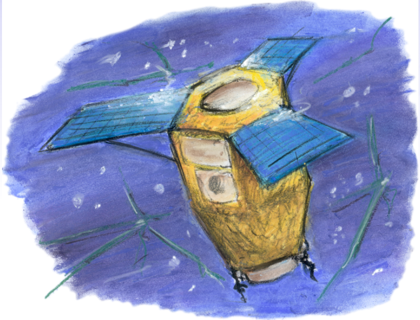
Allocation et partage équitables de ressources indivisibles : modélisation, complexité et algorithmique



Soutenance de thèse, Le 16 novembre 2007.

Sylvain Bouveret

devant le jury composé de :

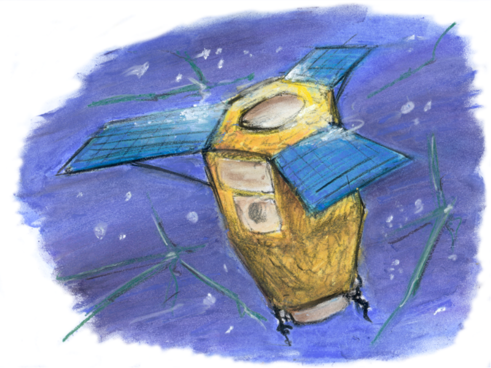
Christian BESSIÈRE, Ulle ENDRISS, Thibault GAJDOS, Jean-Michel LACHIVER (co-directeur de thèse), Jérôme LANG (co-directeur de thèse), Michel LEMAÎTRE (co-directeur de thèse), Patrice PERNY, Thomas SCHIEX

Rapporteurs : **Boi FALTINGS, Patrice PERNY**

La constellation Pléiades

Application CNES

Une constellation de satellites agiles d'observation de la Terre en lumière visible haute définition.



La constellation Pléiades

- Constellation cofinancée de manière inégale par plusieurs pays.
- Requêtes d'observation...
  - envoyées par chaque agent (agences civiles et militaires de chaque pays) au centre de programmation et de planification,
  - simples (photographie simple acquise en un seul passage) ou complexes (photographies *stereo*, zones plus larges que le champs de l'instrument optique,...),
  - d'importances inégales pour les agents (idée de **préférences**).
- Le centre de programmation et de planification...
  - sélectionne et planifie les prises de vue pour chaque journée,
  - les requêtes ne pouvant pas être toutes réalisées en une journée en raison des contraintes physiques.
  - cette sélection doit satisfaire des contraintes d'**équité** et d'**efficacité** (la constellation ne doit pas être sous-exploitée).

Comment sélectionner les requêtes de manière à satisfaire les contraintes physiques et les contraintes d'efficacité et d'équité ?

La constellation Pléiades

- Constellation cofinancée de manière inégale par plusieurs pays.
- Requêtes d'observation...
  - envoyées par chaque agent (agences civiles et militaires de chaque pays) au centre de programmation et de planification,
  - simples (photographie simple acquise en un seul passage) ou complexes (photographies *stereo*, zones plus larges que le champs de l'instrument optique,...),
  - d'importances inégales pour les agents (idée de **préférences**).
- Le centre de programmation et de planification...
  - sélectionne et planifie les prises de vue pour chaque journée,
  - les requêtes ne pouvant pas être toutes réalisées en une journée en raison des contraintes physiques.
  - cette sélection doit satisfaire des contraintes d'**équité** et d'**efficacité** (la constellation ne doit pas être sous-exploitée).

Comment sélectionner les requêtes de manière à satisfaire les contraintes physiques et les contraintes d'efficacité et d'équité ?

La constellation Pléiades

- Constellation cofinancée de manière inégale par plusieurs pays.
- Requêtes d'observation...
  - envoyées par chaque agent (agences civiles et militaires de chaque pays) au centre de programmation et de planification,
  - simples (photographie simple acquise en un seul passage) ou complexes (photographies *stereo*, zones plus larges que le champs de l'instrument optique,...),
  - d'importances inégales pour les agents (idée de **préférences**).
- Le centre de programmation et de planification...
  - sélectionne et planifie les prises de vue pour chaque journée,
  - les requêtes ne pouvant pas être toutes réalisées en une journée en raison des contraintes physiques.
  - cette sélection doit satisfaire des contraintes d'**équité** et d'**efficacité** (la constellation ne doit pas être sous-exploitée).

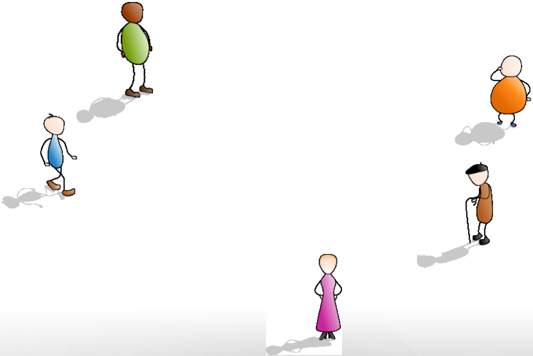
Comment sélectionner les requêtes de manière à satisfaire les contraintes physiques et les contraintes d'efficacité et d'équité ?

Le problème de partage...

Le problème de partage...

Entrées

- Un ensemble fini $\mathscr{N}$ d'**agents** .



Le problème de partage. . .

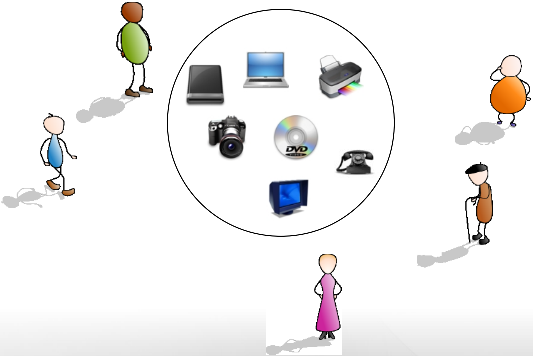
Entrées

- Un ensemble fini $\mathcal{N}$ d'**agents** .
- Une **ressource** commune limitée.

Le problème de partage...

Entrées

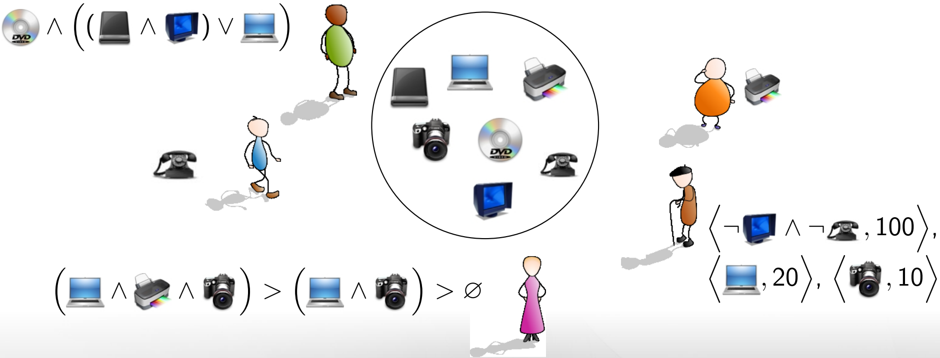
- Un ensemble fini $\mathscr{N}$ d'**agents** .
- Une **ressource** commune limitée.



Le problème de partage...

Entrées

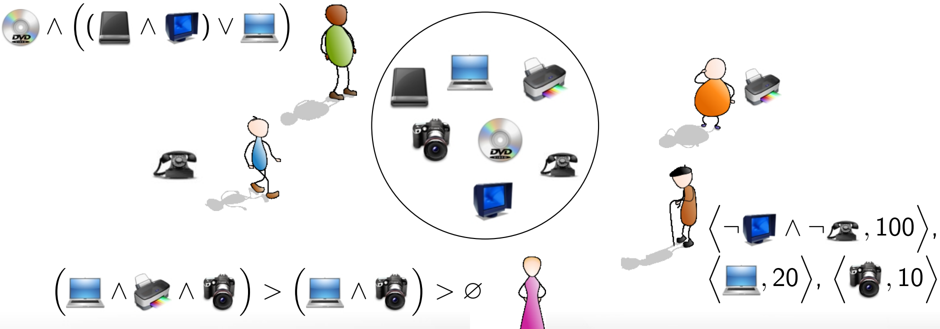
- Un ensemble fini $\mathscr{N}$ d'**agents** exprimant des **demandes** et des **préférences** sur la ressource.
- Une **ressource** commune limitée.



Le problème de partage...

Entrées

- Un ensemble fini $\mathcal{N}$ d'**agents** exprimant des **demandes** et des **préférences** sur la ressource.
- Une **ressource** commune limitée.
- Un ensemble de **contraintes** (physiques, légales, morales,...).

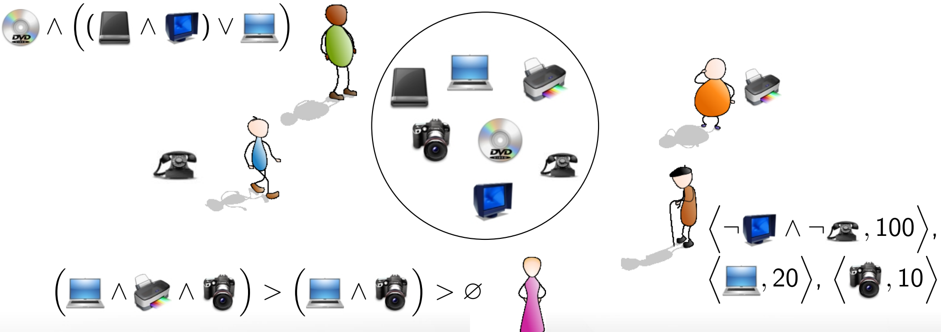


Un lot ne doit pas dépasser la capacité de transport d'un agent.

Le problème de partage...

Entrées

- Un ensemble fini $\mathscr{N}$ d'**agents** exprimant des **demandes** et des **préférences** sur la ressource.
- Une **ressource** commune limitée.
- Un ensemble de **contraintes** (physiques, légales, morales,...).
- Un critère d'**optimisation** ou de **décision**.



Un lot ne doit pas dépasser la capacité de transport d'un agent.

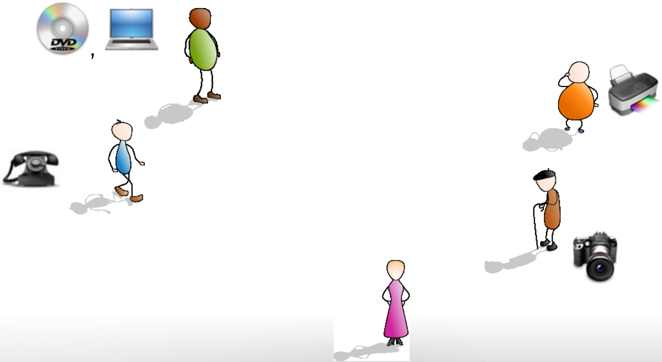
Le problème de partage. . .

Entrées

- Un ensemble fini $\mathscr{N}$ d'**agents** exprimant des **demandes** et des **préférences** sur la ressource.
- Une **ressource** commune limitée.
- Un ensemble de **contraintes** (physiques, légales, morales,. . .).
- Un critère d'**optimisation** ou de **décision**.

Sortie

- Une allocation d'une partie ou de la totalité de la ressource à chaque agent qui vérifie les contraintes sur la ressource et qui optimise ou vérifie le critère donné.



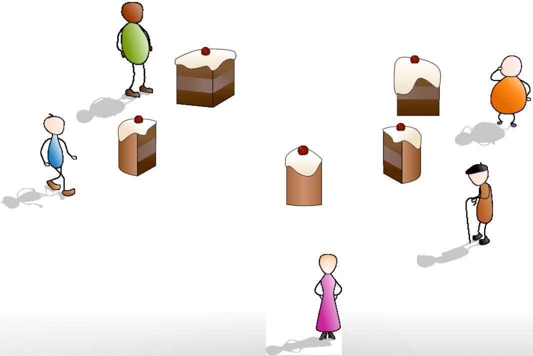
Le problème de partage...

Entrées

- Un ensemble fini $\mathscr{N}$ d'**agents** exprimant des **demandes** et des **préférences** sur la ressource.
- Une **ressource** commune limitée.
- Un ensemble de **contraintes** (physiques, légales, morales,...).
- Un critère d'**optimisation** ou de **décision**.

Sortie

- Une allocation d'une partie ou de la totalité de la ressource à chaque agent qui vérifie les contraintes sur la ressource et qui optimise ou vérifie le critère donné.



Objectifs de la thèse

Une étude théorique et pratique des problèmes de partage équitables de ressource sur des **domaines combinatoires** et **sous contraintes**, dont le problème Pléiades est une instance :

- **modélisation** des éléments du problème ;
- **représentation formelle** du problème, et analyse de **complexité** ;
- développement d'**algorithmes** de résolution.

Autres exemples d'applications

Approche non limitée au problème Pléiades

- Problèmes d'affectations de tâches ou de sujets
- Problèmes d'enchères combinatoires [**?**]
- Partage de réseaux informatiques, problèmes d'emplois du temps, partage de l'espace aérien [**?**],. . .

Partie 1

Modélisation

Partie 1 : Modélisation

Le problème de partage

Entrées

- Un ensemble $\mathscr{N}$ d'**agents** exprimant des **préférences** sur la ressource.
- Une ressource commune limitée.
- Un ensemble de **contraintes** (physiques, légales, morales,...).
- **Un critère** d'optimisation ou de décision.

Sortie

- Une allocation d'une partie ou de la totalité de la ressource à chaque agent qui vérifie les contraintes sur la ressource et qui optimise ou vérifie le critère donné.

Le problème de partage

Entrées

- Un ensemble $\mathscr{N}$ d'**agents** exprimant des **préférences** sur la ressource.
- Une ressource commune limitée.

$\rightsquigarrow$ Ressource continue, discrète, indivisible, mixte ;

$\rightsquigarrow$ Possibilité de compensations monétaires.

- Un ensemble de **contraintes** (physiques, légales, morales,. . .).
- **Un critère** d'optimisation ou de décision.

Sortie

- Une allocation d'une partie ou de la totalité de la ressource à chaque agent qui vérifie les contraintes sur la ressource et qui optimise ou vérifie le critère donné.

Le problème de partage

Entrées

- Un ensemble $\mathcal{N}$ d'**agents** exprimant des **préférences** sur la ressource.
- Une ressource commune limitée.

  $\rightsquigarrow$ Ressource continue, discrète, indivisible, mixte ;

  $\rightsquigarrow$ Possibilité de compensations monétaires.
- Un ensemble de **contraintes** (physiques, légales, morales,...).
- **Un critère** d'optimisation ou de décision.

Sortie

- Une allocation d'une partie ou de la totalité de la ressource à chaque agent qui vérifie les contraintes sur la ressource et qui optimise ou vérifie le critère donné.

Ressource indivisible, part, partage

- Ressource indivisible : ensemble d'objets $\mathcal{O}$.
- Part d'un agent : $\pi \subseteq \mathcal{O}$.
- Partage : $\overrightarrow{\pi} \in 2^{\mathcal{O}^n}$.

Le problème de partage

Entrées	• Un ensemble fini $\mathscr{N}$ d'**agents** exprimant des **préférences** sur la ressource. • **La ressource** $\rightsquigarrow$ un ensemble fini $\mathscr{O}$ d'objets indivisibles. • Un ensemble de contraintes (physiques, légales, morales,...). • **Un critère** d'optimisation ou de décision.
Sortie	• Une allocation d'une partie ou de la totalité de la ressource à chaque agent qui vérifie les contraintes sur la ressource et qui optimise ou vérifie le critère donné.

Contraintes sur la ressource

Contrainte d'admissibilité, partage admissible

- Contrainte : sous-ensemble $C \subseteq 2^{\mathscr{O}^n}$.
- Partage admissible : partage $\overrightarrow{\pi} \in \bigcap_{C \in \mathscr{C}} C$.

Contrainte de préemption

Un objet ne peut être attribué qu'à une seule personne au maximum :

$$C_{preempt} = \{\overrightarrow{\pi} \mid \forall i \neq j,\ \pi_i \cap \pi_j \neq \varnothing\}$$

Contraintes sur la ressource

Contrainte d'admissibilité, partage admissible

- Contrainte : sous-ensemble $C \subseteq 2^{\mathscr{O}^n}$.
- Partage admissible : partage $\overrightarrow{\pi} \in \bigcap_{C \in \mathscr{C}} C$.

Contrainte de préemption

Un objet ne peut être attribué qu'à une seule personne au maximum :

$$C_{preempt} = \{\overrightarrow{\pi} \mid \forall i \neq j,\ \pi_i \cap \pi_j \neq \varnothing\}$$

Le problème de partage

Entrées

- Un ensemble fini $\mathscr{N}$ d'**agents** exprimant des préférences sur la ressource.
- **La ressource** $\rightsquigarrow$ un ensemble fini $\mathscr{O}$ d'objets indivisibles.
- **Des contraintes** $\rightsquigarrow$ un ensemble fini $\mathscr{C} \subset 2^{2^{\mathscr{O}^n}}$.
- **Un critère** d'optimisation ou de décision.

Sortie

- Une allocation d'une partie ou de la totalité de la ressource à chaque agent qui vérifie les contraintes sur la ressource et qui optimise ou vérifie le critère donné.

Structure de préférence

Modèle classique en théorie de la décision :

Structure de préférence

Relation binaire réflexive $\Re_S$ sur l'ensemble des alternatives $\mathscr{E}$.
$x \Re_S y \Leftrightarrow x$ est au moins aussi bonne que y.

Grands types de structures de préférence

- Structure de préférence ordinale.
    - Structure de préférence dichotomique.
- Structure de préférence cardinale.
- Semi-ordres (modèles à seuil), ordres d'intervalle (modèles à seuil variable), structure de préférence floue,. . .

Grands types de structures de préférence

- Structure de préférence ordinale.
  - Structure de préférence dichotomique.
- Structure de préférence cardinale.
- Semi-ordres (modèles à seuil), ordres d'intervalle (modèles à seuil variable), structure de préférence floue,. . .

Structure de préférence ordinale

Un préordre complet $\succeq$ sur les alternatives ($\Re_S$ + transitivité + complétude).

Grands types de structures de préférence

- Structure de préférence ordinale.
  - Structure de préférence dichotomique.
- Structure de préférence cardinale.
- Semi-ordres (modèles à seuil), ordres d'intervalle (modèles à seuil variable), structure de préférence floue,...

Structure de préférence ordinale

Un préordre complet $\succeq$ sur les alternatives ($\Re_S$ + transitivité + complétude).

Structure de préférence dichotomique

Forme dégénérée de préférences ordinales, à deux classes d'équivalence :

- un ensemble de «bonnes» alternatives,
- un ensemble de «mauvaises» alternatives.

Grands types de structures de préférence

- Structure de préférence ordinale.
    - Structure de préférence dichotomique.
- Structure de préférence cardinale.
- Semi-ordres (modèles à seuil), ordres d'intervalle (modèles à seuil variable), structure de préférence floue,...

Structure de préférence cardinale

Enrichissement du modèle ordinal par une **fonction d'utilité** $u : \mathscr{E} \to \mathscr{V}$.
$\mathscr{V}$ espace de valuation totalement ordonné (par exemple $\mathbb{R}$, $\mathbb{N}$).

Grands types de structures de préférence

- Structure de préférence ordinale.
  - Structure de préférence dichotomique.
- Structure de préférence cardinale.
- Semi-ordres (modèles à seuil), ordres d'intervalle (modèles à seuil variable), structure de préférence floue,. . .

Espace cible des préférences

Sur quel ensemble d'alternatives les agents expriment-ils leurs préférences ?

Hypothèse (préférences non exogènes) : Chaque agent n'exprime ses préférences que sur les allocations qu'il reçoit (en particulier, il ne tient pas compte de ce qu'obtiennent les autres agents dans le partage).

ensemble des alternatives = ensemble des parts possibles. Pour l'agent i, $2^{\mathscr{O}}$.

Le problème de partage

Entrées	• Un ensemble fini $\mathscr{N}$ d'**agents** exprimant des **préférences** sur la ressource $\rightsquigarrow$ sous la forme de préordres $\succeq_i$ ou fonctions d'utilité u_i. • **La ressource** $\rightsquigarrow$ un ensemble fini $\mathscr{O}$ d'objets indivisibles. • **Des contraintes** $\rightsquigarrow$ un ensemble fini $\mathscr{C} \subset 2^{2^{\mathscr{O}^n}}$. • Un critère d'optimisation ou de décision.
Sortie	• Une allocation d'une partie ou de la totalité de la ressource à chaque agent qui vérifie les contraintes sur la ressource et qui optimise ou vérifie le critère donné.

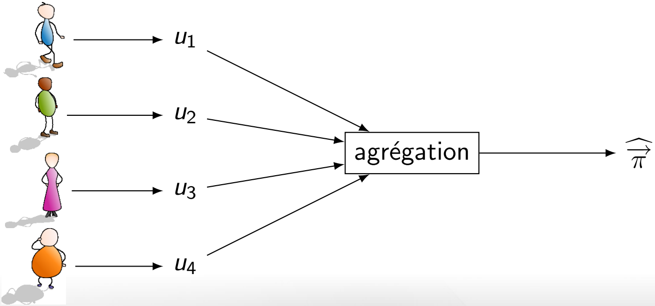
Problématique de l'agrégation de préférences

Problème posé : *Comment distribuer la ressource entre les agents, de manière à tenir compte de manière équitable de leurs préférences souvent antagonistes ?*

Problématique de l'agrégation de préférences

Problème posé : *Comment distribuer la ressource entre les agents, de manière à tenir compte de manière équitable de leurs préférences souvent antagonistes ?*

La théorie du **welfarisme cardinal** traite le problème de décision collective en attachant à chaque alternative faisable le vecteur des utilités individuelles $(u_1, \ldots, u_n)$.



Le *welfarisme* cardinal

La théorie du **welfarisme cardinal** traite le problème de décision collective en attachant à chaque alternative faisable le vecteur des utilités individuelles $(u_1, \ldots, u_n)$.

Ordre de bien-être social

Un ordre de bien-être social est un préordre $\preceq$ sur $\mathscr{V}^n$.

Un ordre de bien-être social reflète l'ordre de préférences collectif vis-à-vis de l'ensemble des partages possibles.

Fonction d'utilité collective

Une fonction d'utilité collective est une fonction de $\mathscr{V}^n$ dans $\mathscr{V}$.

Une fonction d'utilité collective représente un ordre de bien-être social particulier.

Propriétés basiques des ordres de bien-être social

Unanimité

Un vecteur d'utilités $\overrightarrow{u}$ Pareto-domine un autre vecteur d'utilités $\overrightarrow{v}$ ssi pour tout i, $u_i \geq v_i$ et il existe un i t.q. $u_i > v_i$.
Un vecteur non Pareto-dominé est dit Pareto-efficace.
Un ordre de bien-être social $\preceq$ satisfait l'unanimité ssi :

$$\overrightarrow{u} \text{ Pareto-domine } \overrightarrow{v} \Rightarrow \overrightarrow{u} \succ \overrightarrow{v}.$$

Anonymat

Un ordre de bien-être social garantit l'anonymat ssi il est indifférent entre un partage et ce même partage dans lequel on a permuté les parts entre les agents.

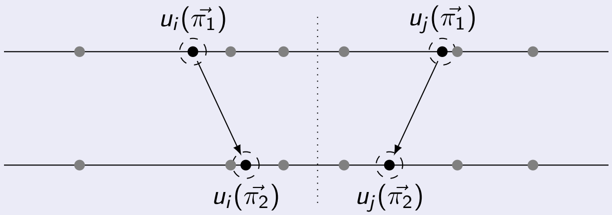
L'équité dans les partages

- Propriétés des ordres de bien-être social :
  - Anonymat (propriété d'équité *ex-ante*).
  - Juste part garantie.
  - Réduction des inégalités.
- Propriétés des partages :
  - Test de juste part.
  - Théorie de la mesure des inégalités. Indices d'Atkinson, de Gini, courbe de Lorenz. . .
  - Test d'absence d'envie.

L'équité dans les partages

- Propriétés des ordres de bien-être social :
  - Anonymat (propriété d'équité *ex-ante*).
  - Juste part garantie.
  - Réduction des inégalités.
- Propriétés des partages :
  - Test de juste part.
  - Théorie de la mesure des inégalités. Indices d'Atkinson, de Gini, courbe de Lorenz. . .
  - Test d'absence d'envie.

Réduction des inégalités (principe de Pigou-Dalton)



L'équité dans les partages

- Propriétés des ordres de bien-être social :
  - Anonymat (propriété d'équité *ex-ante*).
  - Juste part garantie.
  - Réduction des inégalités.
- Propriétés des partages :
  - Test de juste part.
  - Théorie de la mesure des inégalités. Indices d'Atkinson, de Gini, courbe de Lorenz. . .
  - Test d'absence d'envie.

Absence d'envie

Un partage est sans envie si et seulement si chaque agent est autant ou plus satisfait avec sa propre part qu'il ne le serait avec la part d'un autre.

Ordres de bien-être social classiques

- Ordre utilitariste classique.
- Ordre égalitariste.
- Ordre égalitariste leximin.
- Compromis entre utilitarisme classique et égalitarisme : Nash ($\times$), familles OWA et somme des puissances,. . .

Ordres de bien-être social classiques

- Ordre utilitariste classique.
- Ordre égalitariste.
- Ordre égalitariste leximin.
- Compromis entre utilitarisme classique et égalitarisme : Nash ($\times$), familles OWA et somme des puissances,...

Ordre utilitariste classique [Harsanyi]

$\overrightarrow{u} \preceq \overrightarrow{v} \Leftrightarrow \sum_{i=1}^{n} u_i \leq \sum_{i=1}^{n} v_i$.

Caractéristiques

Les agents sont des producteurs d'utilité (principe d'**adéquation**).
Ordre indifférent aux inégalités $\leadsto$ peut conduire à des décisions très inégalitaires.

Ordres de bien-être social classiques

- Ordre utilitariste classique.
- Ordre égalitariste.
- Ordre égalitariste leximin.
- Compromis entre utilitarisme classique et égalitarisme : Nash ($\times$), familles OWA et somme des puissances,. . .

Ordre égalitariste [Rawls]

$\overrightarrow{u} \preceq \overrightarrow{v} \Leftrightarrow \min_{i=1}^{n} u_i \leq \min_{i=1}^{n} v_i$.

Caractéristiques

Il ne prend en compte que l'agent le moins satisfait (principe de **compensation**) $\leadsto$ tendance naturelle à l'équité (dans le sens « équilibre des utilités »).

Ordres de bien-être social classiques

- Ordre utilitariste classique.
- Ordre égalitariste.
- Ordre égalitariste leximin.
- Compromis entre utilitarisme classique et égalitarisme : Nash ($\times$), familles OWA et somme des puissances,. . .

Ordre égalitariste [Rawls]

$\overrightarrow{u} \preceq \overrightarrow{v} \Leftrightarrow \min_{i=1}^{n} u_i \leq \min_{i=1}^{n} v_i$.

Caractéristiques

Il ne prend en compte que l'agent le moins satisfait (principe de **compensation**) $\leadsto$ tendance naturelle à l'équité (dans le sens « équilibre des utilités »).
D'un autre côté, il peut conduire à des décisions non Pareto-optimales (effet de noyade).

Ordres de bien-être social classiques

- Ordre utilitariste classique.
- Ordre égalitariste.
- Ordre égalitariste leximin.
- Compromis entre utilitarisme classique et égalitarisme : Nash ($\times$), familles OWA et somme des puissances,. . .

Ordre égalitariste [Rawls]

$\overrightarrow{u} \preceq \overrightarrow{v} \Leftrightarrow \min_{i=1}^{n} u_i \leq \min_{i=1}^{n} v_i$.

Ordre social égalitariste et Pareto-efficacité

$\langle 1, 1, 1, 1 \rangle \sim \langle 1000, 1, 1000, 1000 \rangle$, alors que $\langle 1, 1, 1, 1 \rangle$ et $\langle 1000, 1, 1000, 1000 \rangle$ sont très différents !

Ordres de bien-être social classiques

- Ordre utilitariste classique.
- Ordre égalitariste.
- Ordre égalitariste leximin.
- Compromis entre utilitarisme classique et égalitarisme : Nash ($\times$), familles OWA et somme des puissances,. . .

Ordre social Leximin [Sen, 1970 ; Kolm, 1972]

Soit $\overrightarrow{x}$ un vecteur. Nous notons $\overrightarrow{x^{\uparrow}}$ la version triée de $\overrightarrow{x}$.
$\overrightarrow{u} \succ_{leximin} \overrightarrow{v} \Leftrightarrow \exists k$ tel que $\forall i \leq k,\ u_i^{\uparrow} = v_i^{\uparrow}$ et $u_{k+1}^{\uparrow} > v_{k+1}^{\uparrow}$.
En d'autres termes, une comparaison lexicographique sur les vecteurs triés.

Effectuer une comparaison leximin. . .

Deux vecteurs à comparer : $\overrightarrow{u} = \langle 4, 10, 3, 5 \rangle$ et $\overrightarrow{v} = \langle 4, 3, 6, 6 \rangle$.

- Nous trions les vecteurs : $\begin{cases} \overrightarrow{u}^{\uparrow} = \langle 3, 4, 5, 10 \rangle \\ \overrightarrow{v}^{\uparrow} = \langle 3, 4, 6, 6 \rangle \end{cases}$
- Nous comparons les vecteurs lexicographiquement : $\overrightarrow{u}^{\uparrow} \prec_{lexico} \overrightarrow{v}^{\uparrow}$

Ordres de bien-être social classiques

- Ordre utilitariste classique.
- Ordre égalitariste.
- Ordre égalitariste leximin.
- Compromis entre utilitarisme classique et égalitarisme : Nash ($\times$), familles OWA et somme des puissances,. . .

Ordre social Leximin [Sen, 1970 ; Kolm, 1972]

Soit $\overrightarrow{x}$ un vecteur. Nous notons $\overrightarrow{x^{\uparrow}}$ la version triée de $\overrightarrow{x}$.
$\overrightarrow{u} \succ_{leximin} \overrightarrow{v} \Leftrightarrow \exists k$ tel que $\forall i \leq k,\ u_i^{\uparrow} = v_i^{\uparrow}$ et $u_{k+1}^{\uparrow} > v_{k+1}^{\uparrow}$.
En d'autres termes, une comparaison lexicographique sur les vecteurs triés.

Caractéristiques

Il prend en compte tous les agents, dans l'ordre de leur niveau de satisfaction $\leadsto$ tendance naturelle à l'équité.
Il raffine l'ordre égalitariste et l'ordre de Pareto.

Ordres de bien-être social classiques

- Ordre utilitariste classique.
- Ordre égalitariste.
- Ordre égalitariste leximin.
- Compromis entre utilitarisme classique et égalitarisme : Nash ($\times$), familles OWA et somme des puissances,. . .

Ordre social Leximin [Sen, 1970 ; Kolm, 1972]

Soit $\overrightarrow{x}$ un vecteur. Nous notons $\overrightarrow{x^{\uparrow}}$ la version triée de $\overrightarrow{x}$.
$\overrightarrow{u} \succ_{leximin} \overrightarrow{v} \Leftrightarrow \exists k$ tel que $\forall i \leq k,\ u_i^{\uparrow} = v_i^{\uparrow}$ et $u_{k+1}^{\uparrow} > v_{k+1}^{\uparrow}$.
En d'autres termes, une comparaison lexicographique sur les vecteurs triés.

Ordre leximin et Pareto-efficacité

$\langle 1, 1, 1, 1\rangle \prec \langle 1000, 1, 1000, 1000\rangle$ (La deuxième valeur des vecteurs triés est discriminante).

Ordres de bien-être social classiques

- Ordre utilitariste classique.
- Ordre égalitariste.
- Ordre égalitariste leximin.
- Compromis entre utilitarisme classique et égalitarisme : Nash ($\times$), familles OWA et somme des puissances,...

Équité et efficacité dans le partage

Deux visions différentes :

- **Réduction des inégalités :**
  - Agrégation des utilités des agents par un ordre de bien-être social ou une fonction d'utilité collective compatible avec le principe de réduction des inégalités (et avec l'ordre de Pareto).
  - Exemples : leximin.
  - Requiert une **comparaison interpersonnelle** des utilités.
- **Absence d'envie :**
  - On cherche un partage sans envie (et (Pareto-)efficace).
  - Critère fondé uniquement sur une appréciation personnelle des agents.
  - En particulier, propriété purement **ordinale**.
  - Cependant, propriété pas toujours pertinente, pour des raisons éthiques ou techniques.

Équité et efficacité dans le partage

Deux visions différentes :

- **Réduction des inégalités :**
  - Agrégation des utilités des agents par un ordre de bien-être social ou une fonction d'utilité collective compatible avec le principe de réduction des inégalités (et avec l'ordre de Pareto).
  - Exemples : leximin.
  - Requiert une **comparaison interpersonnelle** des utilités.
- **Absence d'envie :**
  - On cherche un partage sans envie (et (Pareto-)efficace).
  - Critère fondé uniquement sur une appréciation personnelle des agents.
  - En particulier, propriété purement **ordinale**.
  - Cependant, propriété pas toujours pertinente, pour des raisons éthiques ou techniques.

Le problème de partage

Entrées

- Un ensemble fini $\mathscr{N}$ d'**agents** exprimant des **préférences** sur la ressource $\leadsto$ sous la forme de préordres $\succeq_i$ ou fonctions d'utilité u_i.
- **La ressource** $\leadsto$ un ensemble fini $\mathscr{O}$ d'objets indivisibles.
- **Des contraintes** $\leadsto$ un ensemble fini $\mathscr{C} \subset 2^{2^{\mathscr{O}^n}}$.
- **Un critère** $\leadsto$ maximisation d'un ordre de bien-être social ou d'une fonction d'utilité collective, ou efficacité et absence d'envie.

Sortie

- Une allocation d'une partie ou de la totalité de la ressource à chaque agent qui vérifie les contraintes sur la ressource et qui optimise ou vérifie le critère donné.

Partie 1 : Modélisation

Droits exogènes ?

La propriété d'anonymat (égalité ex-ante) est-elle toujours pertinente ?

Ce n'est pas toujours le cas :

- Ressource cofinancée de manière inégale par les agents (Pléiades).
- Agents représentant des populations de tailles différentes.
- Droits divers (ancienneté, situation familiale, ...).

Comment prendre en compte l'inégalité des droits exogènes dans le cadre du welfarisme cardinal ?

Droits exogènes ?

La propriété d'anonymat (égalité ex-ante) est-elle toujours pertinente ?

Ce n'est pas toujours le cas :

- Ressource cofinancée de manière inégale par les agents (Pléiades).
- Agents représentant des populations de tailles différentes.
- Droits divers (ancienneté, situation familiale, . . .).

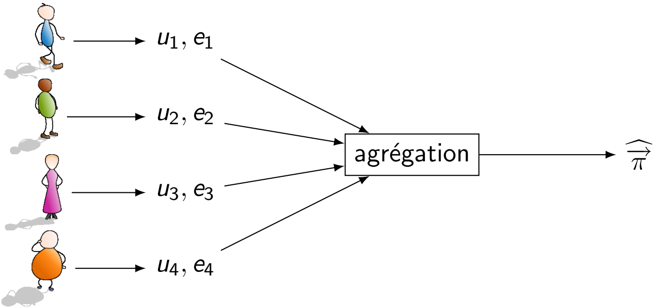
Comment prendre en compte l'inégalité des droits exogènes dans le cadre du welfarisme cardinal ?

Hypothèses

- Les droits exogènes sont représentés par un vecteur d'entiers $\overrightarrow{e}$.
- Ils sont fixes au cours du problème, et sont acceptés par tous les agents avant le début du problème.

Hypothèses

- Les droits exogènes sont représentés par un vecteur d'entiers $\overrightarrow{e}$.
- Ils sont fixes au cours du problème, et sont acceptés par tous les agents avant le début du problème.



Présentation du principe

Nous cherchons un moyen de prendre en compte l'**inégalité des droits exogènes** dans le processus de prise de décision collective.

Idée

L'idée est de remplacer chaque agent par autant de **clones** qu'il possède de droits (ou un nombre proportionnel si les droits ne sont pas entiers), en répartissant l'utilité de chaque agent initial entre ses clones.

Principe proposé dans quelques travaux [**?**] toujours dans un contexte égalitariste.

Illustration du principe

Agents avec droits inégaux

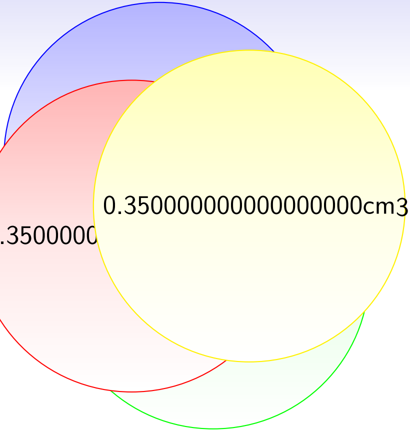


Illustration du principe

Agents avec droits inégaux

Duplication des agents

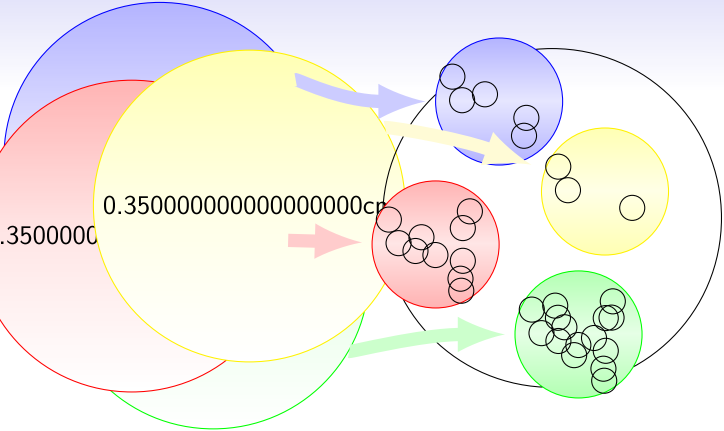


Illustration du principe

Agents avec droits inégaux

Duplication des agents

Décision collective

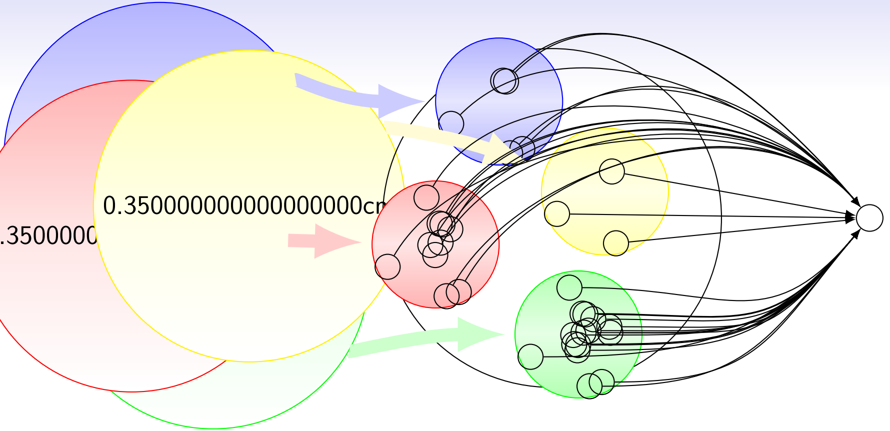


Illustration du principe

Agents avec droits inégaux — Duplication des agents — Décision collective

$$g_{\vec{e}}$$

$$\left.\begin{array}{c}(u_1, e_1)\\ \vdots\\ (u_n, e_n)\end{array}\right\} \xmapsto{\;\div\;} \left\{\begin{array}{rcl} r_1 &=& u_1 \div e_1\\ &\vdots& \\ r_1 &=& u_1 \div e_1\\ && \vdots \\ r_n &=& u_n \div e_n\\ &\vdots& \\ r_n &=& u_n \div e_n\end{array}\right\} \xmapsto{\;g\;} \begin{array}{c} u_c\\ =\\ g(\vec{u} \triangleright \frac{\div}{\vec{e}})\\ =\\ g_{\vec{e}}(u_1, \ldots, u_n)\end{array}$$

Fonction de répartition

Une fonction de répartition est une fonction $\div : \mathbb{R} \times \mathbb{N} \rightarrow \mathbb{R}$.

Exemples : $u \div e = u/e$, $u \div e = u$.

Interprétation de la fonction de répartition

Répartition de l'utilité des agents d'une population entre les clones.

- **Division :** $u \div e \stackrel{def}{=} u/e$.
  - L'utilité est divisée entre les clones (biens préemptifs).
  - Exemple : répartition de nourriture entre plusieurs pays touchés par la famine.
- **Réplication :** $u \div e \stackrel{def}{=} u$.
  - L'utilité est dupliquée entre les clones (prospérité).
  - Exemple : répartition d'infrastructures de loisir dans plusieurs villes de tailles variables.

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.

Équité généralisée

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

Ordres de bien-être social étendu

- Extension des ordres de bien-être social et fonctions d'utilité collective classiques, fondée sur le principe de duplication des agents :
  - Ordre utilitariste classique, égalitariste (lexi)min, somme des puissances, OWA.
  - Fonctions de répartition division et réplication.
- Caractérisation de ces ordres de bien-être social étendus à l'aide des propriétés généralisés.

Conclusion sur les droits inégaux

Notre contribution :

- Un cadre formel pour la construction d'ordres de bien-être social et fonctions d'utilité collective à droits inégaux : le **schéma de duplication des agents**.
- Une extension des ordres de bien-être social et fonctions d'utilité collective classiques fondée sur ce schéma de duplication des agents.
- Une extension des propriétés classiques pour la prise en compte de droits inégaux.
- La proposition de nouvelles propriétés pour caractériser le bon fonctionnement de ces droits inégaux et la vérification de ces propriétés sur les ordres de bien-être social classiques.

À explorer :

- Vers une axiomatisation complète de la prise de décision en présence de droits inégaux.
- Droits inégaux représentés par des ordres de priorité.

Conclusion sur les droits inégaux

Notre contribution :

- Un cadre formel pour la construction d'ordres de bien-être social et fonctions d'utilité collective à droits inégaux : le **schéma de duplication des agents**.
- Une extension des ordres de bien-être social et fonctions d'utilité collective classiques fondée sur ce schéma de duplication des agents.
- Une extension des propriétés classiques pour la prise en compte de droits inégaux.
- La proposition de nouvelles propriétés pour caractériser le bon fonctionnement de ces droits inégaux et la vérification de ces propriétés sur les ordres de bien-être social classiques.

À explorer :

- Vers une axiomatisation complète de la prise de décision en présence de droits inégaux.
- Droits inégaux représentés par des ordres de priorité.

Partie 2

Représentation compacte et complexité

Partie 2 : Représentation compacte et complexité

Un langage de représentation

Entrées

- Un ensemble fini $\mathscr{N}$ d'**agents** exprimant des **préférences** sur la ressource $\rightsquigarrow$ sous la forme de préordres $\succeq_i$ ou fonctions d'utilité u_i.
- **La ressource** $\rightsquigarrow$ un ensemble fini $\mathscr{O}$ d'objets indivisibles.
- **Des contraintes** $\rightsquigarrow$ un ensemble fini $\mathscr{C}$ de sous-ensembles de $2^{\mathscr{O}^n}$: $\mathscr{C} \subset 2^{2^{\mathscr{O}^n}}$
- **Un critère** $\rightsquigarrow$ maximisation d'un ordre de bien-être social ou d'une fonction d'utilité collective, ou efficacité et absence d'envie.
- Éventuellement des droits exogènes inégaux $\overrightarrow{e}$.

Sortie

- Une allocation d'une partie ou de la totalité de la ressource à chaque agent qui vérifie les contraintes sur la ressource et qui optimise ou vérifie le critère donné.

Cette description mathématique ne précise pas comment les instances sont **représentées formellement**, et comment elles doivent être implantées informatiquement.

Ces précisions sont cruciales notamment pour la représentation des **contraintes** et des **préférences**.

Un langage de représentation

Entrées

- Un ensemble fini $\mathcal{N}$ d'**agents** exprimant des **préférences** sur la ressource $\rightsquigarrow$ sous la forme de préordres $\succeq_i$ ou fonctions d'utilité u_i.
- **La ressource** $\rightsquigarrow$ un ensemble fini $\mathcal{O}$ d'objets indivisibles.
- **Des contraintes** $\rightsquigarrow$ un ensemble fini $\mathcal{C}$ de sous-ensembles de $2^{\mathcal{O}^n}$: $\mathcal{C} \subset 2^{2^{\mathcal{O}^n}}$
- **Un critère** $\rightsquigarrow$ maximisation d'un ordre de bien-être social ou d'une fonction d'utilité collective, ou efficacité et absence d'envie.
- Éventuellement des droits exogènes inégaux $\overrightarrow{e}$.

Sortie

- Une allocation d'une partie ou de la totalité de la ressource à chaque agent qui vérifie les contraintes sur la ressource et qui optimise ou vérifie le critère donné.

Cette description mathématique ne précise pas comment les instances sont **représentées formellement**, et comment elles doivent être implantées informatiquement.
Ces précisions sont cruciales notamment pour la représentation des **contraintes** et des **préférences**.

Représentation compacte de préférences

Exemple

Problème de partage à 2 objets o_1 et o_2.
Expression de la fonction d'utilité :
$u(\varnothing) = 0$, $u(o_1) = 5$, $u(o_2) = 7$, $u(\{o_1, o_2\}) = 3$.

Représentation compacte de préférences

Exemple

Problème de partage à 4 objets o_1, o_2, o_3 et o_4.
Expression de la fonction d'utilité :
$u(\varnothing) = 0$, $u(o_1) = 5$, $u(o_2) = 7$, $u(o_3) = 2$, $u(o_4) = 8$, $u(\{o_1, o_2\}) = 3$, $u(\{o_1, o_3\}) = 5$, $u(\{o_1, o_4\}) = 3$, $u(\{o_2, o_3\}) = 0$, $u(\{o_2, o_4\}) = 6$, $u(\{o_3, o_4\}) = 2$, $u(\{o_1, o_2, o_3\}) = 8$, $u(\{o_1, o_2, o_4\}) = 9$, $u(\{o_1, o_3, o_4\}) = 10$, $u(\{o_2, o_3, o_4\}) = 3$, $u(\{o_1, o_2, o_3, o_4\}) = 10$.

Représentation compacte de préférences

Exemple

Problème de partage à 20 objets $o_1,\ldots, o_{20}$

Expression de la fonction d'utilité :

$u(\varnothing) = 0$, $u(o_1) = 5$, $u(o_2) = 7$, $u(o_3) = 2$, $u(o_4) = 8$ $u(o_5) = 5$, $u(o_6) = 0$, $u(o_7) = 1$, $u(o_8) = 15$ $u(o_9) = 4$, $u(o_{10}) = 6$, $u(o_{11}) = 6$, $u(o_{12}) = 8$, $u(o_{13}) = 5$, $u(o_{14}) = 7$, $u(o_{15}) = 2$, $u(o_{16}) = 8$, $u(o_{17}) = 7$, $u(o_{18}) = 2$, $u(o_{19}) = 8$, $u(o_{20}) = 7$, $u(\{o_1, o_2\}) = 15$,

[redacted]

Représentation compacte de préférences

Exemple

Problème de partage à 20 objets $o_1, \ldots, o_{20}$

Expression de la fonction d'utilité :

$u(\varnothing) = 0$, $u(o_1) = 5$, $u(o_2) = 7$, $u(o_3) = 2$, $u(o_4) = 8$ $u(o_5) = 5$, $u(o_6) = 0$, $u(o_7) = 1$, $u(o_8) = 15$ $u(o_9) = 4$, $u(o_{10}) = 6$, $u(o_{11}) = 6$, $u(o_{12}) = 8$, $u(o_{13}) = 5$, $u(o_{14}) = 7$, $u(o_{15}) = 2$, $u(o_{16}) = 8$, $u(o_{17}) = 7$, $u(o_{18}) = 2$, $u(o_{19}) = 8$, $u(o_{20}) = 7$, $u(\{o_1, o_2\}) = 15$,
[redacted]

1048576 utilités $\rightsquigarrow$ l'expression nécessite plus de 12 jours (à raison d'une utilité par seconde).

Représentation compacte de préférences

Trois réponses possibles à l'explosion combinatoire :

1. L'ignorer et supposer que le nombre d'objets reste faible [**?**].
2. Imposer des hypothèses restrictives sur les préférences (par exemple : additivité) qui rendent l'expression possible [**?**] et [**?**].
3. Utiliser un **langage de représentation compacte**.

Représentation compacte de préférences

Trois réponses possibles à l'explosion combinatoire :

1. L'ignorer et supposer que le nombre d'objets reste faible [**?**].
2. Imposer des hypothèses restrictives sur les préférences (par exemple : additivité) qui rendent l'expression possible [**?**] et [**?**].
3. Utiliser un **langage de représentation compacte**.

Représentation compacte de préférences

Trois réponses possibles à l'explosion combinatoire :

1. L'ignorer et supposer que le nombre d'objets reste faible [**?**].
2. Imposer des hypothèses restrictives sur les préférences (par exemple : additivité) qui rendent l'expression possible [**?**] et [**?**].
3. Utiliser un **langage de représentation compacte**.

Langages d'expression compacte des préférences

- Préférences dichotomiques :
  - représentation logique des préférences dichotomiques.
- Préférences ordinales :
  - buts priorisés (best-out, discrimin, leximin...),
  - CP-nets, TCP-nets.
- Préférences cardinales :
  - langages *k*-additifs, GAI-nets,
  - langage à base de buts pondérés,
  - langages de lots pour les enchères combinatoires (OR, XOR, ...),
  - UCP-nets,
  - CSP valués.

Langages d'expression compacte des préférences

- Préférences dichotomiques :
  - représentation logique des préférences dichotomiques.
- Préférences ordinales :
  - buts priorisés (best-out, discrimin, leximin...),
  - CP-nets, TCP-nets.
- Préférences cardinales :
  - langages k-additifs, GAI-nets,
  - langage à base de buts pondérés,
  - langages de lots pour les enchères combinatoires (OR, XOR, ...),
  - UCP-nets,
  - CSP valués.

Partage et représentation compacte

Introduction de langages de représentation compacte (fondés sur la logique propositionnelle), pour les deux problèmes suivants :

- Problème de maximisation de l'utilité collective.
- Problème d'existence d'un partage efficace et sans envie.

Partie 2 : Représentation compacte et complexité

Agents, objets et partage

Partage de biens indivisibles entre des agents

- Ensemble d'**agents** $\mathscr{N} = \{1, \ldots, n\}$.
- Ensemble d'**objets** $\mathscr{O}$.
- Partage $\overrightarrow{\pi} = \langle \pi_1, \ldots, \pi_n \rangle$ ($\pi_i \subseteq \mathscr{O}$ est la **part** de l'agent i).

Expression des contraintes

Un langage propositionnel $L_{\mathscr{O}}^{alloc}$:

- ensemble des symboles propositionnels $\{alloc(o,i) \mid o \in \mathscr{O}, i \in \mathscr{N}\}$.
- ensemble des connecteurs usuels $\neg, \wedge, \vee$

Contrainte

Une contrainte est une formule de $L_{\mathscr{O}}^{alloc}$.

Exemple

La contrainte de préemption s'exprime par l'ensemble de formules :

$$\{\neg(alloc(o,i) \wedge alloc(o,j)) \mid i,j \in \mathscr{N}, i \neq j\}.$$

Un langage à base de logique et de poids

Langage d'expression des préférences :

- Un langage propositionnel $L_{\mathscr{O}}$...
  - ensemble des symboles propositionnels $\mathscr{O}$,
  - ensemble des connecteurs usuels $\neg, \wedge, \vee$
- ... et des poids $w \in \mathscr{V}$.

Un langage à base de logique et de poids

Langage d'expression des préférences :

- Un langage propositionnel $L_{\mathscr{O}}\ldots$
  - ensemble des symboles propositionnels $\mathscr{O}$,
  - ensemble des connecteurs usuels $\neg, \wedge, \vee$
- ... et des poids $w \in \mathscr{V}$.

Exemple

- $\mathscr{O} = \{$, , , , , , $\}$.
- Demandes de l'agent 1 :
  - $\Big\langle$  $\wedge \Big(($  $\wedge$  $) \vee$  $\Big), 110 \Big\rangle$,
  - $\Big\langle$  $, -10 \Big\rangle$,
  - $\Big\langle$  $\wedge$  $, 50 \Big\rangle$.

Utilité individuelle

L'utilité individuelle d'un agent exprime sa satisfaction vis-à-vis d'un partage. Elle dépend :

- de la part qu'il obtient dans le partage (et uniquement de sa part),
- des demandes pondérées qu'il a formulées,

et elle est obtenue par **agrégation** des poids des formules satisfaites, par un opérateur $\oplus$.

Utilité individuelle

Étant donnés un agent i, son ensemble de demandes pondérées Δ_i, et un partage $\overrightarrow{\pi}$, l'utilité individuelle de i s'exprime comme suit :

$$u_i(\pi_i) = \bigoplus \{w \mid \langle \varphi, w \rangle \in \Delta_i \text{ et } x_i \models \varphi\}.$$

Deux choix évidents pour $\oplus$: $+$ ou max.

Utilité individuelle

Exemple

- $\mathscr{O} = \{$, , , , , , $\}$.
- Demandes de l'agent 1 :
  - $\langle$  $\wedge$ $(($  $\wedge$  $) \vee$  $)$, 110 $\rangle$,
  - $\langle$ , -10 $\rangle$,
  - $\langle$  $\wedge$ , 50 $\rangle$.

Exemple de calcul de l'utilité individuelle ($\oplus = +$) :

$\pi_1 = \{$, , , $\}$

Utilité individuelle

Exemple

- $\mathscr{O} = \{$, , , , , , $\}$.
- Demandes de l'agent 1 :
  - $\Big\langle$ $\wedge$ $\Big(($ $\wedge$ $)$ $\vee$ $\Big)$, $110\Big\rangle$,
  - $\Big\langle$, $-10\Big\rangle$,
  - $\Big\langle$ $\wedge$ , $50\Big\rangle$.

Exemple de calcul de l'utilité individuelle ($\oplus = +$) :

$\pi_1 = \{$, , , $\} \Rightarrow u_1(\pi_1) =$ $\wedge(($$\wedge$$)\vee$$)$ 110

Utilité individuelle

Exemple

- $\mathscr{O} = \{$, , , , , , $\}$.
- Demandes de l'agent 1 :
  - $\Big\langle$  $\wedge \Big(($  $\wedge$  $) \vee$  $\Big), 110 \Big\rangle$,
  - $\Big\langle$  $, -10 \Big\rangle$,
  - $\Big\langle$  $\wedge$  $, 50 \Big\rangle$.

Exemple de calcul de l'utilité individuelle ($\oplus = +$) :

$\pi_1 = \{$, , , $\} \Rightarrow u_1(\pi_1) = 110$  -10

Utilité individuelle

Exemple

- $\mathscr{O} = \{$, , , , , , $\}$.
- Demandes de l'agent 1 :
  - $\Big\langle$  $\wedge \Big(($  $\wedge$  $) \vee$  $\Big), 110 \Big\rangle$,
  - $\Big\langle$ $, -10 \Big\rangle$,
  - $\Big\langle$  $\wedge$ $, 50 \Big\rangle$.

Exemple de calcul de l'utilité individuelle ($\oplus = +$) :

~~ $\wedge$ ~~

$\pi_1 = \{$, , , $\} \Rightarrow u_1(\pi_1) = 110 - 10 + 0$

Utilité individuelle

Exemple

- $\mathscr{O} = \{$, , , , , , $\}$.
- Demandes de l'agent 1 :
  - $\Big\langle$$\wedge \Big(($$\wedge$$) \vee$$\Big), 110 \Big\rangle$,
  - $\Big\langle$$, -10 \Big\rangle$,
  - $\Big\langle$$\wedge$$, 50 \Big\rangle$.

Exemple de calcul de l'utilité individuelle ($\oplus = +$) :

$\pi_1 = \{$, , , $\} \Rightarrow u_1(\pi_1) = 110 - 10 + 0 = 100$

Utilité collective

L'utilité collective (la satisfaction globale de la société d'agents) s'exprime comme une agrégation des utilités individuelles.

Utilité collective

Étant donné un partage $\overrightarrow{\pi}$, un ensemble d'agents $\mathscr{N}$ et leurs utilités individuelles,

$$uc(\overrightarrow{\pi}) = g(u_1(\pi_1), \ldots, u_n(\pi_n)),$$

avec g une fonction commutative et non-décroissante de $\mathscr{V}^n$ dans $\mathscr{V}$.

Le calcul de l'utilité collective passe donc par deux niveaux d'agrégation :

$$\left.\begin{array}{ccc} w_1^1, \ldots, w_{p_1}^1 & \overset{\oplus}{\mapsto} & u_1 \\ & \vdots & \\ w_1^n, \ldots, w_{p_n}^n & \overset{\oplus}{\mapsto} & u_n \end{array}\right\} \overset{g}{\mapsto} uc.$$

Utilité collective

L'utilité collective (la satisfaction globale de la société d'agents) s'exprime comme une agrégation des utilités individuelles.

Utilité collective

Étant donné un partage $\overrightarrow{\pi}$, un ensemble d'agents $\mathcal{N}$ et leurs utilités individuelles,

$$uc(\overrightarrow{\pi}) = g(u_1(\pi_1), \ldots, u_n(\pi_n)),$$

avec g une fonction commutative et non-décroissante de $\mathcal{V}^n$ dans $\mathcal{V}$.

Le calcul de l'utilité collective passe donc par deux niveaux d'agrégation :

$$\left.\begin{array}{ccc} w_1^1, \ldots, w_{p_1}^1 & \overset{\oplus}{\longmapsto} & u_1 \\ & \vdots & \\ w_1^n, \ldots, w_{p_n}^n & \overset{\oplus}{\longmapsto} & u_n \end{array}\right\} \overset{g}{\longmapsto} uc.$$

Le problème de partage

En résumé :

Instance du problème de partage

Entrées

- Un ensemble fini $\mathscr{N}$ d'**agents** exprimant des **demandes** $\{\Delta_1, \ldots, \Delta_n\}$ sous forme de formules logiques pondérées de $L_{\mathscr{O}} \times \mathscr{V}$
- Un ensemble fini $\mathscr{O}$ d'objets indivisibles.
- Un ensemble fini $\mathscr{C}$ de contraintes exprimées dans le langage logique $L_{\mathscr{O}}^{alloc}$.
- Un couple d'opérateurs d'agrégation $(\oplus, g)$.

Sortie

- Un partage $\overrightarrow{\pi} \in 2^{\mathscr{O}^n}$ tel que $\{alloc(o, i) \mid o \in \pi_i\} \models \bigwedge_{C \in \mathscr{C}} C$ et qui maximise la fonction d'utilité collective définie par :

$$uc(\overrightarrow{\pi}) = g(u_1, \ldots, u_n), \text{ avec}$$

$$u_i = \bigoplus \{w \mid \langle \varphi, w \rangle \in \Delta_i \text{ et } x_i \models \varphi\}.$$

Problème de maximisation de l'utilité collective

Quelle est la complexité du problème de maximisation de l'utilité collective ?

Problème [MAX-CUF]

Étant donnés une instance du problème de partage défini comme précédemment et un entier K ($\mathscr{V} = \mathbb{N}$), existe-t-il un partage admissible $\overrightarrow{\pi}$ tel que $uc(\overrightarrow{\pi}) \geq K$?

Ce problème est **NP**-complet.

Le reste-t-il dans les cas suivants :

- restrictions sur les opérateurs ($\oplus \in \{+, \max\}$, $g \in \{+, \min, \text{leximin}\}$),
- restriction sur les contraintes (préemption, volume, exclusion),
- restriction sur les préférences (atomiques) ?

Problème de maximisation de l'utilité collective

Quelle est la complexité du problème de maximisation de l'utilité collective ?

Problème [MAX-CUF]

Étant donnés une instance du problème de partage défini comme précédemment et un entier K ($\mathscr{V} = \mathbb{N}$), existe-t-il un partage admissible $\overrightarrow{\pi}$ tel que $uc(\overrightarrow{\pi}) \geq K$?

Ce problème est **NP**-complet.

Le reste-t-il dans les cas suivants :

- restrictions sur les opérateurs ($\oplus \in \{+, \max\}$, $g \in \{+, \min, \text{leximin}\}$),
- restriction sur les contraintes (préemption, volume, exclusion),
- restriction sur les préférences (atomiques) ?

Les résultats de complexité obtenus

[MAX-CUF]

N'importe quel type de contraintes : **NPC**

Pas de contrainte : **P**

Contraintes d'exclusion seulement

$\oplus$ \ g	$+$	(lexi)min
$+$	**NPC**	**NPC**
max	**NPC**	**NPC**

Contraintes de volume seulement

$\oplus$ \ g	$+$	(lexi)min
$+$	**NPC**	**NPC**
max	**NPC**	**NPC**

Contraintes de préemption

Demandes atomiques

$\oplus$ \ g	$+$	min	leximin
$+$	**P**	**NPC**, **P** si pds ég.	**NPC**
max	**P**	**P**	?

N'importe quel type de demande

$\oplus$ \ g	$+$	(lexi)min
$+$	**NPC**	**NPC**
max	**NPC**	**NPC**

Partie 2 : Représentation compacte et complexité

Absence d'envie

Une autre manière de traduire la notion d'équité est de rechercher des partages **sans envie**.

L'absence d'envie seule ne suffit pas : il faut un critère d'efficacité (**Pareto-efficacité**, complétude, maximisation d'une CUF, ...).

Mais... Il n'existe pas toujours un partage efficace et sans envie, et déterminer s'il en existe un peut être une tâche complexe.

*Quelle est la complexité du problème d'existence d'un partage efficace et sans envie, lorsque les préférences des agents sont exprimées de manière **compacte** (ex : logique), et avec contrainte de préemption uniquement ?*

Absence d'envie

Une autre manière de traduire la notion d'équité est de rechercher des partages **sans envie**.

L'absence d'envie seule ne suffit pas : il faut un critère d'efficacité (**Pareto-efficacité**, complétude, maximisation d'une CUF, ...).

Mais... Il n'existe pas toujours un partage efficace et sans envie, et déterminer s'il en existe un peut être une tâche complexe.

Quelle est la complexité du problème d'existence d'un partage efficace et sans envie, lorsque les préférences des agents sont exprimées de manière ***compacte*** *(ex : logique), et avec contrainte de préemption uniquement ?*

Absence d'envie

Une autre manière de traduire la notion d'équité est de rechercher des partages **sans envie**.

L'absence d'envie seule ne suffit pas : il faut un critère d'efficacité (**Pareto-efficacité**, complétude, maximisation d'une CUF, ...).

Mais... Il n'existe pas toujours un partage efficace et sans envie, et déterminer s'il en existe un peut être une tâche complexe.

Quelle est la complexité du problème d'existence d'un partage efficace et sans envie, lorsque les préférences des agents sont exprimées de manière ***compacte*** *(ex : logique), et avec contrainte de préemption uniquement ?*

Absence d'envie

Une autre manière de traduire la notion d'équité est de rechercher des partages **sans envie**.

L'absence d'envie seule ne suffit pas : il faut un critère d'efficacité (**Pareto-efficacité**, complétude, maximisation d'une CUF, ...).

Mais... Il n'existe pas toujours un partage efficace et sans envie, et déterminer s'il en existe un peut être une tâche complexe.

Quelle est la complexité du problème d'existence d'un partage efficace et sans envie, lorsque les préférences des agents sont exprimées de manière ***compacte*** *(ex : logique), et avec contrainte de préemption uniquement ?*

À propos de préférences dichotomiques. . .

Nous allons nous intéresser au cas très particulier pour lequel les préférences des agents sont représentées sous forme **dichotomique**.

Relation de préférence dichotomique

$\succeq$ est dichotomique $\Leftrightarrow$ il existe un ensemble de «bons» lots *Good* tel que $\pi \succeq \pi' \Leftrightarrow \pi \in Good$ ou $\pi' \notin Good$.

Exemple :

$$\mathscr{O} = \{o_1, o_2, o_3\}$$

$$\Rightarrow 2^{\mathscr{O}} = \{\varnothing, \{o_1\}, \{o_2\}, \{o_3\}, \{o_1, o_2\}, \{o_1, o_3\}, \{o_2, o_3\}, \{o_1, o_2, o_3\}\}$$

$$Good \longrightarrow \{\{o_1, o_2\}, \{o_2, o_3\}\}$$

$$\overline{Good} \longrightarrow \{\varnothing, \{o_1\}, \{o_2\}, \{o_3\}, \{o_1, o_3\}, \{o_1, o_2, o_3\}\}$$

Encore un langage logique. . .

Une relation de préférence dichotomique est représentée par son ensemble *Good*. Un moyen évident de représenter cet ensemble est d'utiliser la logique propositionnelle.

Exemple :

		
$Good_i$	$\{\{o_1, o_2\}, \{o_2, o_3\}\}$	$\{\{o_2\}\{o_2, o_3\}\}$
φ_i	$(o_1 \wedge o_2 \wedge \neg o_3) \vee (\neg o_1 \wedge o_2 \wedge o_3)$	$o_2 \wedge \neg o_1$

Préemption, absence d'envie et Pareto-efficacité

- La contrainte de **préemption** s'exprime comme une formule logique de $L^{alloc}_{\mathscr{O}} \rightsquigarrow \Gamma_{\mathcal{P}}$.
- La propriété d'**absence d'envie** s'exprime aussi comme une formule de $L^{alloc}_{\mathscr{O}} \rightsquigarrow \Lambda_{\mathcal{P}}$.
- La propriété de **Pareto-efficacité** équivaut à :
  - la satisfaction d'un nombre maximal (au sens de l'inclusion) d'agents,
  - la consistance de $F(\overrightarrow{\pi})$ avec un sous-ensemble maximal consistant de formules de $\{\varphi_1^*, \ldots, \varphi_n^*\}$.

Existence d'un partage sans envie et Pareto-efficace

$\exists \mathscr{S}$ sous-ensemble maximal $\Gamma_{\mathcal{P}}$-consistant de $\{\varphi_1^*, \ldots, \varphi_n^*\}$ tel que $\bigwedge_{\varphi \in \mathscr{S}} \varphi \wedge \Gamma_{\mathcal{P}} \wedge \Lambda_{\mathcal{P}}$ est consistant.

Préemption, absence d'envie et Pareto-efficacité

- La contrainte de **préemption** s'exprime comme une formule logique de $L^{alloc}_{\mathscr{O}} \leadsto \Gamma_{\mathcal{P}}$.
- La propriété d'**absence d'envie** s'exprime aussi comme une formule de $L^{alloc}_{\mathscr{O}} \leadsto \Lambda_{\mathcal{P}}$.
- La propriété de **Pareto-efficacité** équivaut à :
  - la satisfaction d'un nombre maximal (au sens de l'inclusion) d'agents,
  - la consistance de $F(\overrightarrow{\pi})$ avec un sous-ensemble maximal consistant de formules de $\{\varphi^*_1, \dots, \varphi^*_n\}$.

Existence d'un partage sans envie et Pareto-efficace

$\exists \mathscr{S}$ sous-ensemble maximal $\Gamma_{\mathcal{P}}$-consistant de $\{\varphi^*_1, \dots, \varphi^*_n\}$ tel que $\bigwedge_{\varphi \in \mathscr{S}} \varphi \wedge \Gamma_{\mathcal{P}} \wedge \Lambda_{\mathcal{P}}$ est consistant.

Préemption, absence d'envie et Pareto-efficacité

- La contrainte de **préemption** s'exprime comme une formule logique de $L_{\mathscr{O}}^{alloc} \leadsto \Gamma_{\mathcal{P}}$.
- La propriété d'**absence d'envie** s'exprime aussi comme une formule de $L_{\mathscr{O}}^{alloc} \leadsto \Lambda_{\mathcal{P}}$.
- La propriété de **Pareto-efficacité** équivaut à :
  - la satisfaction d'un nombre maximal (au sens de l'inclusion) d'agents,
  - la consistance de $F(\overrightarrow{\pi})$ avec un sous-ensemble maximal consistant de formules de $\{\varphi_1^*, \ldots, \varphi_n^*\}$.

Existence d'un partage sans envie et Pareto-efficace

$\exists \mathscr{S}$ sous-ensemble maximal $\Gamma_{\mathcal{P}}$-consistant de $\{\varphi_1^*, \ldots, \varphi_n^*\}$ tel que $\bigwedge_{\varphi \in \mathscr{S}} \varphi \wedge \Gamma_{\mathcal{P}} \wedge \Lambda_{\mathcal{P}}$ est consistant.

Préemption, absence d'envie et Pareto-efficacité

- La contrainte de **préemption** s'exprime comme une formule logique de $L_{\mathscr{O}}^{alloc} \leadsto \Gamma_{\mathcal{P}}$.
- La propriété d'**absence d'envie** s'exprime aussi comme une formule de $L_{\mathscr{O}}^{alloc} \leadsto \Lambda_{\mathcal{P}}$.
- La propriété de **Pareto-efficacité** équivaut à :
  - la satisfaction d'un nombre maximal (au sens de l'inclusion) d'agents,
  - la consistance de $F(\overrightarrow{\pi})$ avec un sous-ensemble maximal consistant de formules de $\{\varphi_1^*, \dots, \varphi_n^*\}$.

Existence d'un partage sans envie et Pareto-efficace

$\exists \mathscr{S}$ sous-ensemble maximal $\Gamma_{\mathcal{P}}$-consistant de $\{\varphi_1^*, \dots, \varphi_n^*\}$ tel que $\bigwedge_{\varphi \in \mathscr{S}} \varphi \wedge \Gamma_{\mathcal{P}} \wedge \Lambda_{\mathcal{P}}$ est consistant.

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* [**?**].

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

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

Le problème [EEF EXISTENCE], préférences dichotomiques

Proposition

Le problème [EEF EXISTENCE] pour des agents ayant des préférences dichotomiques monotones sous forme logique est Σ_2^p-complet ($\Sigma_2^p = \mathbf{NP}^{\mathbf{NP}}$).

Ce résultat subsiste si les préférences ne sont plus monotones.

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

Le problème [EEF EXISTENCE], préférences dichotomiques

Proposition

Le problème [EEF EXISTENCE] pour des agents ayant des préférences dichotomiques monotones sous forme logique est Σ_2^p-complet ($\Sigma_2^p = \mathbf{NP}^{\mathbf{NP}}$).

Ce résultat subsiste si les préférences ne sont plus monotones.

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

Pour des préférences non dichotomiques ?

Corollaire (du résultat pour les préférences dichotomiques)

Le problème [EEF EXISTENCE] pour des agents ayant des préférences monotones exprimées dans un langage compact sous forme logique $\mathcal{L}$ est Σ_2^p-complet.

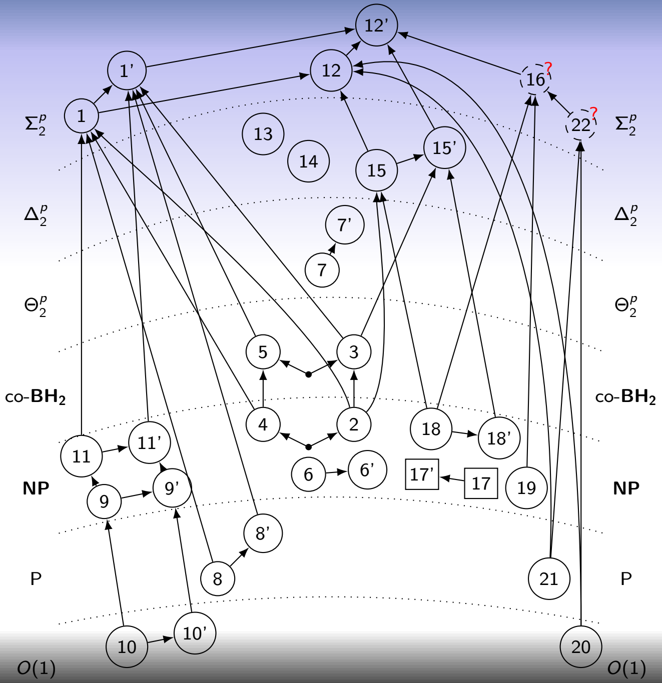
- $\mathcal{L}$ est aussi compact que le langage précédent pour les préférences dichotomiques ;
- on peut comparer deux alternatives en temps polynomial.

Logique pondérée et préférences additives

- **Logique pondérée :** critères d'efficacité alternatifs fondés sur la maximisation de l'utilité collective.
- **Préférences additives :**
  - **Complétude :** résultat déjà connu [?].
  - **Pareto-efficacité :** ? ? ?
    - préférences identiques,
    - préférences 0–1,
    - préférences 0–1–. . .–k (? ? ?),
    - nombre d'objets inférieur au nombre d'agents.

Logique pondérée et préférences additives

- **Logique pondérée :** critères d'efficacité alternatifs fondés sur la maximisation de l'utilité collective.
- **Préférences additives :**
  - **Complétude :** résultat déjà connu [**?**].
  - **Pareto-efficacité :** ? ? ?
    - préférences identiques,
    - préférences 0–1,
    - préférences 0–1–. . .–k (? ? ?),
    - nombre d'objets inférieur au nombre d'agents.



Σ_2^p
Δ_2^p
Θ_2^p
co-$\mathbf{BH_2}$
NP
P
$O(1)$
1'
12'
12
1
16
?
22
?
13
14
15
15'
7'
7
5
3
4
2
18
18'
11
11'
9
9'
6
6'
17'
17
19
8'
8
21
10
10'
20

Partie 3

Algorithmique et expérimentations

Partie 3 : Algorithmique et expérimentations

Maximisation de l'utilité collective ?

Nous avons introduit un langage logique dédié au problème de maximisation de l'utilité collective.

Mais... Comment résoudre ce problème et calculer un partage optimal ?

Utiliser un cadre de modélisation et résolution tel que les **problèmes de satisfaction de contraintes** (CSP) et la **programmation par contraintes**.

Maximisation de l'utilité collective ?

Nous avons introduit un langage logique dédié au problème de maximisation de l'utilité collective.

Mais... Comment résoudre ce problème et calculer un partage optimal ?

Utiliser un cadre de modélisation et résolution tel que les **problèmes de satisfaction de contraintes** (CSP) et la **programmation par contraintes**.

Maximisation de l'utilité collective ?

Nous avons introduit un langage logique dédié au problème de maximisation de l'utilité collective.

Mais... Comment résoudre ce problème et calculer un partage optimal ?

Utiliser un cadre de modélisation et résolution tel que les **problèmes de satisfaction de contraintes** (CSP) et la **programmation par contraintes**.

Réseaux de contraintes

Réseau de contraintes [?]

Un réseau de contraintes est formé par :

- un ensemble de variables $\mathscr{X} = \{\mathbf{x_1}, \ldots, \mathbf{x_p}\}$;
- un ensemble de domaines $\mathscr{D} = \{\mathscr{D}_{\mathbf{x_1}}, \ldots, \mathscr{D}_{\mathbf{x_p}}\}$;
- un ensemble de contraintes $\mathscr{C}$, avec, pour tout $C \in \mathscr{C}$:
  - $\mathscr{X}(C)$ le scope de la contrainte,
  - $\mathscr{R}(C)$ l'ensemble des tuples autorisés par la contrainte.

Le problème de satisfaction de contraintes

CSP classique

Étant donné : un réseau de contraintes $(\mathscr{X}, \mathscr{D}, \mathscr{C})$.
Existe-t-il une instanciation complète cohérente v de $(\mathscr{X}, \mathscr{D}, \mathscr{C})$ *?*

$\rightsquigarrow$ **NP**-complet.

CSP avec variable objectif

Étant donnés : un réseau de contraintes $(\mathscr{X}, \mathscr{D}, \mathscr{C})$ et une variable objectif $\mathbf{o} \in \mathscr{X}$, telle que $\mathscr{D}_\mathbf{o} \subset \mathbb{N}$.
Quelle est la valeur maximale α de $\mathscr{D}_\mathbf{o}$ telle qu'il existe une instanciation complète cohérente $\widehat{v}$ avec $\widehat{v}(\mathbf{o}) = \alpha$?

$\rightsquigarrow$ **NP**-complet (version problème de décision).

Le problème de satisfaction de contraintes

CSP classique

Étant donné : un réseau de contraintes $(\mathscr{X}, \mathscr{D}, \mathscr{C})$.
Existe-t-il une instanciation complète cohérente v de $(\mathscr{X}, \mathscr{D}, \mathscr{C})$ *?*

$\rightsquigarrow$ **NP**-complet.

CSP avec variable objectif

Étant donnés : un réseau de contraintes $(\mathscr{X}, \mathscr{D}, \mathscr{C})$ et une variable objectif $\mathbf{o} \in \mathscr{X}$, telle que $\mathscr{D}_{\mathbf{o}} \subset \mathbb{N}$.
Quelle est la valeur maximale α *de* $\mathscr{D}_{\mathbf{o}}$ *telle qu'il existe une instanciation complète cohérente* $\widehat{v}$ *avec* $\widehat{v}(\mathbf{o}) = \alpha$ *?*

$\rightsquigarrow$ **NP**-complet (version problème de décision).

Partage et CSP

Le problème de maximisation de l'utilité collective peut s'exprimer et se résoudre aisément à l'aide d'un CSP avec variable objectif.

Mais… Ce n'est pas le cas pour l'ordre social leximin, qui ne peut s'exprimer directement comme un problème de maximisation mono-objectif.

Optimisation leximin : Problème d'importance dont la portée dépasse largement le cadre du partage (CSP flous, recherche opérationnelle, …)

Partage et CSP

Le problème de maximisation de l'utilité collective peut s'exprimer et se résoudre aisément à l'aide d'un CSP avec variable objectif.
Mais... Ce n'est pas le cas pour l'ordre social leximin, qui ne peut s'exprimer directement comme un problème de maximisation mono-objectif.

Optimisation leximin : Problème d'importance dont la portée dépasse largement le cadre du partage (CSP flous, recherche opérationnelle, ...)

Partage et CSP

Le problème de maximisation de l'utilité collective peut s'exprimer et se résoudre aisément à l'aide d'un CSP avec variable objectif.
Mais... Ce n'est pas le cas pour l'ordre social leximin, qui ne peut s'exprimer directement comme un problème de maximisation mono-objectif.

Optimisation leximin : Problème d'importance dont la portée dépasse largement le cadre du partage (CSP flous, recherche opérationnelle, ...)

Le problème de satisfaction de contraintes

CSP classique

Étant donné : un réseau de contraintes $(\mathscr{X}, \mathscr{D}, \mathscr{C})$.
Existe-t-il une instanciation complète cohérente v de $(\mathscr{X}, \mathscr{D}, \mathscr{C})$ *?*

$\rightsquigarrow$ **NP**-complet.

CSP avec variable objectif

Étant donnés : un réseau de contraintes $(\mathscr{X}, \mathscr{D}, \mathscr{C})$ et une variable objectif $\mathbf{o} \in \mathscr{X}$, telle que $\mathscr{D}_{\mathbf{o}} \subset \mathbb{N}$.
Quelle est la valeur maximale α *de* $\mathscr{D}_{\mathbf{o}}$ *telle qu'il existe une instanciation complète cohérente* $\widehat{v}$ *avec* $\widehat{v}(\mathbf{o}) = \alpha$ *?*

$\rightsquigarrow$ **NP**-complet (version problème de décision).

Leximin-CSP (CSP multi-objectif)

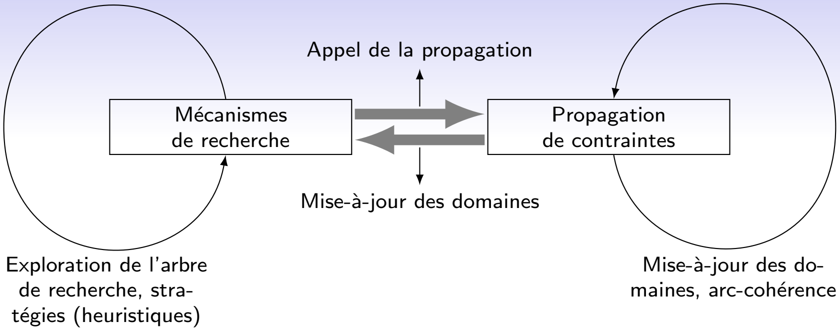
Étant donnés : Un réseau de contraintes $(\mathscr{X}, \mathscr{D}, \mathscr{C})$ et un vecteur de variables $\overrightarrow{u} = \langle \mathbf{u_1}, \ldots, \mathbf{u_n} \rangle$ $(\forall i,\ \mathbf{u_i} \in \mathscr{X}$ and $\mathscr{D}_{\mathbf{u_i}} \in \mathbb{N})$ appelé **vecteur objectif**.
Quel est le vecteur leximin-optimal $\langle \alpha_1, \ldots, \alpha_n \rangle$ *de* $\langle \mathscr{D}_{\mathbf{u_1}}, \ldots, \mathscr{D}_{\mathbf{u_n}} \rangle$ *tel qu'il existe une instanciation complète cohérente* $\widehat{v}$ *avec* $\widehat{v}(\mathbf{u_i}) = \alpha_i$ *pour tout i ?*

La programmation par contraintes

Un outil flexible et efficace pour l'implantation et la résolution de CSP.

- **Notre approche :** utiliser cet outil comme une «boîte noire» pour la résolution de leximin-CSP.
- **Intérêt de l'utilisation de la Programmation par Contraintes :**
  - Développer des algorithmes génériques.
  - Bénéficier de l'utilisation de ce cadre puissant et de son algorithmique.

La programmation par contraintes



Ce sur quoi nous pouvons agir :

- Initialiser le problème (déclarer les variables, domaines, contraintes).
- Implanter de nouveaux algorithmes de propagation.
- Appeler les fonctions **solve** ou **maximize** (boîtes noires).

Algorithme 1

Un algorithme de type *branch-and-bound*

Un algorithme de type *branch-and-bound*

L'algorithme *branch-and-bound* classique (à critère entier) :

- Un algorithme de branchement (exploration de l'arbre de recherche).
- Une borne inférieure sur le critère à maximiser.
- Une borne supérieure et un mécanisme de coupe ($ub \leq lb$).

Notre algorithme (critère vectoriel avec préordre leximin) :

- L'algorithme de branchement est donné par le solveur de contraintes (appel à **solve**).
- Borne inférieure : le vecteur objectif de la dernière solution trouvée.
- Un mécanisme de coupe donné par une procédure de filtrage associée au préordre leximin $\rightsquigarrow$ **Contrainte Leximin** (inspirée de **Multiset Ordering**).

Algorithme 2

Trier pour régner

Version triée du vecteur objectif

Idée initiale

Maximiser le vecteur objectif en utilisant le préordre leximin $\Leftrightarrow$ maximiser les composantes successives du vecteur objectif **trié**.

$\rightsquigarrow$ Nous introduisons donc la version triée du vecteur objectif :

- **Un vecteur de variables** $(\mathbf{y_1}, \ldots, \mathbf{y_n})$.
- **Une contrainte** $\mathbf{Sort}(\overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{y}})$ [?] (filtrage en temps $O(n \log(n))$).

Version triée du vecteur objectif

Idée initiale

Maximiser le vecteur objectif en utilisant le préordre leximin $\Leftrightarrow$ maximiser les composantes successives du vecteur objectif **trié**.

$\rightsquigarrow$ Nous introduisons donc la version triée du vecteur objectif :

- **Un vecteur de variables** $(\mathbf{y_1}, \ldots, \mathbf{y_n})$.
- **Une contrainte** $\mathsf{Sort}(\overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{y}})$ [?] (filtrage en temps $O(n \log(n))$).

1. Maximiser $\mathbf{y_1}$: $\widehat{y_1}$.
2. Maximiser $\mathbf{y_2}$ sous la contrainte $\mathbf{y_1} = \widehat{y_1}$: $\widehat{y_2}$.

$\vdots$

n. Maximiser $\mathbf{y_n}$ sous les contraintes $\mathbf{y_1} = \widehat{y_1}$, …, $\mathbf{y_{n-1}} = \widehat{y_{n-1}}$.

Version triée du vecteur objectif

Idée initiale

Maximiser le vecteur objectif en utilisant le préordre leximin $\Leftrightarrow$ maximiser les composantes successives du vecteur objectif **trié**.

$\rightsquigarrow$ Nous introduisons donc la version triée du vecteur objectif :

- **Un vecteur de variables** $(\mathbf{y_1}, \ldots, \mathbf{y_n})$.
- **Une contrainte Sort**$(\overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{y}})$ [?] (filtrage en temps $O(n \log(n))$).

1. Maximiser $\mathbf{y_1}$: $\widehat{y_1}$.
2. Maximiser $\mathbf{y_2}$ sous la contrainte $\mathbf{y_1} = \widehat{y_1}$: $\widehat{y_2}$.

$\vdots$

n. Maximiser $\mathbf{y_n}$ sous les contraintes $\mathbf{y_1} = \widehat{y_1}$, …, $\mathbf{y_{n-1}} = \widehat{y_{n-1}}$.

Version triée du vecteur objectif

Idée initiale

Maximiser le vecteur objectif en utilisant le préordre leximin $\Leftrightarrow$ maximiser les composantes successives du vecteur objectif **trié**.

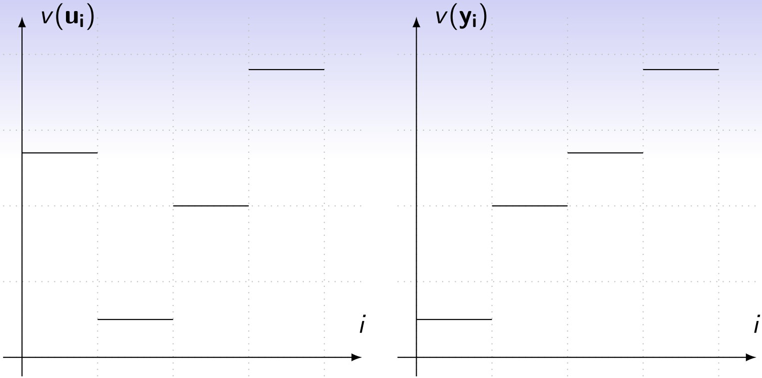
$\rightsquigarrow$ Nous introduisons donc la version triée du vecteur objectif :

- **Un vecteur de variables** $(\mathbf{y_1}, \ldots, \mathbf{y_n})$.
- **Une contrainte Sort**$(\overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{y}})$ [?] (filtrage en temps $O(n \log(n))$).

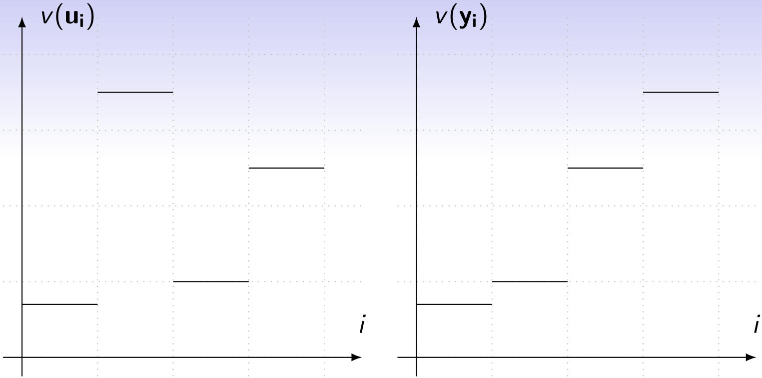
1. Maximiser $\mathbf{y_1}$: $\widehat{y_1}$.
2. Maximiser $\mathbf{y_2}$ sous la contrainte $\mathbf{y_1} = \widehat{y_1}$: $\widehat{y_2}$.

$$\vdots$$

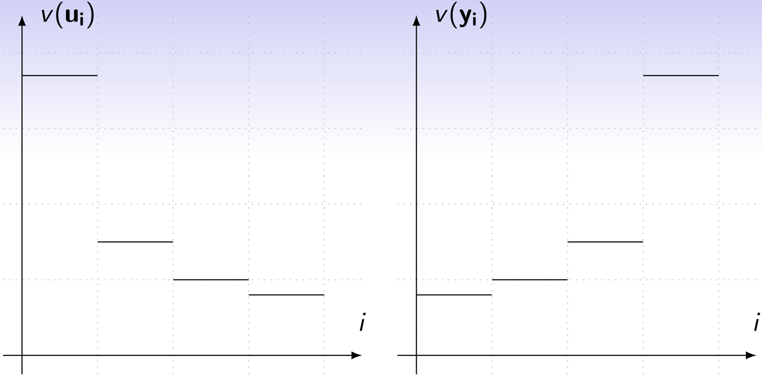
n. Maximiser $\mathbf{y_n}$ sous les contraintes $\mathbf{y_1} = \widehat{y_1}$, . . ., $\mathbf{y_{n-1}} = \widehat{y_{n-1}}$.



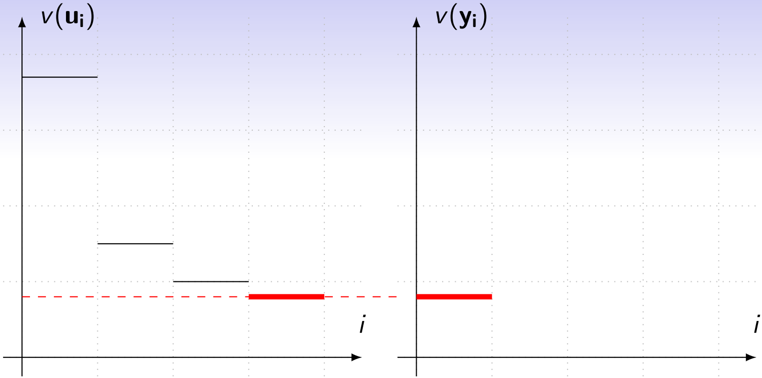
$v(\mathbf{u_i})$
i
$v(\mathbf{y_i})$
i

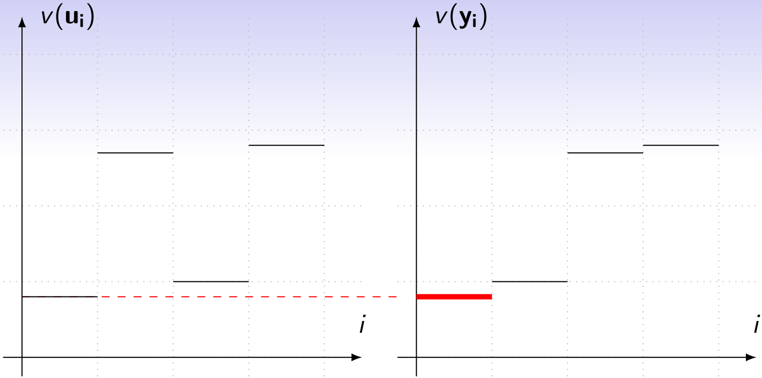


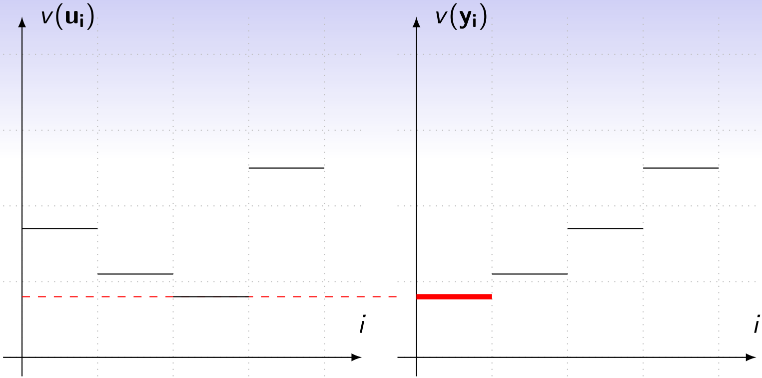
$v(\mathbf{u_i})$
i
$v(\mathbf{y_i})$
i



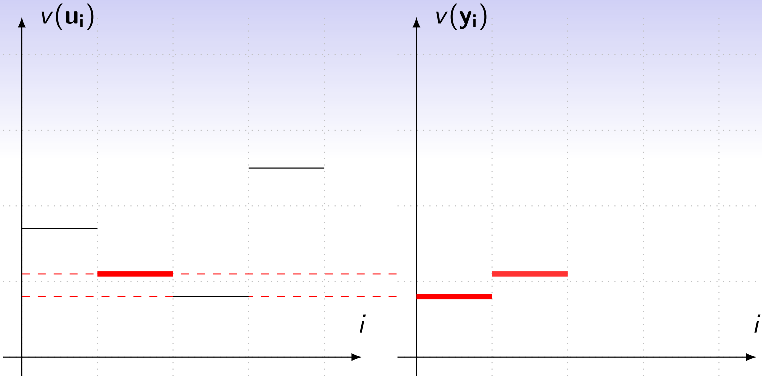
$v(\mathbf{u_i})$
i
$v(\mathbf{y_i})$
i


$v(\mathbf{u_i})$
i
$v(\mathbf{y_i})$
i

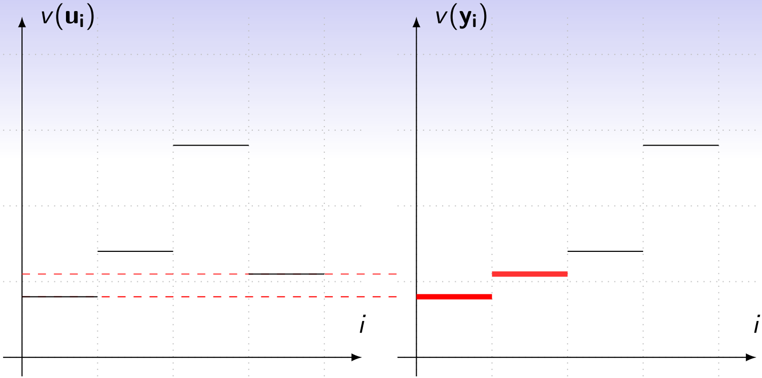

$v(\mathbf{u_i})$
i
$v(\mathbf{y_i})$
i



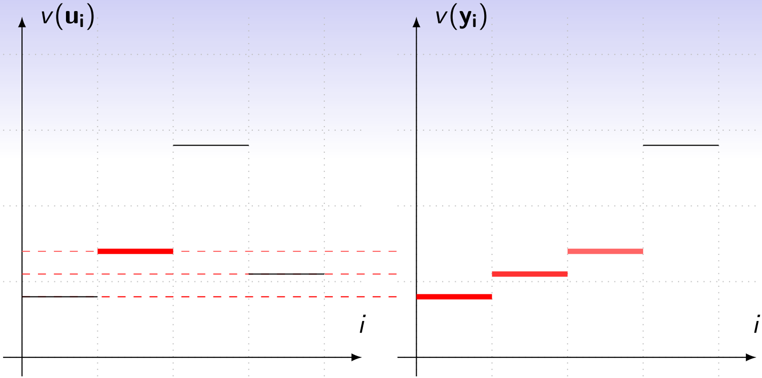
$v(\mathbf{u_i})$
i
$v(\mathbf{y_i})$
i



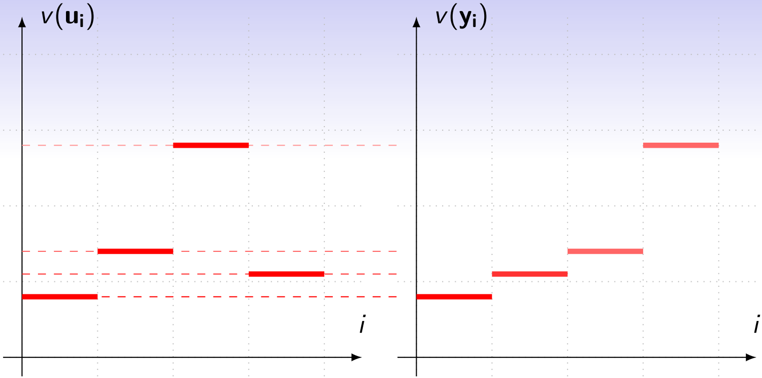
$v(\mathbf{u_i})$
i
$v(\mathbf{y_i})$
i



$v(\mathbf{u_i})$
i
$v(\mathbf{y_i})$
i



$v(\mathbf{u_i})$
i
$v(\mathbf{y_i})$
i



$v(\mathbf{u_i})$
i
$v(\mathbf{y_i})$
i

Algorithme 3

Utiliser une méta-contrainte. . .

Une définition alternative du tri...

Proposition

$\langle y_1, \ldots, y_n \rangle$ est la version triée de $\langle u_1, \ldots, u_n \rangle$ si et seulement si :

- y_1 est la valeur maximale telle que tous les u_i sont $\geq y_1$;
- y_2 est la valeur maximale telle qu'au moins $n-1$ valeurs parmi les u_i sont $\geq y_2$;

$$\vdots$$

- y_n est la valeur maximale telle qu'au moins une valeur parmi les u_i est $\geq y_n$.

Une définition alternative du tri...

Proposition

$\langle y_1, \ldots, y_n \rangle$ est la version triée de $\langle u_1, \ldots, u_n \rangle$ si et seulement si :

- y_1 est la valeur maximale telle que tous les u_i sont $\geq y_1$;
- y_2 est la valeur maximale telle qu'au moins $n - 1$ valeurs parmi les u_i sont $\geq y_2$;

$$\vdots$$

- y_n est la valeur maximale telle qu'au moins une valeur parmi les u_i est $\geq y_n$.

Une définition alternative du tri...

Proposition

$\langle y_1, \ldots, y_n \rangle$ est la version triée de $\langle u_1, \ldots, u_n \rangle$ si et seulement si :

- y_1 est la valeur maximale telle que tous les u_i sont $\geq y_1$;
- y_2 est la valeur maximale telle qu'au moins $n - 1$ valeurs parmi les u_i sont $\geq y_2$;

$$\vdots$$

- y_n est la valeur maximale telle qu'au moins une valeur parmi les u_i est $\geq y_n$.

La méta-contrainte AtLeast

«y_i est la valeur maximale telle qu'au moins $n - i + 1$ valeurs parmi les u_i sont supérieures ou égales à y_i»

$\rightsquigarrow$ une méta-contrainte de cardinalité particulière [**?**] :

$$\textbf{AtLeast}(\{\mathbf{y_i} \geq u_1, \ldots, \mathbf{y_i} \geq u_n\}, n - i + 1)$$

- Un algorithme de filtrage spécifique qui s'exécute en $O(n)$.
- Une implantation possible en utilisant des contraintes linéaires.

Algorithme 4

Remplacement des variables objectif

Remplacements de variables

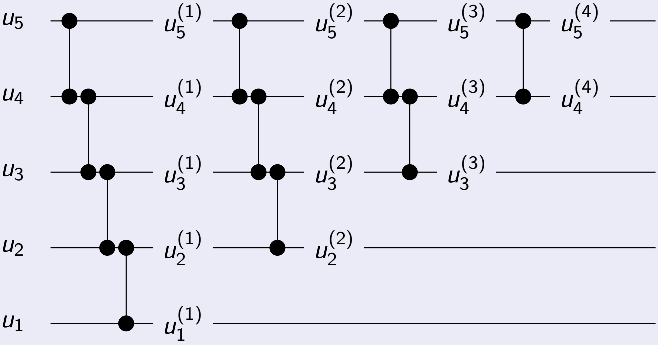
L'algorithme de [?] est fondé sur la remarque suivante :

Proposition

Étant donné un leximin-CSP $(\mathscr{X}, \mathscr{D}, \mathscr{C}, \overrightarrow{\mathbf{u}})$, l'ensemble des solutions leximin-optimales ne change pas lorsque deux des variables objectif $\mathbf{u_i}$ et $\mathbf{u_j}$ sont remplacées par $max(\mathbf{u_i}, \mathbf{u_j})$ et $min(\mathbf{u_i}, \mathbf{u_j})$.

$\leadsto$ Idée de l'algorithme : utiliser récursivement ces transformations pour faire apparaître $min_i(\mathbf{u_i})$ de manière explicite.

Illustration de l'algorithme



Algorithme 5

Des ensembles saturés

Principe de l'algorithme

- Algorithme issu de la littérature sur les CSP flous [**?**].
- Fondé sur le calcul de sous-ensembles **saturés** de variables objectifs (*strong* α-*cuts*).
- À chaque étape de l'algorithme, on opère un branchement sur les sous-ensembles saturés.
- **Inconvénient :** il peut y avoir un nombre exponentiel de sous-ensembles à explorer à chaque étape (sauf dans des cas particuliers : problèmes linéaires continus avec ensemble des alternatives convexe).

Partie 3 : Algorithmique et expérimentations

Des jeux de tests pour les algorithmes...

Pas de jeu de tests générique et réaliste de référence pour les problèmes d'allocation de biens indivisibles (sauf enchères combinatoires).

Nous avons testé nos algorithmes sur :

- une version simplifiée (linéaire) du problème **Pléiades** ;
- des instances de problèmes d'enchères combinatoires générées par CATS ;
- des instances du modèle logique du problème de maximisation de l'utilité collective générées de manière aléatoire ;
- une instance du problème d'allocation de sujets de TREX (Supaéro).

Des jeux de tests pour les algorithmes...

Pas de jeu de tests générique et réaliste de référence pour les problèmes d'allocation de biens indivisibles (sauf enchères combinatoires).

Nous avons testé nos algorithmes sur :

- une version simplifiée (linéaire) du problème **Pléiades** ;
- des instances de problèmes d'enchères combinatoires générées par CATS ;
- des instances du modèle logique du problème de maximisation de l'utilité collective générées de manière aléatoire ;
- une instance du problème d'allocation de sujets de TREX (Supaéro).

Le problème Pléiades simplifié

Un modèle très simplifié du problème de partage de la constellation Pléiades...

Hypothèses simplificatrices :

- **Préférences :** les préférences des agents sont additives.
- **Contraintes :**
  - contraintes de volume particulières à chaque agent (pour introduire un quota d'images par agent),
  - contraintes de volume générales (pour approximer les contraintes physiques réelles)
- Modélisation à l'aide de variables de décision 0–1 et de contraintes linéaires uniquement.

Générateur d'instances :

Nous avons implanté un générateur d'instances (en Java), disponible en ligne :

▸ http ://www.cert.fr/dcsd/THESES/sbouveret/benchmark

Enchères combinatoires

Enchères combinatoires : instance particulière du problème de partage de biens indivisibles :

- un ensemble d'**enchérisseurs** (agents) et un ensemble d'**objets** ;
- chaque agent propose des **mises** : une mise = un ensemble d'objets (un **lot**) associé à un prix (une **utilité**) ;
- un partage valide est une sélection d'un ensemble de mises mutuellement exclusives ;
- dans la formulation traditionnelle, on cherche à maximiser le revenu du commissaire-priseur (vision **utilitariste classique**).

Un générateur d'instances aléatoires réalistes de référence :

CATS (*Combinatorial Auctions Test Suite*) ▸ http ://www.cs.ubc.ca/ kevinlb/CATS

$\rightsquigarrow$ Génération d'instances pour le problème d'optimisation leximin.

Modèle logique

- Une implantation (en Java) du **modèle logique** du problème de partage [MAX-CUF].
- **Trois générateurs de problèmes :**
  - générique,
  - Spot,
  - Pléiades.

Modèle et générateur d'instances disponibles en ligne :

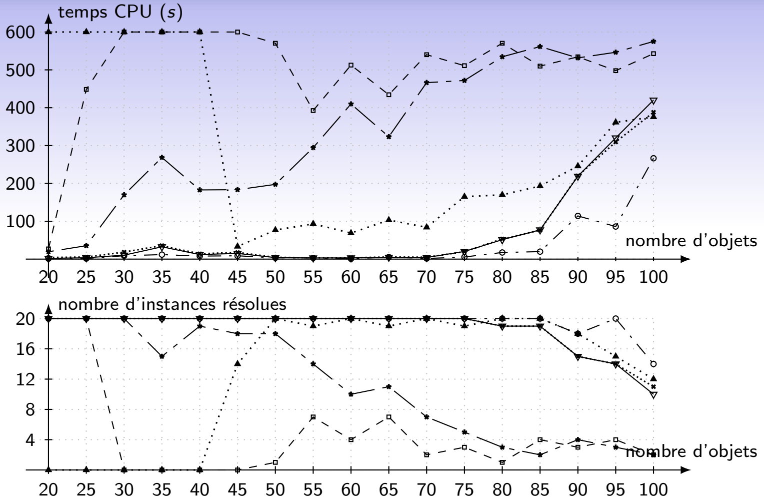
http ://www.cert.fr/dcsd/THESES/sbouveret/benchmark2007

Partie 3 : Algorithmique et expérimentations

Le cadre expérimental...

- Tous les algorithmes ont été implantés en **Java**, avec la bibliothèque de programmation par contraintes **Choco** [?].
- Nous avons cherché à mettre en valeur l'influence du nombre d'objets et du nombre d'agents sur le temps de calcul.

Problème Pléiades linéaire (20 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 [**?**].

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

Conclusion

Synthèse des contributions

1. **Modélisation des problèmes de partage :** un formalisme pour intégrer des droits exogènes inégaux dans le cadre *welfariste*.
2. **Représentation compacte des problèmes modélisés :**
   - Représentation du problème de maximisation de l'utilité collective fondé sur la logique pondérée.
   - Représentation logique du problème d'existence d'un partage sans envie et Pareto-efficace.
3. **Complexité computationnelle :** Identification de la complexité des problèmes [MAX-CUF] et [EEF EXISTENCE], et de certaines de leurs restrictions et dérivés.
4. **Algorithmique :** Adaptation et développement d'algorithmes de programmation par contraintes pour l'optimisation leximin.
5. **Expérimentations :**
   - Génération d'instances réalistes de problèmes de partage.
   - Comparaison expérimentale des algorithmes d'optimisation leximin.

Synthèse des contributions

1. **Modélisation des problèmes de partage :** un formalisme pour intégrer des droits exogènes inégaux dans le cadre *welfariste*.
2. **Représentation compacte des problèmes modélisés :**
   - Représentation du problème de maximisation de l'utilité collective fondé sur la logique pondérée.
   - Représentation logique du problème d'existence d'un partage sans envie et Pareto-efficace.
3. **Complexité computationnelle :** Identification de la complexité des problèmes [MAX-CUF] et [EEF EXISTENCE], et de certaines de leurs restrictions et dérivés.
4. **Algorithmique :** Adaptation et développement d'algorithmes de programmation par contraintes pour l'optimisation leximin.
5. **Expérimentations :**
   - Génération d'instances réalistes de problèmes de partage.
   - Comparaison expérimentale des algorithmes d'optimisation leximin.

Synthèse des contributions

1. **Modélisation des problèmes de partage :** un formalisme pour intégrer des droits exogènes inégaux dans le cadre *welfariste*.
2. **Représentation compacte des problèmes modélisés :**
   - Représentation du problème de maximisation de l'utilité collective fondé sur la logique pondérée.
   - Représentation logique du problème d'existence d'un partage sans envie et Pareto-efficace.
3. **Complexité computationnelle :** Identification de la complexité des problèmes [MAX-CUF] et [EEF EXISTENCE], et de certaines de leurs restrictions et dérivés.
4. **Algorithmique :** Adaptation et développement d'algorithmes de programmation par contraintes pour l'optimisation leximin.
5. **Expérimentations :**
   - Génération d'instances réalistes de problèmes de partage.
   - Comparaison expérimentale des algorithmes d'optimisation leximin.

Synthèse des contributions

1. **Modélisation des problèmes de partage :** un formalisme pour intégrer des droits exogènes inégaux dans le cadre *welfariste*.
2. **Représentation compacte des problèmes modélisés :**
   - Représentation du problème de maximisation de l'utilité collective fondé sur la logique pondérée.
   - Représentation logique du problème d'existence d'un partage sans envie et Pareto-efficace.
3. **Complexité computationnelle :** Identification de la complexité des problèmes [MAX-CUF] et [EEF EXISTENCE], et de certaines de leurs restrictions et dérivés.
4. **Algorithmique :** Adaptation et développement d'algorithmes de programmation par contraintes pour l'optimisation leximin.
5. **Expérimentations :**
   - Génération d'instances réalistes de problèmes de partage.
   - Comparaison expérimentale des algorithmes d'optimisation leximin.

Synthèse des contributions

1. **Modélisation des problèmes de partage :** un formalisme pour intégrer des droits exogènes inégaux dans le cadre *welfariste*.
2. **Représentation compacte des problèmes modélisés :**
   - Représentation du problème de maximisation de l'utilité collective fondé sur la logique pondérée.
   - Représentation logique du problème d'existence d'un partage sans envie et Pareto-efficace.
3. **Complexité computationnelle :** Identification de la complexité des problèmes [MAX-CUF] et [EEF EXISTENCE], et de certaines de leurs restrictions et dérivés.
4. **Algorithmique :** Adaptation et développement d'algorithmes de programmation par contraintes pour l'optimisation leximin.
5. **Expérimentations :**
   - Génération d'instances réalistes de problèmes de partage.
   - Comparaison expérimentale des algorithmes d'optimisation leximin.

Perspectives

- Partage et langages graphiques de représentation de préférences (CP-nets).
- Stratégies et (résistance à la) manipulation.
- Une étude conjointe de l'égalitarisme et de l'absence d'envie (Problématique abordée dans [?]).

Perspectives

- Partage et langages graphiques de représentation de préférences (CP-nets).
- Stratégies et (résistance à la) manipulation.
- Une étude conjointe de l'égalitarisme et de l'absence d'envie (Problématique abordée dans [**?**]).

Perspectives

- Partage et langages graphiques de représentation de préférences (CP-nets).
- Stratégies et (résistance à la) manipulation.
- Une étude conjointe de l'égalitarisme et de l'absence d'envie (Problématique abordée dans [**?**]).

Perspectives (2)

- Génération d'instances aléatoires.
- Approximation de l'équité :
  - définition de la notion d'approximation (mesure d'envie, leximin approché),
  - procédures algorithmiques approchées (schémas d'approximation polynomiaux, algorithmes incomplets).
  - absence d'envie : connaissance limitée des parts des autres agents, connaissance limitée des agents entre eux .
- Partage répété et régulation temporelle.

Perspectives (2)

- Génération d'instances aléatoires.
- Approximation de l'équité :
  - définition de la notion d'approximation (mesure d'envie, leximin approché),
  - procédures algorithmiques approchées (schémas d'approximation polynomiaux, algorithmes incomplets).
  - absence d'envie : connaissance limitée des parts des autres agents, connaissance limitée des agents entre eux .
- Partage répété et régulation temporelle.

- Un **modèle logique** générique, une **implantation logicielle** de ce modèle, et **trois générateurs**.
- $\leadsto$ Base d'un générateur de problèmes de partage réalistes et génériques, à ce jour inexistant.

Perspectives (2)

- Génération d'instances aléatoires.
- Approximation de l'équité :
  - définition de la notion d'approximation (mesure d'envie, leximin approché),
  - procédures algorithmiques approchées (schémas d'approximation polynomiaux, algorithmes incomplets).
  - absence d'envie : connaissance limitée des parts des autres agents, connaissance limitée des agents entre eux .
- Partage répété et régulation temporelle.

- Un **modèle logique** générique, une **implantation logicielle** de ce modèle, et **trois générateurs**.
- $\rightsquigarrow$ Base d'un générateur de problèmes de partage réalistes et génériques, à ce jour inexistant.

CATS  $\rightsquigarrow$ *Combinatorial Auctions Test Suite*

Perspectives (2)

- Génération d'instances aléatoires.
- Approximation de l'équité :
  - définition de la notion d'approximation (mesure d'envie, leximin approché),
  - procédures algorithmiques approchées (schémas d'approximation polynomiaux, algorithmes incomplets).
  - absence d'envie : connaissance limitée des parts des autres agents, connaissance limitée des agents entre eux .
- Partage répété et régulation temporelle.

- Un **modèle logique** générique, une **implantation logicielle** de ce modèle, et **trois générateurs**.
- $\rightsquigarrow$ Base d'un générateur de problèmes de partage réalistes et génériques, à ce jour inexistant.

CATS  $\rightsquigarrow$ *Combinatorial Auctions Test Suite*

RATS  $\rightsquigarrow$ *Resource Allocation Test Suite*

Perspectives (2)

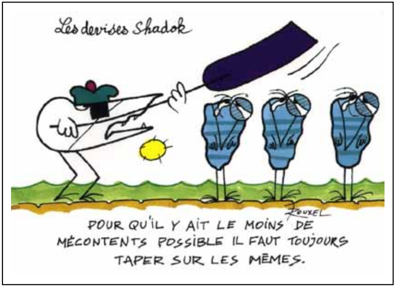
- Génération d'instances aléatoires.
- Approximation de l'équité :
  - définition de la notion d'approximation (mesure d'envie, leximin approché),
  - procédures algorithmiques approchées (schémas d'approximation polynomiaux, algorithmes incomplets).
  - absence d'envie : connaissance limitée des parts des autres agents, connaissance limitée des agents entre eux [**?**].
- Partage répété et régulation temporelle.

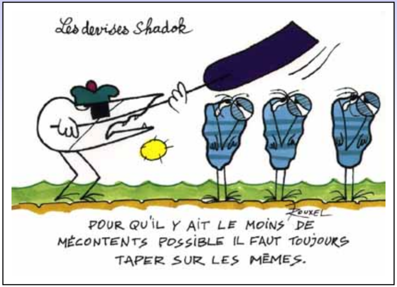
Perspectives (2)

- Génération d'instances aléatoires.
- Approximation de l'équité :
  - définition de la notion d'approximation (mesure d'envie, leximin approché),
  - procédures algorithmiques approchées (schémas d'approximation polynomiaux, algorithmes incomplets).
  - absence d'envie : connaissance limitée des parts des autres agents, connaissance limitée des agents entre eux .
- Partage répété et régulation temporelle.

Perspectives (2)

- Génération d'instances aléatoires.
- Approximation de l'équité :
  - définition de la notion d'approximation (mesure d'envie, leximin approché),
  - procédures algorithmiques approchées (schémas d'approximation polynomiaux, algorithmes incomplets).
  - absence d'envie : connaissance limitée des parts des autres agents, connaissance limitée des agents entre eux .
- Partage répété et régulation temporelle.





Soutenance de thèse : **Allocation et partage équitables de ressources indivisibles : modélisation, complexité et algorithmique**
Sylvain Bouveret

Jury :
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**