Individual Exercises

Benjamin Russell, *fdmw97*

March 27, 2021

Exercise 1

a).

- We have 2 machines M_1, M_2 with speeds $S_1 = S_2 = 1$ and 2 tasks $1, 2$ with weights $w_1 = w_2 = 1$.
- Let $C : [2] \to [2]$ be an assignment of task i to a machine M_j. We have strategy profile A let $A_i^j = P(C(i) = M_j)$ i.e. the probability task i is assigned to machine M_j. For all $i, j \in [2]$ $A_i^j = \frac{1}{2}$
- It is trivial to see that under these definitions $cost(OPT) = 1$ as we assign each task to its own machine giving a makespan of 1 any other assignment would have > 1 task on one of the machines resulting in $\max_{j \in [2]}(l_j) > 1$ in this particular case 2.
- $cost(A) = E[cost(C)]$ i.e. the cost of A is equal to the random assignment C under A.
- $cost(C) \in \{1, 2\}$ i.e. $cost(C)$ is a random variable.
- There are $2^2 = 4$ different possible assignments, 2 with a makespan of 1 and 2 with a makespan of 2. Therefore:

$$E[cost(C)] = \frac{2}{4} \times 1 + \frac{2}{4} \times 2 = 1.5$$
$$\frac{cost(A)}{cost(OPT)} = \frac{1.5}{1} = 1.5$$

b).

- We have 3 machines M_1, M_2, M_3 with speeds $S_1 = S_2 = S_3 = 1$ and 3 tasks $1, 2, 3$ with weights $w_1 = w_2 = w_3 = 1$.
- Let $C : [3] \to [3]$ be an assignment of task i to a machine M_j. We have strategy profile A let $A_i^j = P(C(i) = M_j)$ i.e. the probability task i is assigned to machine M_j. For all $i, j \in [3]$ $A_i^j = \frac{1}{3}$
- It is trivial to see that under these definitions $cost(OPT) = 1$ as we assign each task i to its own machine M_j giving a makespan of 1, as any other assignment would have > 1 task on one machine resulting in $\max_{j \in [2]}(l_j) > 1$ in this case either 2 or 3.
- $cost(A) = E[cost(C)]$ i.e. the cost of A is equal to the random assignment of C under strategy profile A.
- $cost(C)$ is a random variable such that $cost(C) \in \{1, 2, 3\}$
- There are $3^3 = 27$ different possible assignments, 6 with a makespan of 1, 18 with a makespan of 2, and 4 with a makespan of 3. Therefore:

$$E[cost(C)] = \frac{1}{27}(6 \times 1 + 18 \times 2 + 3 \times 3) = \frac{17}{9}$$
$$\frac{cost(A)}{cost(OPT)} = \frac{\frac{17}{9}}{1} = \frac{17}{9} = 1.\dot{8}$$

c).

- For an arbitrary m we have machines M_j where $j \in [m]$ and speeds $S_j = 1$ where $j \in [m]$. We also have tasks $i \in [m]$ with weights $w_i = 1$.
- Let $C : [m] \to [m]$ be an assignment of task i to machine M_j. Our strategy profile A assigns a task i to each machine equiprobably so let $A_i^j = P(C(i) = M_j)$ i.e. the probability that task i is assigned to machine M_j. For all $i, j \in [m]$ $A_i^j = \frac{1}{m}$
- $E[l_j] = \sum_{i=1}^{m} w_i A_i^j = \sum_{i=1}^{m} 1 \times \frac{1}{m} = m \cdot \frac{1}{m} = 1$
- Since all machines are identical and all tasks are identical $\forall i \forall j \in [m]$

$$c_i^j = E[l_j] + (1 - A_i^j)w_i = 1 + \frac{m-1}{m} \cdot 1 = \frac{2m-1}{m}$$

  As this is the same for all tasks and machines under A, A must be a Nash Equilibrium.
- Once again $cost(OPT)$ is trivially 1, as each task can be assigned to an individual machine giving a makespan of 1, since the number of tasks and machines is equal any other assignment would have at least 1 machine with more than 1 task assigned to it giving a makespan > 1.
- $cost(A) = E[cost(C)]$ i.e. the cost of A is equal to the random assignment of C under A.
- As all weights $w_i = 1$ $E[cost(C)]$ is analogous to the expected maximum load of a balls-and-bins problem .
- When the number of balls and bins are equal like in this case and the balls are uniformly distributed as they are under A and $cost(OPT) = 1$ then:

$$cost(A) = \Theta\left(\frac{\log m}{\log \log m}\right)$$

  As shown in [1].
- This implies that as the number of machines and tasks increases, so does the Price of Anarchy on identical machines for mixed Nash Equilibria – in a sub-logarithmic manner.

Exercise 2

a). When n is very large R is approaching r

b). Let B be the set of all bids i.e.

$$B = \{b_1, b_2, ..., b_{n-1}, b_n\}$$

Where b_i is the bid of player $i \in [1, n]$. The maximum bid $\max(B)$ is b_n as $b_n > r > 1$ by the problem definition. Therefore $R = E[\max(B \setminus \{b_n\})]$ i.e. the expected maximum of all bids less than b_n.

$$\text{Let } B' = B \setminus \{b_n\}$$

So $R = E[\max(B')]$, if any $b \in B'$ bids r then $\max(B')$ is necessarily r as $r > 1$ as per the problem definition, else $\max(B')$ is 1.
We can now model B' using a binomial distribution $B(n-1, 0.5)$ where a success is bidder b_i bidding r. Let $X \sim B(n-1, 0.5)$ be a discrete random variable representing the number of successes in $n-1$ bidders.

$$\begin{aligned} P(X \geq 1) &= 1 - P(X = 0) \\ &= 1 - \binom{n-1}{0} \times 0.5^0 \times 0.5^{n-1} \\ &= 1 - 0.5^{n-1} \end{aligned}$$

This gives the probability that at least 1 bidder from B' bids r therefore by taking the limit to infinity we can determine R for very large n.

$$\lim_{n\to\infty} P(x \geq 1) = \lim_{n\to\infty} 1 - 0.5^{n-1} = 1 - 0 = 1$$

This means as n approaches ∞, $P(\max(B') = r) = 1$ and $P(\max(B') = 1) = 0$ therefore

$$E[\max(B')] = 1 \times r + 0 \times 1 = r$$

so $R = E[\max(B')] = r$

Exercise 3

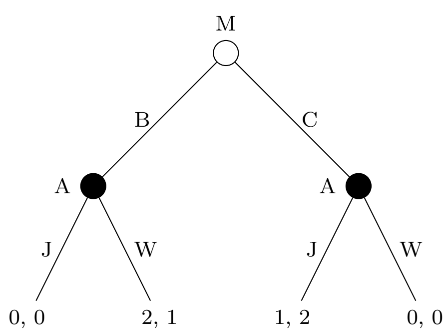
a). Mary's preferences $= u_M$:
$u_M(B, W) = 2, u_M(C, J) = 1, u_M(C, W) = 0, u_M(B, J) = 0$

Alice's preferences $= u_A$:
$u_A(C, J) = 2, u_A(B, W) = 1, u_A(C, W) = 0, u_A(B, J) = 0$

b).

		Alice	
		W	J
Mary	C	$(0, 0)$	$(1, 2)$
	B	$(2, 1)$	$(0, 0)$

c).



d). Solution $= (B, W)$
Using backwards induction first we consider the subgames of length 1, starting with the subgame following B. In this game Alice's optimal choice is W giving a payoff of $1 > 0$. Next we consider the subgame following C. In this game Alice's optimal choice is J giving a payoff of $2 > 0$. Now both subgames of length 1 have been studied we can study the subgames of length 2 of which there is only 1 the full game. Given the optimal actions in the subgames of 1 Mary choosing B would yield her a payoff of 2 and choosing Cwould yield a payoff of 1, therefore Mary's optimal action is choosing B. No subgames exist of length 3 therefore we are done producing the strategy pair (B, WJ) and solution of (B, W).

Exercise 4

a). Item 1 has final price of $k - 1$ all other items $\{2, ..., k\}$ have prices of 0. Buyer x_i buys item 1 at a price of $k - 1$ for a payoff of $k - (k - 1) = 1$.

b).
- For all items indexed $i \in [2, k]$ the payoff for x_i will always be 0 as the valuation is 0 and so is the asking price.
- So long as > 1 buyer x_i has a > 0 payoff from item 1 there will be a constricted set S with $N(S) = \{1\}$ as all other items will give payoff of 0 rather than > 0 from item 1. This results in the asking price of item 1 being increased by 1.

- As there are k buyers with valuations for item 1 of $k-i+1$ the maximum valuation is x_i with k and the second highest is x_2 with $k-1$. So by the time item 1's price has increased to $k-2$ all other buyers have either a negative payoff from item 1 or 0 in the case of x_3. Therefore all $x_i \notin \{x_1, x_2\}$ can be assigned to an item $\neq 1$, of which there must be enough as there are $k-1$ such items and $k-2$ buyers $\notin \{x_1, x_2\}$. Now item 1 increases its price to $k-1$ as x_1 and x_2 still form a constricted set S. After this increase x_2's payoff is now 0, so can be assigned to an item $\neq 1$. Now there is no longer a constricted set S so the final price of item 1 is $k-1$ and all other items are 0.

c). In this case the construction of market clearing prices models a second-price sealed-bid auction.

References

[1] *Algorithmic Game Theory.* Cambridge University Press, 2007, ch. 20. DOI: 10.1017/CBO9780511800481.