A contribution to Algorithmic Game Theory

Nguyen Kim Thang
Department of Computer Science,
Ecole Polytechnique, France
thang@lix.polytechnique.fr

1. Voronoi Games on Graphs [2]

THE Voronoi game on graphs consists of a graph $G(V, E)$ and k players ($k < n = |V|$). The graph induces a distance between vertices $d : V \times V \to \mathbb{N} \cup \{\infty\}$, which is defined as the minimal number of edges of any connecting path, or infinite if the vertices are disconnected.
The strategy set of each player is V. A strategy profile of k players is a vector $f = (f_1, \ldots, f_k)$ associating each player to a vertex.
For every vertex $v \in V$ — called *customer* — the distance to the closest facility is denoted as $d(v, f) := \min_{f_i} d(v, f_i)$. Customers are assigned in equal fractions to the closest facilities as follows. The strategy profile f defines the generalized partition $\{F_1, \ldots, F_k\}$, where for every player $1 \le i \le k$ and every vertex $v \in V$,

$$F_{i,v} = \begin{cases} 1/|\arg\min_j d(v, f_j)| & \text{if } d(v, f_i) = d(v, f), \\ 0 & \text{otherwise.} \end{cases}$$

We call F_i the *Voronoi cell* of player i. Now the *payoff* of player i is the (fractional) amount of customers assigned to it (see figure 1), that is $p_i := \sum_{v \in V} F_{i,v}$.
A *pure Nash equilibrium* is a strategy profile in which no player has an incentive to unilaterally change his strategy.

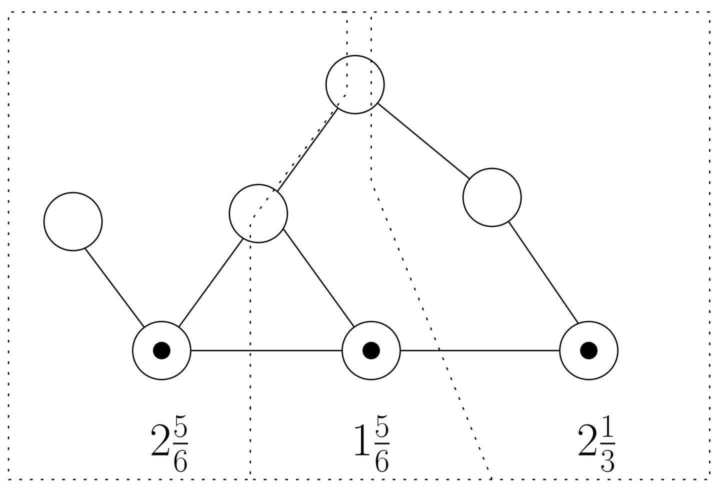

Figure 1: *A strategy profile of a graph (players are dots) and the corresponding payoffs.*

Theorem 1 *Given a graph $G(V, E)$ and a set of k players, deciding the existence of Nash equilibrium for k players on G is $\mathcal{NP}$-complete for arbitrary k, and polynomial for constant k.*

WE introduce a new inefficiency measure, called the *social cost discrepancy*. The social cost discrepancy of the game is defined as the ratio between the cost of the worst Nash equilibrium and the cost of the best one. The idea is that a small social cost discrepancy guarantees that the social costs of Nash equilibria do not differ too much, and measures a degree of choice in the game. Moreover, in some settings it may be unfair to compare the cost of a Nash equilibrium with the optimal cost, which may not be attained by selfish agents.

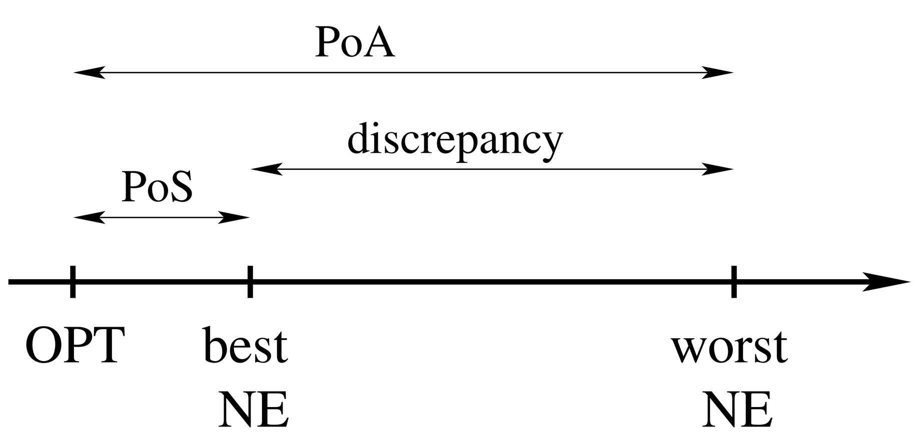

Figure 2: *Illustration of different inefficiency measures, where PoA and PoS stand for the price of anarchy and the price of stability, respectively. Note that, the cost discrepancy is not the ratio between PoA and PoS.*

Theorem 2 *For any connected graph $G(V, E)$ and any number of players k the social cost discrepancy is $O(\sqrt{kn})$ and $\Omega(\sqrt{n/k})$, where $n = |V|$.*

2. Scheduling Games in the Dark [1]

IN a scheduling game, each player owns a job and chooses a machine to execute it. While the *social cost* is the maximal load over all machines (makespan), the *disutility (cost)* of each player is the completion time of its own job. In the game, players may follow selfish strategies to optimize their cost and therefore their behavior does not necessarily lead the game to an equilibrium. Even in the case there is an equilibrium, the social cost may be much larger than the social optimum, and this inefficiency is measured by the price of anarchy (PoA) – the worst ratio between the makespan of an equilibrium and the optimum. *Coordination mechanisms* aim to reduce the price of anarchy by designing scheduling *policies* that specify how jobs assigned to a same machine are to be scheduled.

TYPICALLY, these policies define the schedule according to the processing times as announced by the jobs. One could wonder if there are policies that do not require this knowledge, even no information of jobs but still provide a good PoA. This would make the processing times be private information and naturally turn the policy to be *truthful*, i.e., no job has an incentive to report an incorrect information. We study these so-called *non-clairvoyant* policies. In particular, we study the RANDOM policy that schedules the jobs in a random order without preemption, and the EQUI policy that schedules the jobs in parallel using time-multiplexing, assigning each job an equal fraction of CPU time.

CONSIDER a schedule σ in which there are k jobs with processing times $p_{1j} \le p_{2j} \le \ldots \le p_{kj}$ assigned to machine j, then the cost of job i scheduled in machine j under different policies are the following.

$$c_i = p_{ij} + \frac{1}{2} \sum_{1 \le i' \le k} p_{i'j} \qquad \text{(RANDOM)}$$

$$c_i = p_{1j} + \ldots + p_{i-1,j} + (k - i + 1) p_{ij} \qquad \text{(EQUI)}$$

Theorem 3 *Consider the scheduling game under different policies in different machine environments. Job i is* balanced *if $\max_j p_{ij} / \min_j p_{ij} \le 2$. We have the following.*

1. *For the RANDOM policy on unrelated machines, it is not a potential game for 3 or more machines, but it is a potential game for 2 machines and balanced jobs. On uniform machines with balanced speeds (all jobs are balanced), the RANDOM policy induces Nash equilibrium.*
2. *For the EQUI policy, it is an exact potential game.*

Theorem 4 *In term of inefficiency measured by PoA, EQUI is asymptotically as good as the best* strongly local *policy – policies in which each machine may look at processing times of jobs assigned to it and decides the schedule.*

3. Online Auction with single-minded customers [3]

CONSIDER a production site, that produces some perishable good, at the regular rate, for every time unit, one item is produced. These items have to be delivered immediately to some customer, as they cannot be stored, as for example electricity. Each decision is irrevocable. In this scenario, the *single-minded* customers arrive online. Every customer i arrives at some release time $r_i \in \mathbb{N}$, and announces that he would pay p_i € if he gets k_i items before the deadline d_i, otherwise he pays nothing. In optimization problem, the goal is to maximize the *welfare*, which is the sum of p_i over all satisfied customers.

WE are also interested in *truthful online* mechanism. Roughly speaking, a mechanism consist of two algorithms: (i) *allocation algorithm* that to determine which customers receive items (ii) *payment algorithm* that if a customer is satisfied, how much she has to pay. If q_i be the price that a satisfied player i has to pay then player i's *utility* is $p_i - q_i$. If i is not satisfied then her utility is 0. All customers are rational, they may misreport the values p_i in order to maximize their utility. A truthful (incentive-compatible) mechanism is a mechanism in which players have an incentive to report their true values.

Theorem 5 *For optimization problem, there exists an algorithm that gives a tight bound $\Theta(k/\log k)$-competitive where $k = \max_i k_i$. Moreover, there is a truthful mechanism with the same competitive ratio based on this algorithm.*

References

[1] Christoph DÜRR and NGUYEN Kim Thang. Non-clairvoyant Scheduling Games. Submitted.

[2] Christoph DÜRR and NGUYEN Kim Thang. Nash Equilibria in Voronoi Games on Graphs. In *ESA*, 2007.

[3] Christoph DÜRR, NGUYEN Kim Thang, and Łukasz JEŻ. Auctions with perishable items. Submitted.

35th International Conference on Current Trends in Theory and Practice of Computer Science, Špindlerův Mlýn, Czech Republic, January 24-30, 2009.