Game Theory and Planning

Selected Topics

Radu Prodan, Georg Moser

Topics Overview

- Learning, regret minimisation, and equilibria
- Computation of market equilibria by convex programming
- Graphical games
- Mechanism design
- Combinatorial auctions
- Routing games
- Selfish load balancing
- Price of anarchy and the design of scalable resource allocation mechanisms
- Cascading behaviour in networks: algorithmic and economic issues
- Sponsored search auctions

Topics Overview

- **Learning, regret minimisation, and equilibria**
- Computation of market equilibria by convex programming
- Graphical games
- Mechanism design
- Combinatorial auctions
- Routing games
- Selfish load balancing
- Price of anarchy and the design of scalable resource allocation mechanisms
- Cascading behaviour in networks: algorithmic and economic issues
- Sponsored search auctions

Learning, Regret Minimisation, and Equilibria

- Repeatedly making decisions in an uncertain environment against opponents with an unknown strategy
- M players, set of N actions
  - What route to drive to work?
  - Rock, scissors, paper
- Design adaptive algorithms
- Regret analysis (against a simple alternative policy)
  - External regret (combining expert advice)
  - Internal or swap regret
- Full versus partial information model
- Models for approaching Nash equilibrium when minimising external regret
- Price of anarchy when using selfish adaptive behaviour

Topics Overview

- Learning, regret minimisation, and equilibria
- **Computation of market equilibria by convex programming**
- Graphical games
- Mechanism design
- Combinatorial auctions
- Routing games
- Selfish load balancing
- Price of anarchy and the design of scalable resource allocation mechanisms
- Cascading behaviour in networks: algorithmic and economic issues
- Sponsored search auctions

Economy Models

- Market equilibrium problem
  - Find a set of prices and allocation of goods to economic agents such that each agent maximises his utility subject to budget constraints and the market clears
- Exchange economy model (Arrow and Debreu, 1954)
  - m traders, n goods
  - Concave utility function: $u_i : R_+^n \to R_+, i \in [1..m]$
    - ✓ Non-satiable: $\forall x \in R_+^n, \exists y \in R_+^n : u_i(y) > u_i(x)$
    - ✓ Monotone: $u_i(y) \geq u_i(x), y \geq x$
  - Initial goods endowment: $w_i = (w_{i1},\ldots,w_{in}) \in R_+^m, \exists j \in [1..n] : w_{ij} > 0, \forall i \in [1..m]$
  - Find an equilibrium vector of prices: $\pi = (\pi_1, \ldots, \pi_n) \in R_+^n$ and an allocation of goods $x_i = (x_{i1}, \ldots, x_{in})$ such that $u_i(x)$ is maximised:

$$\max \{u_i(x)\}, \forall i \in [1,m]$$

$$\pi \cdot x \leq \pi \cdot w_i \wedge x \in R_+^n$$

$$\sum_{i=1}^{m} x_{ij} = \sum_{i=1}^{m} w_{ij}, \forall j \in [1,n]$$

- Fisher model
  - Buyers have fixed money endowment e_i
  - Each good j is sold by one trader in quantity q_j

$$\pi \cdot x \leq e_i \wedge x \in R_+^n$$

$$\sum_{i=1}^{m} x_{ij} = q_j, \forall j \in [1,n]$$

Topics Overview

- Learning, regret minimisation, and equilibria
- Computation of market equilibria by convex programming
- **Graphical games**
- Mechanism design
- Combinatorial auctions
- Routing games
- Selfish load balancing
- Price of anarchy and the design of scalable resource allocation mechanisms
- Cascading behaviour in networks: algorithmic and economic issues
- Sponsored search auctions

Graphical Games

- Classical multiplayer game (in normal form)
  - N players
  - Binary action space (pure strategies): $a_i \in \{ 0, 1 \}, \forall i \in [1..N]$
  - Mixed strategy: probability $p_i \in [0,1]$ to play 0
  - Payoff of player i: M_i: $\{ 0, 1 \}^N \to [0,1]$
  - Approximate ε-Nash equilibrium is a mixed strategy: $\bar{p} = (p_1,\ldots,p_N)$

    $M_i(\bar{p}) + \varepsilon \geq M_i(\bar{p}[i: p_i']), \forall p_i' \in [0,1], \forall i \in [1..N]$
  - p_i is an ε–best response to the rest of $\bar{p}$
- Graph theoretic model for multiplayer games
  - $G = (N, Edges)$
  - $N_i \subseteq \{ 1, \ldots, N \}$ is the neighbourhood of player i

    $Edges = \{(i,j) : \forall i \in [1..N], \forall j \in N_i\}$
  - Graphical game: $\left(G, \bigcup_{i=1}^{N} M_i'\right)$
  - Local game matrix M_i' is the projection of M_i onto N_i

Topics Overview

- Learning, regret minimisation, and equilibria
- Computation of market equilibria by convex programming
- Graphical games
- **Mechanism design**
- Combinatorial auctions
- Routing games
- Selfish load balancing
- Price of anarchy and the design of scalable resource allocation mechanisms
- Cascading behaviour in networks: algorithmic and economic issues
- Sponsored search auctions

Mechanism Design

- Subfield of economic theory with an engineering perspective
- Engineer of games rules so that the outcome of the games is optimal
- Design mechanisms in terms of social choices assuming rational participants
  - Elections, market, auctions, governmental policies
- Algorithmic mechanism design (economics for computer science)
  - Internet operated by parties with different goals and preferences
  - Routing of messages
  - Scheduling of tasks
- Electronic market design (computer science for economics)
- Voting methods
  - Majority vote, strategic vote
  - Condorcet's paradox
  - Social welfare function
  - Social choice function
- Mechanisms with and without money

(1) $a \succ_1 b \succ_1 c$

(2) $b \succ_2 c \succ_2 a$

(3) $c \succ_3 a \succ_3 b$

Topics Overview

- Learning, regret minimisation, and equilibria
- Computation of market equilibria by convex programming
- Graphical games
- Mechanism design
- **Combinatorial auctions**
- Routing games
- Selfish load balancing
- Price of anarchy and the design of scalable resource allocation mechanisms
- Cascading behaviour in networks: algorithmic and economic issues
- Sponsored search auctions

Combinatorial Auctions

- *m* indivisible items
- Concurrently auctioned among *n* bidders
- Valuation function $v(S)$ for each subset S
  - Monotone: $S \subseteq T \Rightarrow v(S) \leq v(T)$
  - Normalised: $v(\emptyset) = 0$
  - Private to the bidder
- $S \cap T = \emptyset$
  - Complements: $v(S \cup T) > v(S) + v(T)$
  - Substitutes: $v(S \cup T) < v(S) + v(T)$
- Item allocation among bidders: $S_1, \ldots, S_n$
  - $S_i \cap S_j = \emptyset, \forall i \neq j$
- Social welfare: $\sum_{i=1}^{n} v_i(S_i)$
- Computational complexity, representation and communication, strategies

Topics Overview

- Learning, regret minimisation, and equilibria
- Computation of market equilibria by convex programming
- Graphical games
- Mechanism design
- Combinatorial auctions
- **Routing games**
- Selfish load balancing
- Price of anarchy and the design of scalable resource allocation mechanisms
- Cascading behaviour in networks: algorithmic and economic issues
- Sponsored search auctions

Routing Games

- Route traffic in large communication networks with no central authority, such as the Internet
- Networks with source routing
  - Distributed shortest path routing
- Non-atomic selfish routing
  - Each commodity is a large population of individuals controlling a small amount of traffic
- Atomic selfish routing
  - Each commodity is a single player controlling a larger amount of traffic on a single path

Topics Overview

- Learning, regret minimisation, and equilibria
- Computation of market equilibria by convex programming
- Graphical games
- Mechanism design
- Combinatorial auctions
- Routing games
- **Selfish load balancing**
- Price of anarchy and the design of scalable resource allocation mechanisms
- Cascading behaviour in networks: algorithmic and economic issues
- Sponsored search auctions

Selfish Load Balancing

- Makespan scheduling on uniformly related machines
  - $[m] = \{ 1, \ldots, m \}$ set of machines with speeds $s_1, \ldots, s_m$
  - $[n] = \{ 1, \ldots, n \}$ set of tasks with weights $w_1, \ldots, w_n$
  - Allocation: $A: [n] \rightarrow [m]$
  - Load of machine $j \in [m]$: $l_j = \sum_{\substack{i \in [n] \\ j = A(i)}} \frac{w_i}{s_j}$
  - Goal: minimise the maximum load (makespan)
- Makespan scheduling on identical machines
  - $s_1 = \ldots = s_m = 1$
- Multiple selfish users in Internet assigning tasks to machines

Topics Overview

- Learning, regret minimisation, and equilibria
- Computation of market equilibria by convex programming
- Graphical games
- Mechanism design
- Combinatorial auctions
- Routing games
- Selfish load balancing
- **Price of anarchy and the design of scalable resource allocation mechanisms**
- Cascading behaviour in networks: algorithmic and economic issues
- Sponsored search auctions

Price of Anarchy and Design of Scalable Resource Allocation Mechanisms

- R users compete for sharing an infinitely divisible resource of capacity $C > 0$
- Each user $r \in [1..R]$ gives a bid $w_r \geq 0$ to the resource manager
- Given the vector $w=(w_1, \ldots, w_R)$, resource manager selects an allocation $d=(d_1, \ldots, d_R)$
- Utility $U_r(d_r)$ of user $r \in [1..R]$ is concave, strictly increasing, continuous, …
- No price discrimination: each user is charged the same price $\mu>0$
- Goal:

$$max\left\{\sum_{r=1}^{R} U_r(d_r)\right\}$$

$$\sum_{r=1}^{R} d_r \leq C$$

$$d_r \geq 0$$

- Proportional allocation mechanism
- Full efficiency for users as price takers
- Price of anarchy for users as price anticipators
- Implication of price discrimination (per user)

$$d_r = \frac{w_r}{\mu}$$

$$\sum_{i=1}^{R} \frac{w_r}{\mu} = C$$

$$\mu = \frac{\sum_{i=1}^{R} w_r}{C}$$

Topics Overview

- Learning, regret minimisation, and equilibria
- Computation of market equilibria by convex programming
- Graphical games
- Mechanism design
- Combinatorial auctions
- Routing games
- Selfish load balancing
- Price of anarchy and the design of scalable resource allocation mechanisms
- **Cascading behaviour in networks: algorithmic and economic issues**
- Sponsored search auctions

Cascading Behaviour in Networks

- Social networks
- New ideas and behaviours spread through a population
  - Religious beliefs, political movements, technological innovations, new products, celebrities
- $G = (V, E)$
  - V are individuals
  - $(v, w) \in E$ are friends
  - Two behaviours: A (old) and B (new)
  - d_v = degree of node v

    $d_v = d_v^A + d_v^B$
- Contagious threshold of social network G (max q)
- Progressive versus non-progressive
- Influential nodes

$$payoff(v) = \begin{cases} q \cdot d_v^A, & v \text{ chooses } A \\ (1-q) \cdot d_v^B, & v \text{ chooses } B \end{cases}$$

$$choice(v) = \begin{cases} A, & d_v^B < q \cdot d_v \\ B, & d_v^B \geq q \cdot d_v \end{cases}$$

vw	Payoff
AA	*q*
AB	0
BA	0
BB	1-*q*

Topics Overview

- Learning, regret minimisation, and equilibria
- Computation of market equilibria by convex programming
- Graphical games
- Mechanism design
- Combinatorial auctions
- Routing games
- Selfish load balancing
- Price of anarchy and the design of scalable resource allocation mechanisms
- Cascading behaviour in networks: algorithmic and economic issues
- **Sponsored search auctions**

Sponsored Search Auctions

- Sponsoring search
  - Advertisers bid for placement in an auction-style format
  - Keywords, bids, budget (daily, weekly)
  - Pay-per-click
- n bidders, $m<n$ slots
- Estimated click through rate α_{ij}
  - Probability that a user clicks on the i^{th} the slot occupied by bidder j
  - $\alpha_{ij} \geq \alpha_{i+1,j}, \forall i \in [1..m-1]$
- Weight w_j assigned to advertiser j as a relevance or quality metric
  - Rank by bid: $w_j = 1$
  - Rank by revenue: $w_j = \alpha_{1j}$
- Advertiser score: $s_j = w_j \cdot b_j$
- Generalised second price auction: agent on slot j pays $\frac{s_{j+1}}{w_j}$
- Static and dynamic aspects