Coordination et Consensus Multiagents (3)

N. Maudet

2017–2018

CoCoMA — Master ANDROIDE

Auctions

Auctions are a very convenient way to allocate resources or tasks:

- agents submit bids on single or bundle of items, ie. prices that are ready to pay for that item/bundle.
- an auctioneer gathers the bids and computes the optimal allocation
- agents pay according to the payment scheme.

Summary of the course

Single Item Auctions

Matching by Auctioning Items

Combinatorial Auctions

Application: Keywords Auctions

Application: Auctions for Robot Coordination

Single Item Auctions

Basic Setting: Allocating an Item

There are many different types of auctions: English, Dutch, etc.

- ascending vs. descending
- sealed bids vs. open bids
- first price vs. second price

Truthfulness: Allocating an Item

Suppose we want to allocate a single item among n agents and implement the following rule:

- agents report sealed bids
- the item goes to the agent who values it the most
- the price payed by this agent is the second highest price

Truthfulness: Allocating an Item

Suppose we want to allocate a single item among n agents and implement the following rule:

- agents report sealed bids
- the item goes to the agent who values it the most
- the price payed by this agent is the second highest price

This is called a Vickrey auction. It has the enjoyable property that it is a dominant strategy for agents to reveal their true preferences ($\overline{v}(o)$). It is said to be strategy-proof.

Truthfulness: Allocating an Item

Intuition:

- suppose the agent i overbids: but the only case where this affects the allocation is if an agent bidded between $v_i(o)$ and $\overline{v_i}(o)$: i would get the item for a price higher than valuation!
- suppose that agent i underbids: only makes less likely to get the item...

Matching by Auctioning Items

Matching by Auctioning Items

- assume cardinal preferences of agents
- agents not initially held by agents
- interestingly, a simple protocol based on auctions allows to find a matching

D. Bertsekas. *Auction Algorithms for Network Flow Problems.* Computational Optimization and Applications, 1992.

Allouer une resssource par agent

1. on associe à chaque ressource un prix, initialisé à 0.
2. on choisit un agent auquel aucune ressource n'est encore allouée
3. on cherche l'objet o^* qui, à son prix actuel p_{o^*} et pour cet agent, donne la meilleure utilité

$$o^* \in \operatorname{argmax}_{o \in \mathcal{R}}(v_i(o) - p_o)$$

4. l'agent augmente le prix de l'objet le plus possible sans rendre un autre objet plus attractif.
5. l'agent se voit alloué l'objet (qui est donc retiré à un autre agent s'il était déjà alloué)

Allouer une resssource par agent (Exemple)

Les agents ont les utilités suivantes:

	a	b	c
x	1	0	4
y	6	8	0
z	0	6	1

tour	p_a	p_b	p_c	agent	objet	offre	allocation
0	0	0	0	x	c	3	$\langle x, c \rangle$

Allouer une resssource par agent (Exemple)

Les agents ont les utilités suivantes:

	a	b	c
x	1	0	4
y	6	8	0
z	0	6	1

tour	p_a	p_b	p_c	agent	objet	offre	allocation
0	0	0	0	x	c	3	$\langle x, c\rangle$
1	0	0	3	y	b	2	$\langle x, c\rangle, \langle y, b\rangle$

Allouer une resssource par agent (Exemple)

Les agents ont les utilités suivantes:

	a	b	c
x	1	0	4
y	6	8	0
z	0	6	1

tour	p_a	p_b	p_c	agent	objet	offre	allocation
0	0	0	0	x	c	3	$\langle x, c\rangle$
1	0	0	3	y	b	2	$\langle x, c\rangle, \langle y, b\rangle$
2	0	2	3	z	b	4	$\langle x, c\rangle, \langle z, b\rangle$

Allouer une resssource par agent (Exemple)

Les agents ont les utilités suivantes:

	a	b	c
x	1	0	4
y	6	8	0
z	0	6	1

tour	p_a	p_b	p_c	agent	objet	offre	allocation
0	0	0	0	x	c	3	$\langle x, c\rangle$
1	0	0	3	y	b	2	$\langle x, c\rangle, \langle y, b\rangle$
2	0	2	3	z	b	4	$\langle x, c\rangle, \langle z, b\rangle$
3	0	6	3	y	a	4	$\langle x, c\rangle, \langle z, b\rangle, \langle y, a\rangle$

Le processus stoppe avec l'allocation
$A(x) = \{c\}, A(z) = \{b\}, A(y) = \{a\}$ et le vecteur de prix $\langle 4, 6, 3\rangle$.
Les utilités des agents sont donc: 1 pour x, 2 pour y, et 0 pour z.

Equilibre compétitifs et allocations efficaces

- Une allocation A et un vecteur de prix sont en équilibre compétitif quand, pour tout agent i, on a

$$\forall o, u_i(A(i)) \geq u_i(o)$$

- le processus d'enchère termine (hmm...) en équilibre compétitif
- en fait:
  (1) si une allocation et un vecteur de prix sont en équilibre compétitif alors cette allocation est nécessairement optimale, et
  (2) pour toute allocation efficace A il existe un vecteur de prix p tels que A et p sont en équilibre compétitif

Combinatorial Auctions

Combinatorial Auctions

In the combinatorial auction setting, agents bid on bundle of items (as opposed to single items) There are two questions which occur:

- how can agents express preferences on bundles of items?
- how can the auctioneer compute the optimal (= maximizing the revenue) allocation given the submitted bids
  This is known as the Winner Determination Problem (of course, NP-complete)

. *Combinatorial Auctions.* .

Bidding Languages (1)

Consider single-minded (aka atomic) bids.

Agents submit a single bid on one bundle: $\langle B, p \rangle$

The agent is committed to pay p for any $S \supseteq B$

Problem: not expressive at all.

N. Nisan. *Bidding Languages*. .

Bidding Languages: XOR language

When an agent submits bids of the form:

$$\langle B_1, p_1 \rangle \text{XOR} \langle B_2, p_2 \rangle \ldots \langle B_n, p_n \rangle$$

it is committed to pay, for a set of items S, $\max_{i | S_i \subseteq S} p_i$

Intuition: "At most one of the bids"

Bidding Languages: OR language

When an agent submits bids of the form:

$$\langle B_1, p_1 \rangle \mathrm{OR} \langle B_2, p_2 \rangle \ldots \langle B_n, p_n \rangle$$

it is committed to pay, for a set of items S, the maximum (over all the non-overlapping collections W of subsets S_i) of $\sum_{i \in W} p_i$.

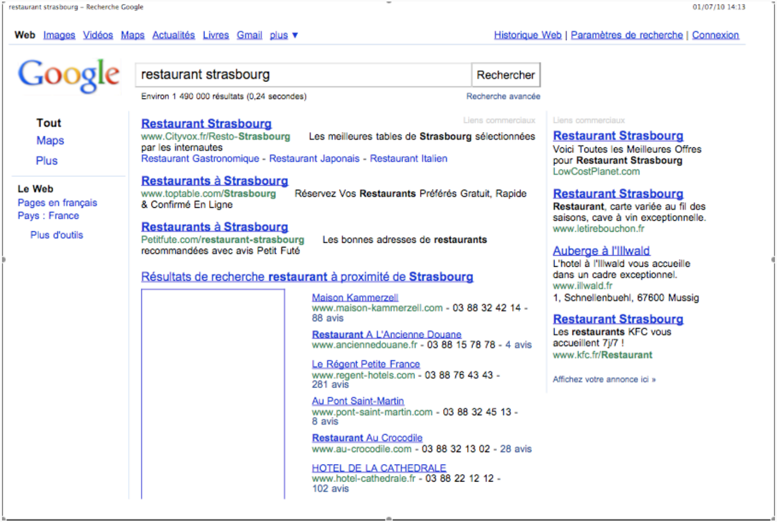
Intuition: same as different atomic bids from several bidders

Bidding Languages: OR^* language

Idea: introduce fake items to discard some collections of the OR language.

Application: Keywords Auctions

Enchères de mots-clefs



Allocations de positions publicitaires par les moteurs de recherche

- le processus d'allocation des positions aux agents est géré grâce à des enchères qui ont lieu *en continu*
- les ressources à allouer sont les *positions* sur la page de recherche donnée en réponse à un (ensemble de) mot-clefs(s).
- les agents sont d'accord sur les positions les plus souhaitables sur la page, classées par *taux de clic* t_j
- chaque agent possède une valuation personnelle pour un clic $\widehat{v_i}$
- la valuation d'une position j pour un agent i est donc

$$v_i(j) = \widehat{v_i} \times t_j$$

- l'utilité d'un agent i pour une position j est toujours

$$u_i(j) = v_i(j) - p_i$$

B. Edelman, M. Ostrovsky, M. Schwartz. *Internet Advertising and the Generalized Second Priced Auction*. American Economic Review, 2007.

Enchères de premier prix

- chaque agent i fait une offre unique b_i (très simple!), et les positions sont attribuées dans l'ordre des valeurs des offres
- lorsqu'un utilisateur clique sur le lien sponsorisé, l'agent paye b_i
- problèmes de "cycles" en raison du caractère continu de l'enchère

Enchères de premier prix

- chaque agent i fait une offre unique b_i (très simple!), et les positions sont attribuées dans l'ordre des valeurs des offres
- lorsqu'un utilisateur clique sur le lien sponsorisé, l'agent paye b_i
- problèmes de "cycles" en raison du caractère continu de l'enchère

Exemple: 2 agents x_1 avec $\widehat{v_1} = 0.6$ et agent x_2 avec $\widehat{v_2} = 0.8$.

- agent x_1 offre 0.4
- agent x_2 offre 0.5
- agent x_1 offre 0.6
- agent x_2 offre 0.7

Enchères de premier prix

- chaque agent i fait une offre unique b_i (très simple!), et les positions sont attribuées dans l'ordre des valeurs des offres
- lorsqu'un utilisateur clique sur le lien sponsorisé, l'agent paye b_i
- problèmes de "cycles" en raison du caractère continu de l'enchère

Exemple: 2 agents x_1 avec $\widehat{v_1} = 0.6$ et agent x_2 avec $\widehat{v_2} = 0.8$.

- agent x_1 offre 0.4
- agent x_2 offre 0.5
- agent x_1 offre 0.6
- agent x_2 offre 0.7

... mais à ce moment là l'agent 1 a intérêt à faire l'offre minimale acceptable (pour garder la deuxième position)

Enchères de deuxième prix généralisées (GSP)

- chaque agent i fait une offre unique b_i (très simple!), et les positions sont attribuées dans l'ordre des valeurs des offres (avec $\pi(k)$ l'agent placé à la position k)
- le principe du *Generalized Second Price* est proposé en 2002: chaque agent paie l'offre de celui juste derrière lui dans l'ordre

$$p_{\pi(k)} = b_{\pi(k+1)} \cdot t_k$$

- Google affime que

> *unique auction model uses Nobel Prize-winning economic theory to eliminate (...) that feeling that you have paid too much.*

GSP: analyse des équilibres

- Nash: aucun agent n'a intérêt à dévier seul
- Quelles sont les déviations envisageables pour un agent à la position j?
  - obtenir une position mieux placée $k < j$: il faut alors battre l'offre de $\pi(k)$, pour une utilité

$$t_k(\hat{v}_i - b_{\pi(k)})$$

  - obtenir une position moins bien placée $k > j$: il faut faire une offre plus basse que $b_{\pi(k)}$, pour une utilité

$$t_k(\hat{v}_i - b_{\pi(k+1)})$$

- Note: les deux situations ne sont pas symétriques

Leme & Tardos. *Pure and Bayes-Nash Price of Anarchy for Generalized Second Price Auctions*. FOCS-2010.

GSP: analyse des équilibres

Exemple: agents x_1 avec $\widehat{v_1} = 1$ et agent x_2 avec $\widehat{v_2} = 0$, $r \in]0, 1[$.

position	taux	agent	offre
1	1	x_2	$b_2 = 1 - r$
2	r	x_1	$b_1 = 0$

- agent x_1 ($\Uparrow$), si $1(1 - r) > r(1 - 0)$
- agent x_2 ($\Downarrow$): si $r(0 - 0) > 1(0 - 0)$

En prenant $r \to 0$ on a bien un équilibre de Nash et le rapport entre la somme des utilités des agents du meilleur équilibre vs. cet équilibre (prix de l'anarchie) est arbitrairement grand ($1/r$).

GSP: analyse des équilibres

Exemple: agents x_1 avec $\widehat{v_1} = 1$ et agent x_2 avec $\widehat{v_2} = 0$, $r \in]0, 1[$.

position	taux	agent	offre
1	1	x_2	$b_2 = 1 - r$
2	r	x_1	$b_1 = 0$

- agent x_1 ($\Uparrow$), si $1(1 - r) > r(1 - 0)$
- agent x_2 ($\Downarrow$): si $r(0 - 0) > 1(0 - 0)$

En prenant $r \to 0$ on a bien un équilibre de Nash et le rapport entre la somme des utilités des agents du meilleur équilibre vs. cet équilibre (prix de l'anarchie) est arbitrairement grand ($1/r$).

- mais cet équilibre semble très improbable... pourquoi faire une offre plus haute que 0 pour x_2?
- de manière genérale, faire une offre $b_i > v_i$ est dominé par $b_i = v_i$

GSP: garantie de véracité?

- deux positions disponibles, avec $t_1 = 200$ et $t_2 = 100$
- trois agents avec les valuations par clic $\widehat{v_1} = 10$, $\widehat{v_2} = 4$, $\widehat{v_3} = 2$
- si les agents font des offres honnêtes

position	taux	agent	offre
1	200	x_1	$b_1 = 10$
2	100	x_2	$b_2 = 4$
$---$	$---$	x_3	$b_3 = 2$

1. x_1 obtient s_1 au prix de 4, x_2 obtient s_2 au prix de 2.
2. $u_1(s_1) = 200 \times (10 - 4) = 1200$, et $u_2(s_2) = 100 \times (4 - 2) = 200$
3. le moteur de recherche obtient $200 \times 4 + 100 \times 2 = 1000$

- aucun agent n'a intérêt à changer de position

  Est-ce toujours le cas?

GSP: garantie de véracité?

- deux positions disponibles, avec $t_{s_1} = 200$ et $t_{s_2} = \mathbf{190}$
- trois agents avec les valuations par clic $v_1 = 10$, $v_2 = 4$, $v_3 = 2$
- si les agents font des offres honnêtes

position	taux	agent	offre
1	200	x_1	$b_1 = 10$
2	190	x_2	$b_2 = 4$
$---$	$---$	x_3	$b_3 = 2$

1. x_1 obtient s_1 au prix de 4, x_2 obtient s_2 au prix de 2.
2. $u_1(s_1) = 200 \times (10 - 4) = 1200$, et $u_2(s_2) = 190 \times (4 - 2) = 380$
3. le moteur de recherche obtient $200 \times 4 + 100 \times 2 = 1180$

- mais si l'agent x_1 offre seulement 3, il aura s_2 et son utilité sera $u_1(s_2) = 190 \times (10 - 2) = 1520 (> 1200)$

GSP: équilibres avec absence d'envie locale

- en pratique, les agents appliquent des stratégies simples: par exemple en augmentant leurs offres petit à petit, de manière à gagner la position juste au dessus (squeezing).
- note: le prix payé par l'agent lui-même n'augmente pas, mais celui de l'agent au-dessus si.
- un équilibre est localement sans envie si aucun agent n'a intérêt à échanger sa position avec celle de l'agent juste au dessus de lui dans le classement.

Application: Auctions for Robot Coordination

Several places to be visited (only once) by different agents.

Multi-robot exploration problem: allocate places to agents and find a path for each agent such the sum of all paths is minimized.

Auction-based approaches:

- combinatorial auctions: compute the TSP for each subset of nodes. Place the bids. Auctioneer computes optimal allocation.
  Problem: not feasible to bid on all bundles!
- single-item auctions: each agent bid the cost from their current location to the place they bid for.

Is there a better way to proceed?

Canwe provide performance guarantees and escape the combinatorial auction computational burden?

Recall that a Minimum Spanning Tree (MST) can be found in polynomial time.

Prim's algorithm: to build the tree, start from a node, then add new nodes by selecting the cheapest edge between any node already in the tree and any node not in the tree.

The MST solution provide a 2-approximation TSP.

The multi-robot exploration problem seeks a minimum spanning forest (a collection of trees).

Lagoukadis et al.. *Simple Auctions with Performance Guarantees for Multi-Robot Task Allocation*. IROS-2004.

Auction version

Multi-round auction where one target is allocated at each round.

- each agent estimates the minimum cost from one of its allocated target or the current robot location to each target still available
- at a given round, each agent submits only its best (=lowest) bid
- the auctioneer collects the bids and assign the target corresponding to the lowest bidding agent