

Resource Allocation Algorithms

Lecture 10

Lecture by Dr. Tami Tamir

Typeset by Steven Karas

2018-05-29

Last edited 19:39:10 2018-05-29

Disclaimer These lecture notes are based on the lecture for the course Resource Allocation Algorithms, taught by Dr. Tami Tamir at IDC Herzliyah in the spring semester of 2017/2018. Sections may be based on the lecture slides written by Dr. Tami Tamir.

Admin Note There will be another lecture this thursday, but no lectures for the following two weeks. The final lecture (in three weeks) will be a review session only, as the exam is the week after that. There will be homeworks during the two week break. Some practice questions for the exam will be uploaded during this time as well, however there are no exams from previous years, as this is the first time the course is being taught.

1 Recap

1.1 Network formation games

For every network formation game, there is a pure Nash equilibrium. We have thus far only discussed pure strategies. Best Response Dynamics (BRD) converges on a Nash equilibrium. In a NFG with k players, the cost of anarchy is k . In a NFG with k players, the cost of stability is $H_k = \Theta(\ln k)$. We proved that the problem of finding the Nash equilibrium is PLS-hard¹. The problem of deciding if a NE exists with a given cost is \mathcal{NP} -hard, which we proved by reduction from 3-matching.

1.2 Load games

For every resource there is a cost function that increases with the number of users. The cost of using the resource j if it has load of k users is $C_j(k)$, which is a nondecreasing function. The strategy space of every player is a subset of the resources. For example, a path from s to t .

Every load game has a pure NE, and BRD converges on such a solution. We proved this by using a potential function that is strictly decreasing. Finding the NE is \mathcal{NP} -hard. There is an algorithm based upon a reduction to max-flow for symmetric load games in a network.

If all the price functions are linear, then $\text{PoS} \leq \text{PoA} \leq \frac{4}{3}$. There is a network for which $\text{PoS} = \text{PoA} = \frac{4}{3}$.

1.3 Job scheduling games

Consider each job as a player, where the strategy space is the machines. Each job has a length p_j , and a cost which is the load of the machine it's on. LPT provides a stable matching. The price of anarchy is $2 - \frac{2}{m+1}$.

1.4 A practice problem

n jobs are scheduled on two machines. The cost of each job is the weighted load on its machine.

1. Give an example of an input and the steps of a BRD in which a job makes more than one improving step
2. Prove that the weighted load difference $|L_1 - L_2|$ is strictly smaller in each step of BRD.

¹between \mathcal{P} and \mathcal{NP}

3. Prove that if for every step of the BRD the largest job that wants to move does so, then every job moves at most once.

1.4.1 Example

Given three jobs with weights 2, 6, and 5. Assume the initial schedule is all jobs on M2. In the first step, job 2 moves to M1. In the next step, job 6 moves to M1. In the next step, job 2 moves back to M2.

1.4.2 Decreasing load

Denote $\Delta = |L_1 - L_2|$. Assume w.o.l.g. that $L_1 > L_2$ before a job with weight x moves. If the job chose to move, then $L_1 > L_2 + x$; thus $\Delta > x$. Denote as L'_1, L'_2 as the loads after moving. Let $\Delta' = |L'_1 - L'_2|$. Note that $L'_1 = L_1 - x$, and $L'_2 = L_2 + x$. The remainder of the proof is algebraic and trivial.

1.4.3 Quick Convergence

Assume for contradiction that there is a job that is interested in moving twice with weight p . Consider the first such job w.o.l.g. that wants to move from M_1 to M_2 and then later back from M_2 to M_1 . Denote L_1, L'_1, L_2, L'_2 as the loads before the first move and before the second move. Note that because the job wants to move, $L_1 > L_2$ and $L'_2 > L'_1$... I didn't get the rest of the proof, but it will be uploaded to the course website.

2 Job Scheduling

2.1 Strong Equilibrium

From Aumann 1959, a strong equilibrium is one where no coalition can deviate and strictly improve the utility of all of its members. A k -Nash equilibrium is a NE that cannot be improved by a coalition of size less than or equal to k . This separates the effects of selfishness from lack of coordination, and may predict rational behavior better. Most games do not allow Strong Equilibriums, however in job scheduling games, there is always a strong equilibrium, even for unrelated machines.

Proof of existence Given a schedule, characterize it by a vector of decreasingly sorted² loads: (L_1, \dots, L_m) . For two vectors $X = (x_1, \dots, x_m), Y = (y_1, \dots, y_m)$, we say that X is lexicographically smaller than Y , denoted as $X \prec Y$ if there exists some i such that $x_i < y_i$, and for all $j < i$, $x_j = y_j$. Let p be a profile of a schedule with the lexicographically minimal load vector. We claim that p is a strong equilibrium.

It is trivial to show that such a schedule is a Nash equilibrium because any step that improves the schedule would produce a lexicographically smaller load vector. Assume for contradiction that there is a coalition in p . Let Γ be the coalition with the minimal number of jobs. Let $M(\Gamma)$ be the set of machines that the jobs in Γ are scheduled on them. First, we will show that for any step that improves Γ , at least one job leaves, and at least one jobs joins the machines in $M(\Gamma)$. Let $M_a \in M(\Gamma)$. If there is no job that left M_a , then a job must have joined M_a , which would imply that this job did not need the others in the coalition). However, this contradicts that this is a Nash equilibrium. If there is no job that joined M_a , then denote as j the job that left M_a . This implies that $\Gamma \setminus \{j\}$ is a smaller coalition.

Consider the machine with the highest load $M_i \in M(\Gamma)$. Denote the load of M_i before and after the movement of the coalition as L_i and L'_i . As the movement is beneficial for the coalition, it follows that $L'_i < L_i$. Therefore the load of the most loaded machine in $M(\Gamma)$ decreased. As the move is beneficial for the entire coalition (by definition), then all the machines have load $\leq L_1$.

NE approximation of SE? This material will not be on the exam for lack of time to cover it. Is there a bound on the improvement ratio for all members, or for a single member? Is there a bound on the damage ratio for non-coalition members?

²Such that $L_1 \geq L_2 \geq \dots \geq L_m$; that is to say that L_1 is the load of the most loaded machine, etc.

3 Network Congestion Games

slides 28 onwards from agt2.

3.1 Weighted Unsplittable Routing

Let w_i denote the weight of player i . The loads and costs are proportional to the weights.

Consider as a specific case the graph located on slide 31. This instance does not have a Nash equilibrium. The proof follows from exhaustive enumeration of the entire strategy space (three possible weights among 4 player strategies).

This shows that there may not be a NE in a weighted congestion game.

4 Mechanism Design

A field in economics and game theory that applies engineering principles to designing protocols for strategic decision making with purely rational actors. We will show various applications of this, such as private value versions of shortest path, MST, auctions, etc.

4.1 Set System Auctions

Shortest Path Auction In which we try to form a path from a source to a target in a graph.

Each player owns an edge in the graph. The graph, the source, the target, and the complete protocol are publicly known. The cost c_e that each player incurs to have their edge e included in the solution is only known to that player.

Protocol Each player submits sealed bids b_e . The auctioneer selects a path P from s to t . For each edge e in P , the player who owns the edge e incurs the cost c_e , but also gets a payment $p_e(b) \geq b_e$.

Spanning Tree Auction As above, but the auctioneer selects a spanning tree and pays out.

Generalization details can be found on slide 10.

4.1.1 Vickrey Auctions

Proposed by William Vickrey in 1961[2].

Agents competing to purchase a single item. Agents submit bids, and the winner is the person who bid the most, but only pays the second highest bid.

There is a proof of the truthfulness of this can be found on slides 17-18.

4.1.2 Vickrey-Clarke-Graves Mechanism

Selects a min-cost feasible set for set selection auctions.

For $e \in \text{OPT}$:

$$p(e, c) = \text{OPT}(c(e) \rightarrow \infty) - \text{OPT}(c(e) \rightarrow 0)$$

For all $e \notin \text{OPT}$:

$$p(e, c) = 0$$

Applied to MST: Define the payoff of edge e when it is in the selected solution to be the maximum cost e can bid, which is the cost of an MST that doesn't use e , less the cost of the other edges in the MST.

$$p(e) = \text{MST}(G - e) - \text{MST}(G \mid c(e) = 0)$$

The incentive for the edges is such that they don't need to know what the maximal value, as they are guaranteed to get that at least, so the incentive is to state their real value.

The proof of this is left as an exercise for the student and will likely be in a homework.

References

- [1] Michael L Pinedo. *Scheduling: theory, algorithms, and systems*. Springer, 2016.
- [2] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961.