Stav Yosef, 316298876

Advanced Algorithms - Homework 5

Question 1

Players possible path

Player 1: $\{(s,A)\},\{(s,V),(V,A)\}$
Player 2: $\{(s,V),(V,B)\}$
Player 3: $\{(s,C)\},\{(s,V),(V,C)\}$

Profiles

$R_1 = \{(s,A)\} + \{(s,V),(V,B)\} + \{(s,C)\} = 3 + (x+6) + 2 = x + 11$

$R_2 = \{(s,A)\} + \{(s,V),(V,B)\} + \{(s,V),(V,C)\} = 3 + \left(\frac{x}{2}+6\right) + \left(\frac{x}{2}+1\right) = x + 10$

$\mathrm{R}_3 = \{(\mathrm{s},\mathrm{V}),(\mathrm{V},\mathrm{A})\} + \{(\mathrm{s},\mathrm{V}),(\mathrm{V},\mathrm{B})\} + \{(\mathrm{s},\mathrm{C})\} = \left(\frac{\mathrm{x}}{2}+1\right) + \left(\frac{\mathrm{x}}{2}+6\right) + 2 = \mathrm{x}+9$

$R_4 = \{(s,V),(V,A)\} + \{(s,V),(V,B)\} + \{(s,V),(V,C)\} =$
$= \left(\frac{x}{3}+1\right) + \left(\frac{x}{3}+6\right) + \left(\frac{x}{3}+1\right) = \boldsymbol{x+8}$

Social Optimum - $R_4 = x + 8$

$(\textbf{Nash Equilibrium}, \textbf{PoS}, \textbf{PoA})$

1. $\mathrm{x} > 4$ - $\left(\mathrm{R}_1, \frac{x+11}{x+8}, \frac{x+11}{x+8}\right)$
2. $\mathrm{x} = 4$ - $\left((\mathrm{R}_1,\mathrm{R}_3), \frac{x+9}{x+8} = \frac{13}{12}, \frac{x+11}{x+8} = \frac{5}{4}\right)$
3. $4 > \mathrm{x} > 3$ - $\left(\mathrm{R}_3, \frac{x+9}{x+8}, \frac{x+9}{x+8}\right)$
4. $\mathrm{x} = 3$ - $\mathrm{R}_3, \mathrm{R}_4 \left((\mathrm{R}_3,\mathrm{R}_4), \frac{x+8}{x+8} = 1, \frac{x+9}{x+8} = \frac{12}{11}\right)$
5. $3 > \mathrm{x} \geq 0$ - $\left(\mathrm{R}_4, \frac{x+8}{x+8} = 1, \frac{x+8}{x+8} = 1\right)$

Stav Yosef, 316298876

Question 2

1. In the profile p_x, the cost of the strategy $\{sa, at\}$ is $3x + 6$, and the cost of the strategy $\{st\}$ is $3(10\text{-}x)$. The strategy P_x is a NE if (players on top path are stable) $3x + 6 \leq 3(11\text{-}x)$ and (players on lower path are stable) $3(10\text{-}x) \leq 3x + 9$ That is P_4.

2. We need to find a function of social optimum then derive it, in order to find the global minimum. (we can make sure that is minimum by checking $x = 3$ and $x = 5$)

$SC(x) = 3(10\text{-}x)^2 + (6 + x)^2 + 2 \cdot x^2 = 6x^2\text{-}48x + 336$

$SC'(x) = 12x\text{-}48$

$0 = 12x\text{-}48, x = 4$

$SC(4) = 240$

3. There is only one NE and is P_4 → $PoS = \frac{NE_{best}}{SC} = \frac{240}{240} = 1$

Stav Yosef, 316298876

Question 3

W.l.o.g let's say that the algorithm terminates when $C_1 > C_2$. Assume towards contradiction that $\frac{C_1}{2} > C_2$.

Mark $j' = \underset{j \in M_1}{\operatorname{argmin}}(p_j)$ and $C'_1 = C_1 - p_{j'}$, $C'_2 = C_2 + p_{j'}$.

If $job_{j'}$ is moved from M_1 to M_2. The condition of termination is that for every job_j that scheduled in M_1, moving j form M_1 to M_2 will not lower $|C_1\text{-}C_2|$.

If $|C'_1\text{-}C'_2| \geq |C_1\text{-}C_2|$ then $C'_2 > C'_1$

$C'_2 - C'_1 = C_2 + p_{j'} - (C_1\text{-}p_{j'})$

$C_2 - C_1 + 2p_{j'} \geq C_1 - C_2$ (because, $C_1 > C_2$)

$2p_{j'} \geq 2C_1 - 2C_2 > 2\left(C_1\text{-}\frac{1}{2}C_1\right) = C_1$ (because what we assumed at the start)

$p_{j'} > \frac{C_1}{2}$

$job_{j'}$ must be the only job on M_1. If not, job_j, $p_j + p_{j'} \geq 2p_{j'} > C_1$.

Hence $job_{j'}$ is the only job on M_1 and $p_{j'} = C_1$. But

$$C_1 = p_{j'} \leq \frac{1}{2}\sum_j p_j = \frac{1}{2}(C_1 + C_2) < \frac{1}{2}\left(C_1 + \frac{1}{2}C_1\right) = \frac{3}{4}C_1$$

Contradiction! $\frac{C_2}{2} \leq C_1 \leq 2C_2$. Let P be the approximation ratio, $P = \sum_j p_j = C_1 + C_2$, $OPT \geq \frac{P}{2} = \frac{1}{2}(C_1 + C_2)$.

In the worst case the algorithm terminates with $C_2 = \frac{C_1}{2} \rightarrow P = \frac{3C_1}{2}$, then $\frac{ALG}{OPT} \leq \frac{\frac{2P}{3}}{\frac{P}{2}} = \frac{4}{3}$

W.l.o.g let $C_1 > C_2$, and let job_j be the longest job that can be migrated that improves the gap.

Mark the gap as $D = C_1 - C_2$, moving job_j improves the gap so the $D' \leq D$.

1. If $C'_2 > C'_1$ then $D' = C'_2 - C'_1 = C_2 - C_1 + 2p_j = -D + 2p_j \leq D$, then $D' = -D + 2p_j \leq -p_j + 2p_j = p_j$. The new gap $< p_j$, and in every iteration the gap decreases until it stops. job_j will not be migrated again, therefore it is migrated at most once.
2. If $C'_1 > C'_2$ then the gap decreased, and job_j won't move again until $C_2 > C_1$ but then migrating job_j will only increase the gap so it won't move again.
3. If $C'_1 = C'_2$ then the algorithm terminates.

Therefore, every job is moved at most once $\rightarrow$ algorithm terminates after at most n moves.

Stav Yosef, 316298876

Question 4

As we discussed in the class the worst case of the algorithm is when the cow walks towards the hole and just before the hole the cow turns around.

Let $\mathrm{OPT} = 2^{\mathrm{j}} + \epsilon$ where $\mathrm{j} = \#\mathrm{iterations}$

Each distance the cow walked 4 times except the last d which the cow walks 2 times and the OPT value.

$$4\left(2^0 + 2^1 + 2^2 + \cdots + 2^{\mathrm{j}}\right) + 2 \cdot 2^{\mathrm{j}+1} + 2^{\mathrm{j}} + \epsilon = 4\left(2^{\mathrm{j}+1}\text{-}1\right) + 4 \cdot 2^{\mathrm{j}} + 2^{\mathrm{j}} + \epsilon$$
$$= 8 \cdot 2^{\mathrm{j}}\text{-}4 + 4 \cdot 2^{\mathrm{j}} + 2^{\mathrm{j}} + \epsilon \leq 8 \cdot 2^{\mathrm{j}} + 4 \cdot 2^{\mathrm{j}} + 2^{\mathrm{j}} + \epsilon = 13 \cdot 2^{\mathrm{j}} + \epsilon$$
$$\leq 13 \cdot \mathrm{OPT}$$

The competitive ratio is 13

Stav Yosef, 316298876

Question 5

Let M = maximum match and N = maximal match. Let e be an edge in $M\backslash N$, then in $N\backslash M$ there is at most 2 edges touching the vertices of e → $|M\backslash N| \le 2|N\backslash M|$
We know that for any set the following statement is valid: $|M| = |M \cap N| + |M\backslash N|$

$$|M| = |M \cap N| + |M\backslash N| \le |M \cap N| + 2|N\backslash M| \le 2|M \cap N| + 2|N\backslash M| = 2|N|$$

$|M| \le 2|N|$ → maximal matching is a 2-approximation for the maximum matching.

Online Algorithm

1. Let M be empty set (the matching group of edges)
2. Foreach $u \in U$
   a. Foreach $e = (u, v \mid v \in V)$ (foreach edge that touching u and any $v \in V$)
      If v still exists, Then $M = M \cup \{e\}, V = V\backslash\{v\}$. Break to step 2.

Correctness

The algorithm finds matching because there are no 2 edges that sharing the same vertex (we are achieving it by "$V = V\backslash\{v\}$")

2-Competitive

Let M be matching, the output of the algorithm. Assume towards contradiction that M is not a maximal matching then there is M' matching that $M \subset M'$ then there is an edge, $e = (u, v)$ s.t $e \in M'\backslash M$, this is impossible situation because when the vertex u was revealed the algo chose not to add it because either v is already deleted or another edge that touching u is picked, hence M' is not matching and M is maximal. Straight from 5.1 we can say that $2 \cdot M \le matching_{\text{maximum}}$

Let $V = \{v_1, v_2\}$ and $U = \{u_1, u_2\}$. Given a deterministic algorithm $Algo$ we need to prove that its competitive ratio is at least 2.

1. u_1 is revealed, edges are $(v_1, u_1), (v_2, u_1)$
2. Algo have 2 options
   a. If $Algo$ selects (v_1, u_1) reveal u_2 with edge (v_1, u_2) → $Algo$ can't add u_2 into the match but $OPT = 2$, $\{(v_2, u_1), (v_1, u_2)\}$
   b. If $Algo$ selects (v_2, u_1) reveal u_2 with edge (v_2, u_2) → $Algo$ can't add u_2 into the match but $OPT = 2$, $\{(v_2, u_2), (v_1, u_1)\}$

Therefore, we can see that $\frac{\text{OPT}}{\text{Algo}} = 2$ → $Algo$ is 2-competitive