjà (d'après [Lipton et al., 2004]) que :

Proposition

Le problème d'existence d'une allocation **complète** et sans envie pour des agents ayant des préférences additives est **NP**-complet.

Mais... Quelle est la complexité exacte du problème d'existence d'une allocation **Pareto-efficace** et sans envie pour des agents ayant des préférences additives ?
Tout ce que nous pouvons dire est que ce problème est dans Σ_2^p et **NP**-difficile...

Lipton, R., Markakis, E., Mossel, E., and Saberi, A. (2004).
On approximately fair allocations of divisible goods.
In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC-04)*, New York, NY. ACM.

Préférences additives (demandes atomiques)

Restrictions du problème général avec des préférences additives :

- Préférences identiques $\rightsquigarrow$ **NP**-complet (même pour un n fixé).
- Préférences 0–1 $\rightsquigarrow$ **NP**-complet.
- Préférences 0–1–…–k (pour k fixé) $\rightarrow$? ? ?
- Nombre d'agents strictement supérieur au nombre d'objets $\rightsquigarrow$
  - pas de partage Pareto-efficace et sans envie si les préférences sont monotones,
  - aussi difficile que le problème général si aucune hypothèse n'est faite sur la monotonie ;
- Nombre d'agents égal au nombre d'objets $\rightsquigarrow$ polynomial.

Préférences additives (demandes atomiques)

Restrictions du problème général avec des préférences additives :

- Préférences identiques $\rightsquigarrow$ **NP**-complet (même pour un n fixé).
- Préférences 0–1 $\rightsquigarrow$ **NP**-complet.
- Préférences 0–1–…–k (pour k fixé) $\rightarrow$? ? ?
- Nombre d'agents strictement supérieur au nombre d'objets $\rightsquigarrow$
  - pas de partage Pareto-efficace et sans envie si les préférences sont monotones,
  - aussi difficile que le problème général si aucune hypothèse n'est faite sur la monotonie ;
- Nombre d'agents égal au nombre d'objets $\rightsquigarrow$ polynomial.

Préférences additives (demandes atomiques)

Restrictions du problème général avec des préférences additives :

- Préférences identiques $\rightsquigarrow$ **NP**-complet (même pour un n fixé).
- Préférences 0–1 $\rightsquigarrow$ **NP**-complet.
- Préférences 0–1–…–k (pour k fixé) $\rightarrow$? ? ?
- Nombre d'agents strictement supérieur au nombre d'objets $\rightsquigarrow$
  - pas de partage Pareto-efficace et sans envie si les préférences sont monotones,
  - aussi difficile que le problème général si aucune hypothèse n'est faite sur la monotonie ;
- Nombre d'agents égal au nombre d'objets $\rightsquigarrow$ polynomial.

Préférences additives (demandes atomiques)

Restrictions du problème général avec des préférences additives :

- Préférences identiques $\rightsquigarrow$ **NP**-complet (même pour un n fixé).
- Préférences 0–1 $\rightsquigarrow$ **NP**-complet.
- Préférences 0–1–...–k (pour k fixé) $\rightarrow$???
- Nombre d'agents strictement supérieur au nombre d'objets $\rightsquigarrow$
  - pas de partage Pareto-efficace et sans envie si les préférences sont monotones,
  - aussi difficile que le problème général si aucune hypothèse n'est faite sur la monotonie ;
- Nombre d'agents égal au nombre d'objets $\rightsquigarrow$ polynomial.

Préférences additives (demandes atomiques)

Restrictions du problème général avec des préférences additives :

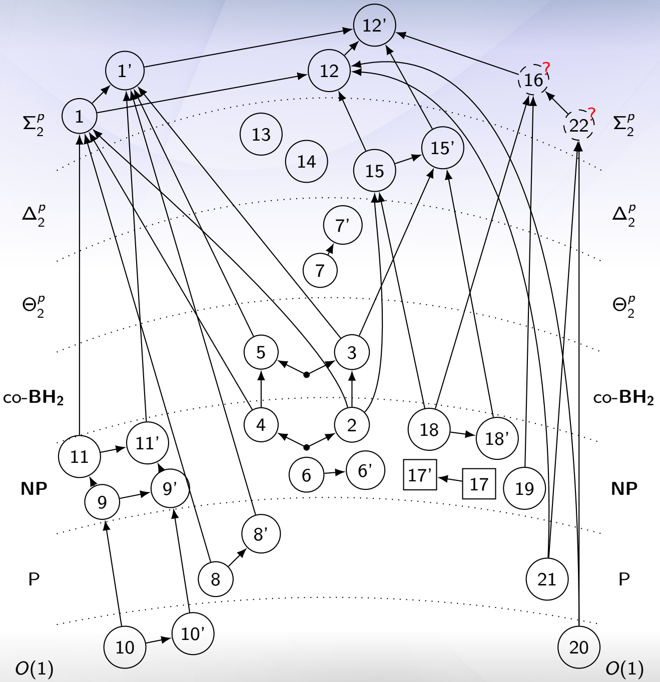
- Préférences identiques $\leadsto$ **NP**-complet (même pour un n fixé).
- Préférences 0–1 $\leadsto$ **NP**-complet.
- Préférences 0–1–…–k (pour k fixé) $\rightarrow$? ? ?
- Nombre d'agents strictement supérieur au nombre d'objets $\leadsto$
  - pas de partage Pareto-efficace et sans envie si les préférences sont monotones,
  - aussi difficile que le problème général si aucune hypothèse n'est faite sur la monotonie ;
- Nombre d'agents égal au nombre d'objets $\leadsto$ polynomial.

Résultats de complexité (préférences dichotomiques)

	Efficacité	nb d'agents	préférences	monotonie
1, 1'	Pareto-efficacité	non fixé	—	oui (1) ou non (1')
2	Pareto-efficacité	non fixé ou fixé avec $n \geq 2$	identiques	oui
3	Pareto-efficacité	non fixé ou fixé avec $n \geq 2$	identiques	non
4	Pareto-efficacité	2 agents	—	oui
5	Pareto-efficacité	2 agents	—	non
6, 6'	allocation complète	non fixé ou fixé avec $n \geq 2$	identiques ou non	oui (6) ou non (6')
7, 7'	nb max d'agents	non fixé	—	oui (7) ou non (7')
8, 8'	Pareto-efficacité	—	clauses	oui (8) ou non (8')
9, 9'	Pareto-efficacité	—	cubes	oui (9) ou non (9')
10, 10'	Pareto-efficacité	—	cubes avec cond. spéc.	oui (10) ou non (10')
11, 11'	Pareto-efficacité	—	$\mathcal{C}$ tq $SAT[\mathcal{C}] \in \mathbf{P}$ et clos par $\wedge$	oui (11) ou non (11')

Résultats de complexité (préférences non dichotomiques

	Efficacité	nb d'agents	préférences	monotonie
12, 12'	Pareto-efficacité	non fixé	numériques (logiques)	oui (12) ou non (12')
13	CUF util.	non fixé ou fixé avec $n \geq 2$	numériques (logiques)	non
14	CUF egal.	non fixé ou fixé avec $n \geq 2$	numériques (logiques)	non
15, 15'	Pareto-efficacité	non fixé ou fixé avec $n \geq 2$	numériques (logiques), identique	oui (15) ou non (15')
16	Pareto-efficacité	non fixé	additives	non
17, 17'	allocation complète	non fixé	additives	oui (17) ou non (17')
18, 18'	Pareto-efficacité	non fixé ou fixé avec $n \geq 2$	additives identiques	oui (18) ou non (18')
19	Pareto-efficacité	—	additives 0–1	oui
20	Pareto-efficacité	$>$ Nb d'objets	additives	oui
21	Pareto-efficacité	$=$ Nb d'objets	additives	oui
22	Pareto-efficacité	$\geq$ Nb d'objets	additives	non



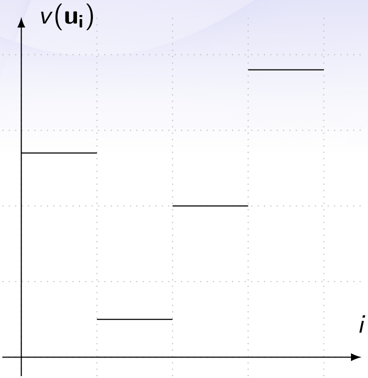
12'
1'
12
16?
1
22?
Σ₂ᵖ
13
14
15
15'
Δ₂ᵖ
7'
7
Θ₂ᵖ
5
3
co-BH₂
4
2
18
18'
11
11'
9
9'
6
6'
17'
17
19
NP
8'
8
21
P
10
10'
20
O(1)

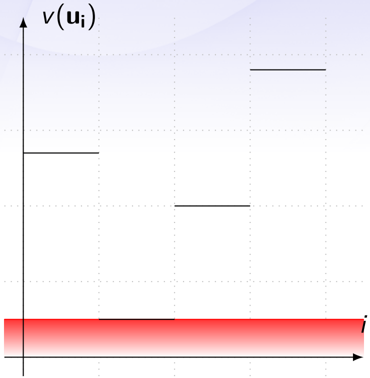
Les scènes coupées au montage

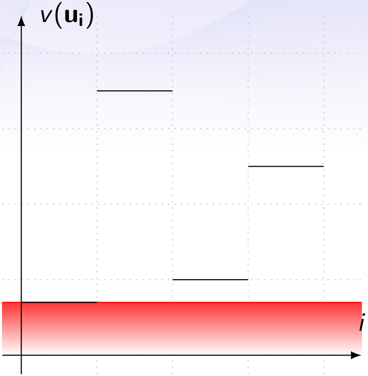
Une contrainte Leximin

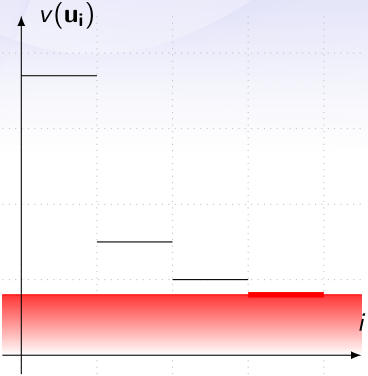
- Nous utilisons une contrainte **Leximin** : **Leximin**$(\overrightarrow{\lambda}, \overrightarrow{\mathbf{x}})$ (le vecteur $\overrightarrow{\mathbf{x}}$ doit être leximin-supérieur au vecteur d'entiers $\overrightarrow{\lambda}$).
- Cette contrainte est fondée sur la contrainte **Multiset Ordering**, introduite dans [Frisch et al., 2003] (filtrage en $O(n \log(n))$).

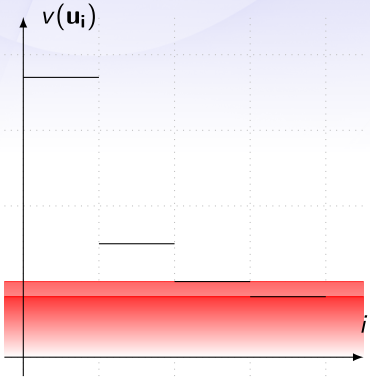
Frisch, A. M., Hnich, B., Kiziltan, Z., Miguel, I., and Walsh, T. (2003).
Multiset ordering constraints.
In Gottlob, G. and Walsh, T., editors, *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, Acapulco, Mexico. Morgan Kaufmann.

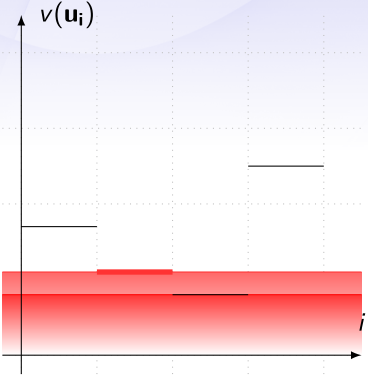

$v(\mathbf{u_i})$
i

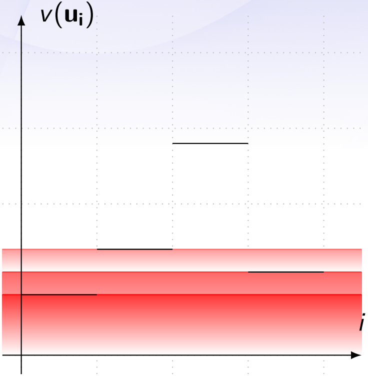

$v(\mathbf{u_i})$
i

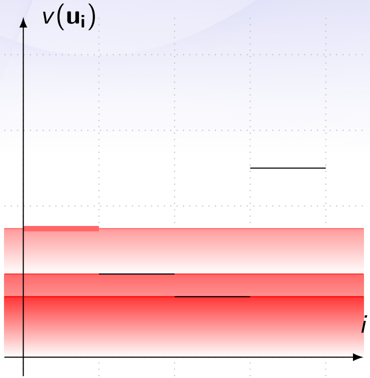

$v(\mathbf{u_i})$
i

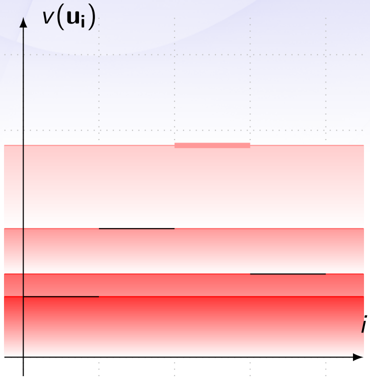

$v(\mathbf{u_i})$
i


$v(\mathbf{u_i})$
i


$v(\mathbf{u_i})$
i


$v(\mathbf{u_i})$
i


$v(\mathbf{u_i})$
i


$v(\mathbf{u_i})$
i

Implementation

Implementation of the algorithms :

The algorithms have been implemented using Choco[Laburthe, 2000] in Java, and the second one (**AtLeast**) has also been implemented using CPLEX.

Remark

Our specific constraint **AtLeast** can be encoded with a set of linear constraints :
$\mathbf{AtLeast}(\{\mathbf{x_1} \geq \mathbf{y}, \ldots, \mathbf{x_n} \geq \mathbf{y}\}, k) \Leftrightarrow$
$\{\mathbf{x_1} + \delta_1 \overline{y} \geq \mathbf{y}, \ldots, \mathbf{x_n} + \delta_n \overline{y} \geq \mathbf{y}, \sum_{i=1}^{n} \delta_\mathbf{i} \leq n - k\}$, with $\{\delta_1, \ldots, \delta_\mathbf{n}\}$ 0–1 variables.

Laburthe, F. (2000).
Choco : Implementing a cp kernel.
In *Proceedinfs of TRICS'2000, Workshop on techniques for implementing CP systems*, Singapore.
http://sourceforge.net/projects/choco.

Implementation

Implementation of the algorithms :

The algorithms have been implemented using Choco[Laburthe, 2000] in Java, and the second one (**AtLeast**) has also been implemented using CPLEX.

Remark

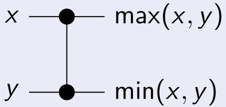
Our specific constraint **AtLeast** can be encoded with a set of linear constraints :
$\mathbf{AtLeast}(\{\mathbf{x_1} \geq \mathbf{y}, \ldots, \mathbf{x_n} \geq \mathbf{y}\}, k) \Leftrightarrow$
$\{\mathbf{x_1} + \delta_1 \overline{y} \geq \mathbf{y}, \ldots, \mathbf{x_n} + \delta_n \overline{y} \geq \mathbf{y}, \sum_{i=1}^{n} \delta_i \leq n - k\}$, with $\{\delta_1, \ldots, \delta_n\}$ 0–1 variables.

Laburthe, F. (2000).
Choco : Implementing a cp kernel.
In *Proceedinfs of TRICS'2000, Workshop on techniques for implementing CP systems*, Singapore.
http://sourceforge.net/projects/choco.

Trier par réseau de comparaisons

De manière intéressante, cet algorithme est lié à la notion de tri par **réseau de comparaisons** [Cormen et al., 2001, page 704].

Comparateur



L'idée est d'implanter des algorithmes de tri par seule utilisation de comparateurs.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001).
Introduction to Algorithms, Second Edition.
MIT Press.

Principle of the algorithm

This algorithm comes from the litterature on fuzzy CSPs [Dubois and Fortemps, 1999].

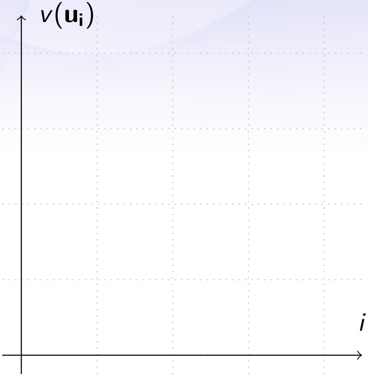
Saturated subset

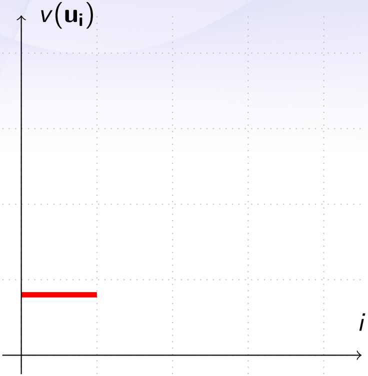
Let $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ be a constraint networtk, $\overrightarrow{u}$ be a set of objective variables, and α be the maximal admissible value of $\min(\overrightarrow{u})$.
A subset $\mathcal{S}$ of variables from $\overrightarrow{u}$ is said **saturated** if there is a solution with all $u_i \in \mathcal{S}$ instantiated to α and all $u_i \notin \mathcal{S}$ instantiated to a strictly greater value than α.

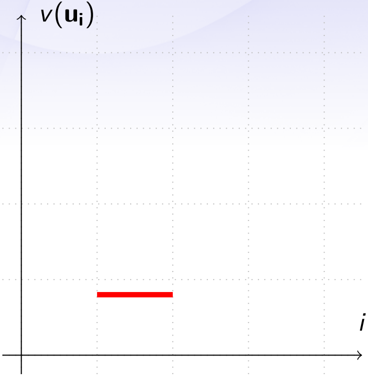
Dubois, D. and Fortemps, P. (1999).
Computing improved optimal solutions to max-min flexible constraint satisfaction problems.
European Journal of Operational Research.

Principle of the algorithm

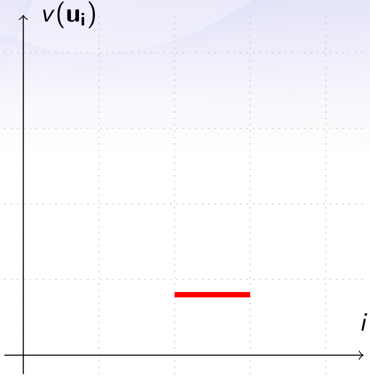
- At each step of the algorithm, we maximize the current component y_i of the leximin vector (as in the previous algorithms).
- For each cardinality-minimal saturated subset $\mathcal{S}$:
  - all the $u_i \in \mathcal{S}$ are fixed to $\widehat{y_i}$ and are removed from $\overrightarrow{u}$,
  - we restart the procedure on the new constraint network until no variable remains in $\overrightarrow{u}$.
- At the end, we compare all the potential leximin-optimal vectors found.

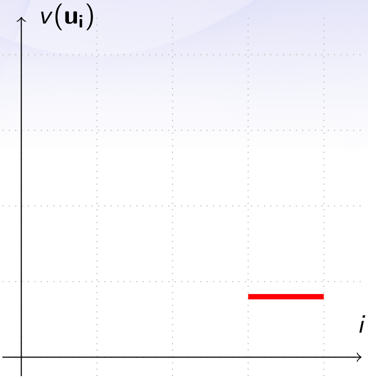

$v(\mathbf{u_i})$
i

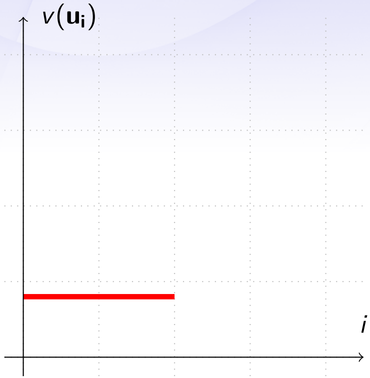

$v(\mathbf{u_i})$
i

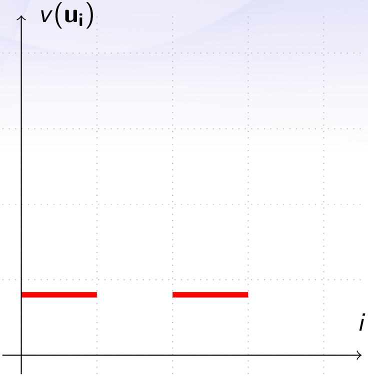


$v(\mathbf{u_i})$
i


$v(\mathbf{u_i})$
i


$v(\mathbf{u_i})$
i

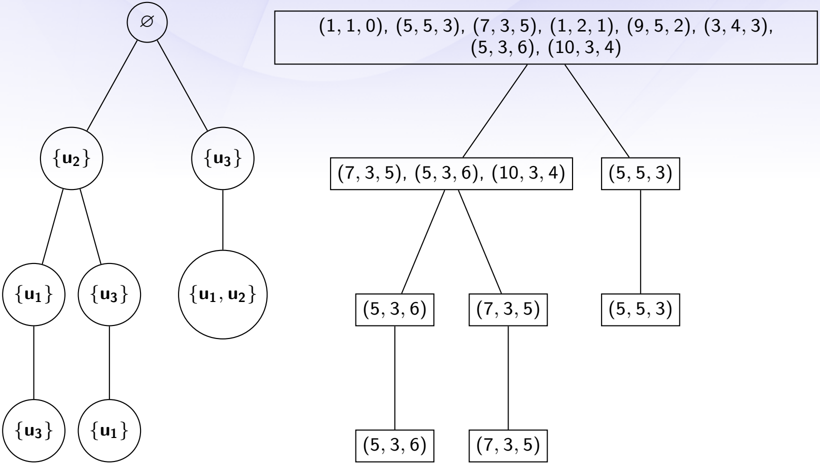

$v(\mathbf{u_i})$
i


$v(\mathbf{u_i})$
i

Example

Example

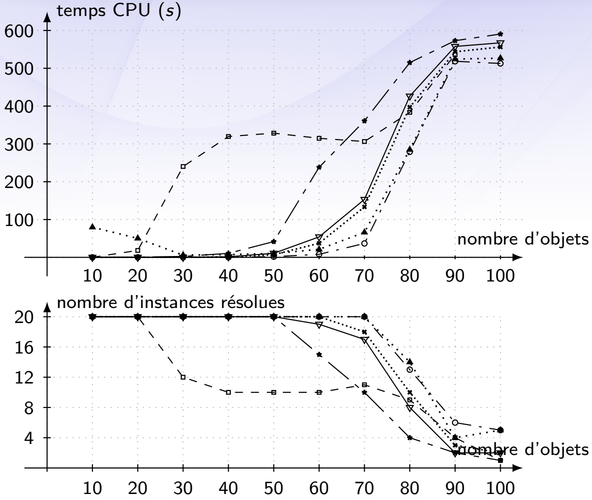
Let $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ be a constraint network and let $(\mathbf{u_1}, \mathbf{u_2}, \mathbf{u_3}) \in \mathcal{X}^3$ be an objective vector. We suppose that the set of solutions of the constraint network leads to the following set of possible values for the objective vector : $(1, 1, 0)$, $(5, 5, 3)$, $(7, 3, 5)$, $(1, 2, 1)$, $(9, 5, 2)$, $(3, 4, 3)$, $(5, 3, 6)$ and $(10, 3, 4)$.


∅
{u2}
{u3}
{u1}
{u3}
{u1, u2}
{u3}
{u1}
(1, 1, 0), (5, 5, 3), (7, 3, 5), (1, 2, 1), (9, 5, 2), (3, 4, 3), (5, 3, 6), (10, 3, 4)
(7, 3, 5), (5, 3, 6), (10, 3, 4)
(5, 5, 3)
(5, 3, 6)
(7, 3, 5)
(5, 5, 3)
(5, 3, 6)
(7, 3, 5)

Les scènes coupées au montage

1. Droits inégaux
   - Détail des propriétés
   - CUF à droits inégaux
   - Propriétés des CUF introduites
2. Problème EEF
   - Détails du modèle logique
   - Quelques rappels de complexité
   - Détails des résultats de complexité
3. Optimisation leximin
   - La contrainte leximin (multiset ordering)
   - Fonctionnement de l'algorithme atleast
   - La méta-contrainte avec des contraintes linéaires
   - Référence réseau de comparaisons
   - Sous-ensembles saturés
4. **Résultats en vrac**

Problème Pléiades linéaire (10 agents)



Algorithme 1 fondé sur la contrainte **Leximin**.

Algorithme 2 fondé sur la contrainte **Sort**.

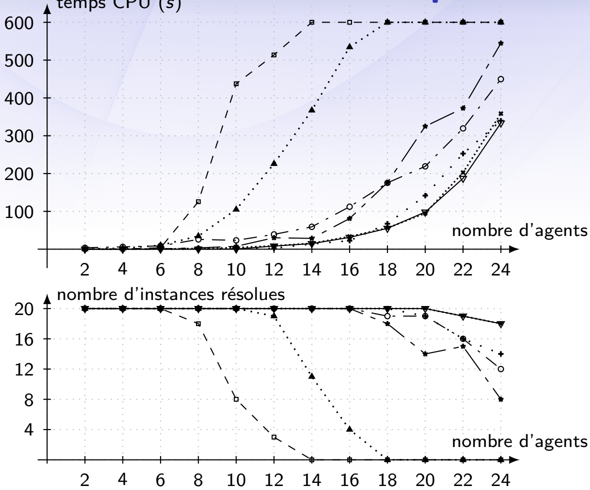
Algorithme 3 fondé sur la contrainte **AtLeast**.

Algorithme 4 fondé sur les transformations max-min.

Algorithme 5, inspiré de [Dubois and Fortemps, 1999].

Algorithme fondé sur le calcul et la comparaison de toutes les solutions du problème.

Enchères combinatoires équitables



Algorithme 1 fondé sur la contrainte **Leximin**.

Algorithme 2 fondé sur la contrainte **Sort**.

Algorithme 3 fondé sur la contrainte **AtLeast**.

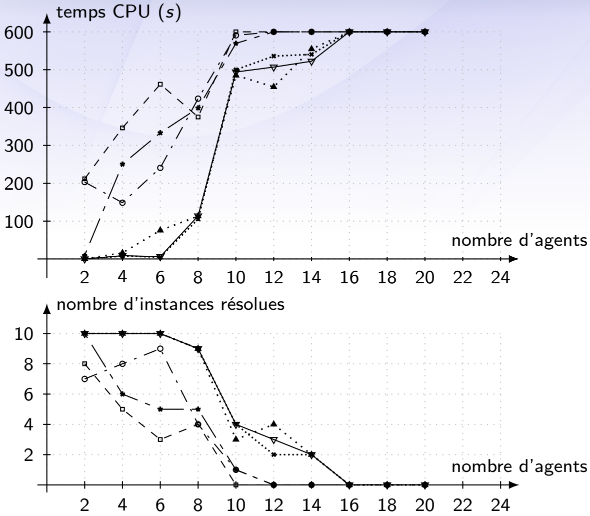
Algorithme 4 fondé sur les transformations max-min.

Algorithme 5, inspiré de [Dubois and Fortemps, 1999].

Algorithme fondé sur le calcul et la comparaison de toutes les solutions du problème.

Calcul de la solution optimale au sens de la somme (correspond au *Winner Determination Problem*).

RATS



Algorithme 1 fondé sur la contrainte **Leximin**.

Algorithme 2 fondé sur la contrainte **Sort**.

Algorithme 3 fondé sur la contrainte **AtLeast**.

Algorithme 4 fondé sur les transformations max-min.

Algorithme 5, inspiré de [Dubois and Fortemps, 1999].

Algorithme fondé sur le calcul et la comparaison de toutes les solutions du problème.