AGT Summative Assignment – Individual Component Report

wcrr51

2021

Exercise 1. Consider the following instance of the load balancing game where the number of tasks is equal to the number of machines, and in particular we have:

- m identical machines $M_1, M_2, \ldots, M_m$ (all of speed 1),
- m identical tasks $w_1 = w_2 = \cdots = w_m = 1$.

Consider also the mixed strategy profile A where each of the tasks is assigned to all machines equiprobably (i.e. with probability $1/m$).

(a) Calculate the ratio $cost(A)/cost(OPT)$ in the special case where $m = 2$. **[3 marks]**

Trivially, makespans of 1 and 2 have 2 assignments each. Hence, for $m = 2$, there are a total of $2^2 = 4$ possible assignments. These assignments are shown in Table 1.

	M_1	M_2	Makespan
1	1, 2	-	2
2	-	1, 2	2
3	1	2	1
4	2	1	1

Table 1: Task-machine assignments for $m = 2$ in Exercise 1

From this, $cost(A)$ can be calculated with

$$cost(A) = E[cost(B)] = \sum_{i=1}^{m} P(cost(B) = i) \cdot i \tag{1}$$

Since there is capacity for one machine per task, and this would be the optimal assignment for any positive m, hence, it holds true that

$$\forall m > 0, cost(OPT) = 1 \tag{2}$$

For $m = 2$, using (1), $cost(A) = E[cost(B)] = \frac{1}{4}(1 \cdot 2 + 2 \cdot 2) = \frac{6}{4} = \frac{3}{2} = 1.5$.
Combining this with (2), the ratio $cost(A)/cost(OPT)$ for $m = 2$ is $\frac{3}{2}/1 = \frac{3}{2} = 1.5$

(b) Calculate the ratio $cost(A)/cost(OPT)$ in the special case where $m = 3$. **[3 marks]**

For a makespan of 1, there are ${}^3P_3 = 3! = 6$ assignments, for 2, there are $3 \cdot {}^3P_2 = 18$ assignments and for 3, there are trivially 3 assignments. Hence, for $m = 3$ there are a total of $3^3 = 27$ possible assignments. These assignments are shown in Table 2.

For $m = 3$, using (1), $cost(A) = E[cost(B)] = \frac{1}{27}(1 \cdot 3 + 2 \cdot 18 + 3 \cdot 3) = \frac{51}{27} = \frac{17}{9} \approx 1.89$.
Combining this with (2), the ratio $cost(A)/cost(OPT)$ for $m = 3$ is $\frac{17}{9}/1 = \frac{17}{9} \approx 1.89$

(c) Discuss what this ratio is for arbitrary m. What does this imply about the Price of Anarchy on identical machines for mixed Nash equilibria? **[5 marks]**

As (2) holds true for all $m > 0$, the denominator of the $cost(A)/cost(OPT)$ ratio is always one. This means that, for this instance of the load balancing game, $cost(A)$ determines what the fraction will be.

There are m^m possible task to machine assignments, distributed over m makespan values (from 1 to m). Through equiprobable assignment of tasks to machines, let p_i be the probability of an assignment with makespan $i = 1, 2, \ldots, m$ where $\sum_{i=1}^{m} p_i = 1$, given by

$$p_i = \frac{c_i}{m^m}$$

	M_1	M_2	M_3	Makespan
1	1, 2, 3	-	-	3
2	-	1, 2, 3	-	3
3	-	-	1, 2, 3	3
4	1	2, 3	-	2
5	1	-	2, 3	2
6	-	1	2, 3	2
7	2, 3	1	-	2
8	2, 3	-	1	2
9	-	2, 3	1	2
10	2	1, 3	-	2
11	2	-	1, 3	2
12	-	2	1, 3	2
13	1, 3	2	-	2
14	1, 3	-	2	2
15	-	1, 3	2	2
16	3	1, 2	-	2
17	3	-	1, 2	2
18	-	3	1, 2	2
19	1, 2	3	-	2
20	1, 2	-	3	2
21	-	1, 2	3	2
22	1	2	3	1
23	2	1	3	1
24	2	3	1	1
25	3	2	1	1
26	3	1	2	1
27	1	3	2	1

Table 2: Task-machine assignments for $m = 3$ in Exercise 1

Where c_i is the number of combinations of task to machine assignments with a makespan i (for example, for $m = 3$, and makespans $i = 1, 2, 3$, $c_1 = 6, c_2 = 18, c_3 = 3$, and hence $p_1 = \frac{6}{27} = \frac{2}{9}, p_2 = \frac{2}{3}, p_3 = \frac{1}{9}$).
This would, for arbitrary m, yield

$$cost(A) = \sum_{i=1}^{m} \frac{i \cdot c_i}{m^m}$$

Note that $\sum_{i=1}^{m} c_i = \implies \forall i > 1, c_i < m^m$, this means that $cost(A)$ is bound to lie within the makespan distribution (between 1 and m), implying the Price of Anarchy for identical machines is relatively low. Furthermore, this is backed up by a known theorem (Slide 50 of Week 5) which states that $\frac{cost(Q)}{cost(OPT)} = O\left(\frac{\log m}{\log \log m}\right) => \texttt{PoA} = \Theta\left(\frac{\log m}{\log \log m}\right)$ for any Nash equilibrium strategy profile Q on identical machines. Because of this, it also implies that the price of anarchy for mixed Nash equilibria on identical machines is quite small, and hence, provides a lower bound on the price of anarchy across all load balancing games.

Exercise 2. We consider a second-price sealed-bid auction where there are n bidders who bid as follows:

- Bidders 1 up to $n-1$ bid either 1 dollar or $r > 1$ dollars equiprobably and independently of the rest.
- Bidder n bids h dollars, where $h > r$.

The seller's expected revenue R is the expectation of the second highest value.

(a) What is the value that R is approaching when n is very large? **[1 marks]**

For large n, R approaches r.

(b) Justify your answer by taking the limit. **[9 marks]**

Trivially and by the definition, R must be less than h.

Let X be a binomially distributed random variable representing the number of times r is chosen (instead of 1) from a set of $n-1$ independent trials (representing independent bidders 1 to $n-1$), each of probability $\frac{1}{2}$.

$$X \sim B\left(n-1, \frac{1}{2}\right) \tag{3}$$

$P(X=0)$ is the probability that r is chosen 0 times across the $n-1$ independent trials.

$$\begin{aligned} P(X=0) &= {}^{n-1}C_0 \cdot \frac{1}{2}^{0} \cdot \frac{1}{2}^{(n-1)-0} = \frac{1}{2}^{n-1} = 2^{-(n-1)} = 2^{1-n} = 2 \cdot 2^{-n} \\ \implies \lim_{n->\infty} P(X=0) &= \lim_{n->\infty} \left(2 \cdot 2^{-n}\right) = 2 \cdot \lim_{n->\infty} 2^{-n} = 2 \cdot 0 = 0 \end{aligned} \tag{4}$$

This means that as n increases, the probability of r not being chosen approaches zero. Hence, the larger the value of n, the more likely an r will be chosen, and thus the more likely $R = r$.

Exercise 3. Mary and Alice are buying items for Sunday lunch. Mary buys either chicken (C) or beef (B) for the main course and Alice buys either juice (J) or wine (W). Both people prefer wine with beef and juice with chicken. The opposite alternatives are equally displeasing. However, Mary prefers beef over chicken, while Alice prefers chicken over beef.

We assume that Mary buys first and then tells Alice what she bought, so when Alice makes her decision, she knows if the main course is beef or chicken.

(a) Express the above preferences as payoffs by using numbers
(e.g. $u_M(B,W) = 2$, $u_A(B,W) = \ldots$ etc.) **[2 marks]**

As Mary prefers B over C, and Alice prefers C over B, $u_M(B,W) > u_A(B,W)$ and $u_M(C,J) < u_A(C,J)$. As both prefer B with W and C with J, $u_i(B,W) > 0, u_i(C,J) > 0 : i \in \{M,A\}$ As both are equally displeased by B with J and C with W, $u_i(B,J) = u_i(C,W), u_i(B,J) < u_i(B,W), u_i(C,W) < u_i(B,J) : i \in \{M,A\}$

The following assignment satisfies these constraints:

$$\begin{aligned} u_M(B,W) = 2, u_A(B,W) = 1, \\ u_M(B,J) = 0, u_A(B,J) = 0, \\ u_M(C,W) = 0, u_A(C,W) = 0, \\ u_M(C,J) = 1, u_A(C,J) = 2 \end{aligned} \tag{5}$$

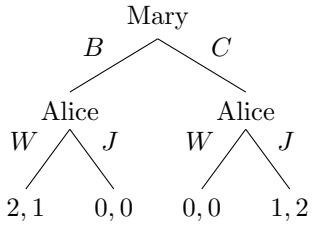
(b) Write down a bimatrix game with Mary as the row player and Alice as the column player, using your chosen payoffs. **[4 marks]**

		Alice			
		J		W	
Mary	C	1	2	0	0
	B	0	0	2	1

Table 3: Bimatrix game representation of the scenario in Exercise 3

(c) Write down a game tree representing this game as an extended game. **[4 marks]**

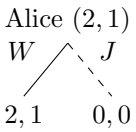
As Mary buys first, she is at the root of the tree, the two outward edges represent the choice between B and C. Whether B or C is chosen, Alice is the next node, these each have two edges representing the choice between W and J. Finally, the leaves represent the payoffs for each player from the corresponding actions taken from the root to the leaf.



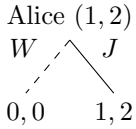
(d) Find a solution for the extended game using backward induction.
Describe your steps. **[5 marks]**

For this solution, an edge with a solid line represents the action that is chosen.

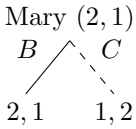
Firstly consider the left subgraph (taking action B), Alice may either choose W with reward 1 or J with reward 0. As $1 > 0$, Alice chooses W and her node takes the payoff value $2, 1$.



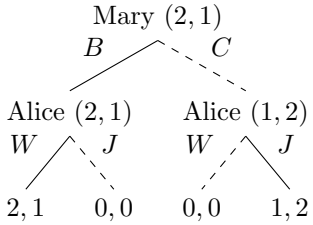
Next, return to the root and consider its right subgraph (taking action C), Alice may either choose W with reward 0 or J with reward 2. As $0 < 2$, Alice chooses W and her node takes the payoff value $1, 2$.



Finally, consider the root. Mary may either choose B with reward 2 or C with reward 1. As $2 > 1$, Alice chooses B.



The full annotated graph is as follows.



Hence, by backwards induction, Mary gets a higher payoff than Alice and they have Beef with Wine for Sunday lunch.

Exercise 4. We consider a (matching) market of k sellers and k buyers, where k is an integer, $k > 0$. Each seller sells an item and the prices of the items are initially all zero. Buyer i has valuation $k - i + 1$ for the first item and valuation 0 for every other item, as shown in the following diagram.

Buyers	Valuations (for items 1 to k)
x_1	$k, \quad 0, \quad \ldots, \quad 0$
x_2	$k-1, \quad 0, \quad \ldots, \quad 0$
$\vdots$	$\vdots$
x_k	$1, \quad 0, \quad \ldots, \quad 0$

The sellers find the market-clearing prices using the procedure discussed in the lectures.

(a) What are the prices of the sellers' items (1^{st} item, 2^{nd} item, ..., k^{th} item) when the market clears? Which buyer gets the 1^{st} item and at what price? **[3 marks]**

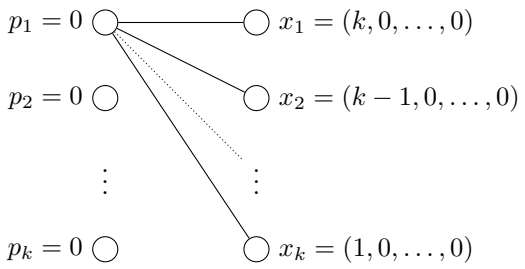
The 1^{st} item is sold at a price of $k - 1$. The 2^{nd} to k^{th} items are sold at a price of 0.
Buyer x_1 gets the 1^{st} item at a price of $k - 1$.

(b) Justify your answers to (a). **[6 marks]**

Let Algorithm 1 be the procedure for finding market-clearing prices as shown in the lectures, Y be the set of items where item y_i is sold at $p_i, 1 \leq i \leq k$, X be the set of all buyers $x_j, 1 \leq j \leq k$, and $x_{j,i}$ be the valuation of item i by buyer j.

Theorem 0.1. *For k items and k buyers, the matching market with the above setup results in the market-clearing prices $(k-1, 0, \ldots, 0)$ for items $1, 2, ..., k$ using Algorithm 1.*

Proof. Direct proof on k.
Applying Algorithm 1, the sellers' prices are initialised $p_i = 0$ for $1 \leq i \leq k$ (step 1). The preferred-seller graph is constructed as follows.
Since all buyers have a non-zero valuation of the first item (and zero of the rest), and all items are initially sold for zero, the only preferred seller for each buyer is item 1. Thus, an edge (p_1, x_j) is formed for $1 \leq j \leq k$ (step 2).



This clearly does not form a perfect matching, so a constricted set S of buyers is found (step 4). In this case, the neighbourhood of the constricted set $N(S)$ is p_1 no matter what constricted set S is chosen (as p_1 is fully connected to the buyers and p_i for $2 \leq i \leq k$ are disconnected).

Lemma 0.2. *For the above setup, the neighbourhood $N(S)$ of any constricted set of buyers S (at step 4 in Algorithm 1) is always $\{p_1\}$.*

Proof. Induction on the iterations of Algorithm 1 (m).
Base case:
Let $m = 1$.
For any k, the initial bipartite preferred-seller graph is the same as above. By inspection, $N(S) = y_1$, hence, for one iteration of Algorithm 1, the lemma holds.
Inductive step:
Now assume the lemma holds true for the $m = r^{th}$ iteration.
For step 2 of Algorithm 1, if there is a perfect matching, there cannot be a constricted set of buyers - in which case the algorithm terminates at step 3. Otherwise, at step 4, p_1 is the only vertex in $N(S)$ (by assumption of the truth of the lemma).

p_1 is incremented by one in step 5. Given the lemma is assumed true up until this point, p_1 is the only price that has ever been incremented. This means that, before step 5 of iteration r, $p_1 = r - 1$ and after, $p_1 = r$, and $p_i = 0$ for $2 \leq i \leq k$.

As there exists a zero price, no normalisation takes place in step 6 of Algorithm 1. This means that going into iteration $m = r + 1$, p_1 holds the value r.

For step 2 of iteration $r + 1$, the preferred-seller graph is reconstructed. Again, if there is a perfect matching, the algorithm terminates at step 3.

Observe the following three cases for a buyer x_j.

Case 1:
When $p_1 < x_{j,1}$, x_j is only interested in (and connects to) the first item. This is because buying y_1 gives the highest payoff compared to a payoff of zero from all the other items ($x_{j,1} - p_1 > 0$).

Case 2:
When $p_1 = x_{j,1}$, x_j is interested in (and connects to) all items (including the first). This is because buying the item y_1 has a payoff of zero - the same as all the other items ($x_{j,1} - p_1 = 0$).

Case 3:
When $p_1 > x_{j,1}$, x_j is interested in (and connects to) all items except the first. This is because the payoff for buying y_1 becomes negative, so they favour the next highest payoff which is zero for all the other items ($x_{j,1} - p_1 < 0$).

Firstly consider the situation where x_1 finds its self in Case 1. This means means that $p_1 < x_{1,1} \implies r+1 < k$. Thus, $\exists j : 2 \leq j \leq k \implies r+1 = x_{j,1}$. The buyers can now be partitioned into two subsets. First, a subset who can win y_1, defined by $I = X - E = \{x_j : r+1 \geq x_{j,1}, 2 \leq j \leq k\}$. Second, a subset who can't win y_1, defined by $E = \{x_j : r + 1 < x_{j,1}, 2 \leq j \leq k\}$. The first subset of length a consists of the buyers in Case 1, the second subset of length $k - a$ consists of the buyers in Case 3. As it is established that buyers in Case 3 are fully connected to all items except the first, they form the complete bipartite graph $K_{k-1,k-a}$. A subgraph of this is Ka, a since $n - a \leq k - 1$ which has a perfect matching.

Suppose that x_1 found itself in Case 3 on the $r + 1^{th}$ iteration, $r + 1 > x_{1,1}$. By extension, all other buyers must also be in Case 3 – overall $r + 1 > x_{j,1}$ for $1 \leq j \leq k$. This means that no buyers are interested in item y_1 and hence no perfect matching can exist. Because of this, $S = X$ and $N(S) = Y - \{y_1\}$.

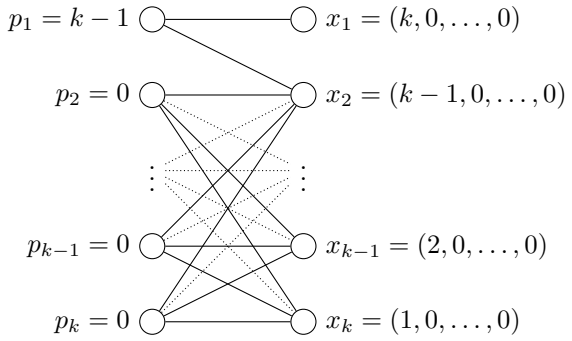
At step 6 in Algorithm 1, $p_i = p_i + 1$ for $2 \leq i \leq k$ giving $\min p_i = 1$. This means that on the next iteration, all selling prices are reduced by 1, returning it to the same state as in iteration r.

As this results in an infinite loop, the algorithm should never enter the state where $p_i > x_{1,1}$. For that to happen it would need to pass through the situation where x_1 finds its self in Case 2 ($p_1 = r + 1 = x_{1,1}$). This would also mean that all other buyers are in Case 3 ($r + 1 > x_{j,1}$ for $2 \leq j \leq k$). This situation would also not occur because a perfect matching would have been found, terminating the algorithm at step 3.

Because of this, continuation to step 4 of the algorithm means that x_1 must be (and other consecutive buyers might be) in Case 1, there is exactly one buyer in Case 2, and the rest are in Case 3. As such, item y_1 must be connected to all buyers in Case 1 (including at minimum x_1) and the one buyer in Case 2. Trivially, all buyers in Case 3 (those in the second subset) form a complete bipartite graph with all items except for the first. Furthermore, the second subset (buyers in Case 1 and the buyer in Case 2) must have a size at least 2. If the size is 2, there would exist a perfect matching and so the algorithm would have terminated. Hence, since the size must be greater than 2, there is a subset S of at least 2 buyers in Case 1 which are only connected to item y_1. Finally, this shows that, for iteration $m = r + 1$, S (and any subset of $S, |S| > 1$) forms the only constricted set with neighbourhood $N(S) = \{y_1\}$. □

Using Lemma 0.2, at step 5, Algorithm 1 only ever increments p_1 (because y_1 is the only item in $N(S)$ and S, and its subsets, form the only constricted set). Combining this with the additional remarks in Lemma 0.2, Algorithm 1 must therefore increment p_1 at each iteration until x_2 switches to Case 2, leaving a size of 2 for the first set (from Lemma 0.2), making it no longer a constricted set.

Because p_1 is incremented by one each iteration, all other item prices remain 0, and $x_{1,1} > p_1, x_{2,1} = p_1$, it is necessary that $p_1 = k - 1$ and $p_i = 0, 0 < i \leq k$. This will result in the preferred-seller graph below.



Removing the vertices for seller y_1 and buyer x_1 yields the complete bipartite graph $K_{k-1,k-1}$. Hence, at step 2, Algorithm 1 now terminates because there is a clear parallel matching between x_i and y_i for $1 \leq i \leq k$. Indeed, the final item prices are $(y_1, y_2, \cdots, y_k) = (k - 1, 0, \cdots, 0)$ as required.

□

(c) Which kind of auction does the construction of market-clearing prices procedure implement in this case? **[3 marks]**

As the winner of item 1 is the buyer (x_1 in this case) who values it the highest. They pay the valuation of item 1 by the second highest buyer ($k - 1$ in this case). Thus, the construction of market-clearing prices implements a second-price (Vickrey) auction for item 1 in this case.