Théorie des jeux algorithmique
(algorithmic game theory)

Exposé « Eco et Opti », Nanterre, le 16/12/2008

La TJA : motivations

- Fin 20e siècle : développement d'internet
  => changement du rôle d'un ordinateur.
- Avant: entité isolée, exécutant des logiciels.
- Maintenant: interaction avec d'autres ordinateurs, partage de ressources et informations, commerce.

La TJA : motivations

- Informatique (théorique) : nouveaux problèmes, buts, techniques d'analyse et de conception de protocoles compatibles avec cette nouvelle réalité.
- Théorie des jeux : étude approfondie des interactions entre individus en compétition (ou qui coopèrent) : rôle crucial.
- Interface info et TJ => Théorie des jeux algorithmique.

Différences avec la théorie des jeux

- Applications : réseaux liés à internet; enchères non traditionnelles.
- Approche : on modélise des applications via des problèmes d'optimisation concrets et on cherche:
  - Solution optimale
  - Résultats d'impossibilité
  - Bornes inf, sup sur le rapport d'approximation, etc.
- Contrainte: Complexité raisonnable (polynomiale).

Connexion avec l'info théorique

- Analyse d'équilibres : outils de la conception d'algorithmes approchés, programmation mathématique, fonctions potentielles.
- Complexité de calcul d'équilibre : regain d'intérêt pour certaines classes de complexité relatives par ex. à la recherche locale.
- Systèmes d'enchères: techniques usuelles en info théorique (par ex algorithmes primal-dual).

Plan

1. Mesure de la qualité des équilibres
2. Complexité de calcul des équilibres
3. « Algorithmic mechanism design »
4. Conclusion et perspectives

Plan

1. Mesure de la qualité des équilibres
   - Problème d'optimisation combinatoire
   - Rapport d'approximation
   - Equilibre de Nash
   - Jeu
   - Prix de l'anarchie
   - Prix de la stabilité

Problème d'optimisation combinatoire

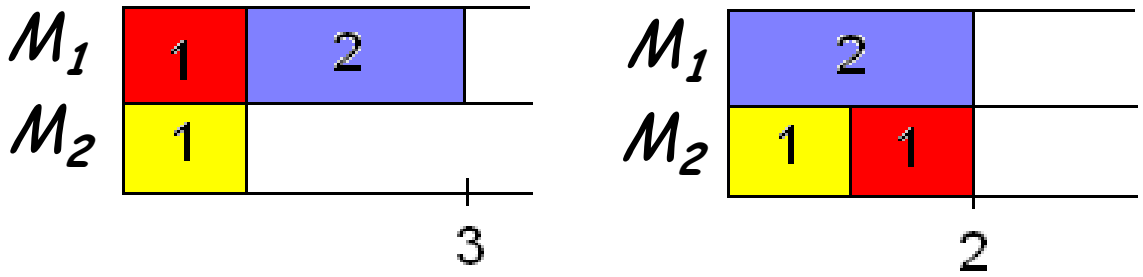
- On a :
  - Un ensemble d'instances (données)
  - Pour chaque instance : un ensemble de solutions réalisables
  - Une fonction objectif
- But :

  Trouver un algorithme qui retourne la meilleure solution pour la fonction objectif.

Problème d'optimisation combinatoire

- Exemple : un problème d'ordonnancement
  - m machines identiques
  - n tâches de longueurs différentes

Ordonnancements réalisables :



  - But: minimiser la date de fin maximale (le makespan) .

Algorithmes d'approximation

Rapport d'approximation :

$$\max_{\text{instances}} \frac{\text{Valeur de la fonction objectif dans la solution retournée}}{\text{Valeur de la fonction objectif dans une solution optimale}}$$

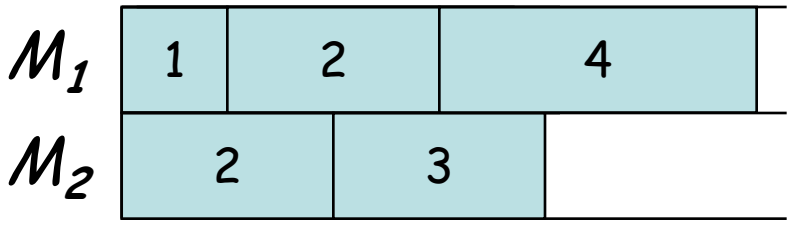
Un algorithme 2-approché retournera, pour chaque instance, une solution au pire 2 fois moins bonne que la solution optimale.

Algorithmes d'approximation

Exemple :

SPT: ordonnance les tâches de la plus petite à la plus grande

2 machines, tâches de longueur 1, 2, 2, 3, 4 :



Algorithme 2-1/m approché.

Equilibre de Nash

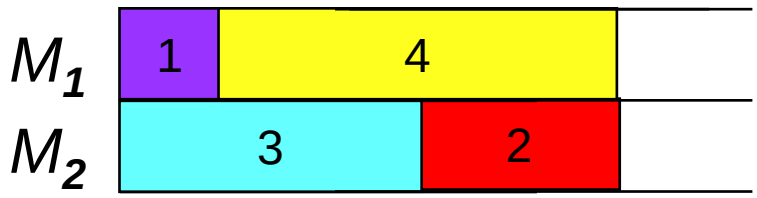
Ensemble d'agents, chacun ayant un but (fonction objectif individuelle) et un ensemble de stratégies possibles.

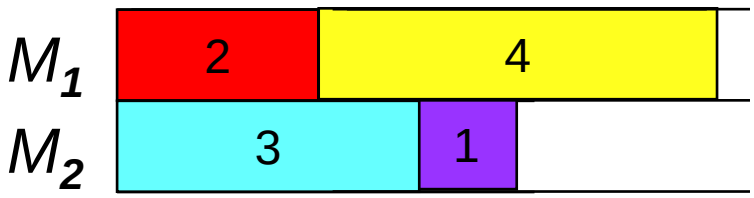
On suppose que chacun connait les stratégies des autres.

Equilibre de Nash (EN): situation dans laquelle aucun agent n'a intérêt à changer unilatéralement de stratégie.

Jeu

- Un ensemble d'agents.
- (éventuellement) des règles qui gèrent les ressources utilisées par les agents.
- Une fonction obj. globale («social welfare»)
- Exemple :
  - Agents= tâches. Leur but : être exécuté sur la machine la moins chargée.
  - Fonction obj. globale : minimiser le makespan.





Prix de l'anarchie

- But: Analyser la qualité de la solution obtenue quand les agents se conduisent de façon individualiste.

- Définition :[Koutsoupias et Papadimitriou, STACS'99]

$$\max_{\text{instances}} \frac{\text{Valeur de la fonction objectif dans le pire EN}}{\text{Valeur de la fonction objectif dans une solution optimale}}$$

Prix de l'anarchie

- Motivation :
  - Analyse de la perte de performance due au comportement égoïste des agents (si prix de l'anarchie faible: un équilibre optimise presque la fonction obj. globale).
  - Fixer des règles au système (aux ressources) de façon à diminuer le prix de l'anarchie.

- Exemple : routage de tâches dans un réseau, ordonnancement

Prix de la stabilité

- On utilise un algorithme qui suggère une stratégie à chaque agent.
- But: obtenir une solution bonne vis-à-vis de la fonction obj. globale et stable.
- Prix à payer: prix de la stabilité [Schultz et al, SODA'03] et [Anshelevich et al, FOCS'04] :

$$\max_{\text{instances}} \frac{\text{Valeur de la fonction objectif dans le meilleur EN}}{\text{Valeur de la fonction objectif dans une solution optimale}}$$

Plan

1. Mesure de la qualité des équilibres
2. Complexité de calcul des équilibres
3. « Algorithmic mechanism design »
4. Conclusion et perspectives

Motivation

- En combien de temps les participants à un jeu convergent vers un équilibre?
- En combien de temps un algorithme centralisé peut retourner un équilibre ?

- En effet:
  - Utilité de savoir construire un équilibre
  - Doute possible sur la crédibilité du modèle s'il n'est pas possible de construire un équilibre en un temps polynomial.

Complexité du calcul d'un équilibre

- En informatique théorique : problèmes partitionnés dans des classes de complexité (par ex. P, NP)
- Calcul des équilibres de Nash mixtes dans des jeux finis : appartient à la classe PPAD. [Chen et al, J. of the ACM 2008] et [Daskalakis et al, SIAM J. of Computing 2008] .

- Autres résultats pour d'autres équilibres ou pour des jeux plus spécifiques.

Plan

1. Mesure de la qualité des équilibres
2. Complexité de calcul des équilibres
3. « Algorithmic mechanism design »
4. Conclusion et perspectives

Algorithmic Mechanism Design

- Etude de problèmes d'optimisation dans lequel les données (par ex: valeur d'un bien, coût pour effectuer une tâche) sont inconnues du concepteur de l'algorithme et sont déclarées par des utilisateurs individualistes.
- But: construire un protocole (mécanisme) qui interagit avec les agents et t.q. leur comportement individualiste mène a une solution désirable.

Algorithmic Mechanism Design

- Terme introduit par [Nisan et Ronen, Games and Economic Behavior, 2001] : étude systématique de ce qui peut être efficacement calculé ou approximé quand les données sont détenues par des agents individualistes.
- AMD: s'appuie sur le domaine du « mecanism design », mais s'intéresse surtout à savoir à quel point il est plus difficile de concevoir des algorithmes résistant à des données « privées ».

Algorithmes à véracité garantie

- Algorithmes avec lesquels chaque agent a intérêt à déclarer sa vraie valeur.

- => On est sûr des performances d'un algorithme à véracité garantie (truthful)

Mécanisme

- Chaque agent a une seule valeur secrète ti et déclare bi.
- Mécanisme M: algorithme d'allocation x(b1,…,bn) et algorithme de paiement Π(b1,…,bn). Détermine une solution s et des paiements p1,…,pn.
- Mécanisme à véracité garantie (incentive compatible) si chaque agent i maximise son utilité en reportant bi=ti.

Algorithme implémentable

- Algorithme qui, avec une fonction de paiement bien choisie, mène à un mécanisme à véracité garantie.
- Est-ce que les algorithmes implémentables sont moins puissants que des algorithmes classiques ?
- Question intéressante pour des algorithmes polynomiaux (peu étudié en économie) ou non (déjà beaucoup étudié).

Quelques résultats

- Résultat 1 (single parameter mechanism design) [Mookherjee et Reichelstein, J. of Economic Theory, 1992] :Un algorithme est implémentable ssi il est monotone.

- Résultat 2 (problème d'ordonnancement) [Dhangwatnotai et al., FOCS'08] : Pour tout $\varepsilon>0$, il existe un algorithme $(1+\varepsilon)$-approché et implémentable.

Mécanismes sans paiement

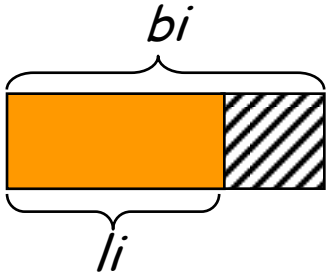
- Exemple: problème d'ordonnancement

- Chaque tâche *i* est détenue par un agent qui seul connaît sa longueur.

- Les tâches communiquent leur longueur à un protocole qui doit les ordonnancer de sorte à minimiser le *makespan.*

Mécanismes sans paiement

- L'algorithme d'ordonnancement est connu.
- Le but de chaque tâche est de minimiser sa date de fin d'exécution.
- Chaque tâche *i* déclare une valeur *bi* représentant sa longueur.
- Hyp : $bi \geq li$ (exécution incomplète si $bi<li$)



Résultats

- Etude du rapport d'approximation d'un algorithme à véracité garantie :

	Déterministe		Probabiliste	
	inf.	sup.	inf.	sup.
Rapport d'approximation	m=2: 1.1 m≥3: 7/6	4/3-1/(3m)	1	1

Conclusion et Perspectives

- Nombreux problèmes ouverts en TJA. Domaine en plein essor.
- Utilisation de modèles plus appropriés pour modéliser le comportement des agents.