Coursework2

s1926784

March 2020

1

(1) Figure 1 from lecture slides shows an example of a finite extensive form game of perfect information because every information set contains only 1 nodes. Let $Pl_1 = \{\emptyset\}$, $Pl_2 = \{B\}$($\emptyset$ denotes the root node). There is pure NE in this extensive form game: $s = (s_1 : s_1(\emptyset) = A, s_2 : s_2(B) = b)$ because none of the two players can unilaterally switch to other pure strategies to increase their own expected payoff. If player 1 switch to action B, his payoff will decrease to -10. If player 2 switch to action a, his payoff will remain the same. However this pure NE is not a SPNE. There is a sub-game tree starting at node 2. In this sub-game tree, the SPNE is $s_2(B) = a$. Therefore, The profile $s = (s_1, s_2)$ can not define a NE in this subgame.

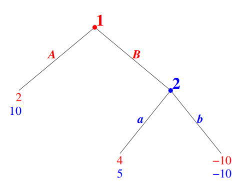

Figure 1: A finite extensive form game of perfect information

(2)First, in the finite games of perfect information, the backward induction says: When it is player i's turn to move, the player has a finite number of possible actions available, so the optimal actions necessarily exist at that stage(There can be more than one optimal actions). The optimal actions for player 1 can be described as: $a^{'} : h_i^{wa'}(s^{wa'}) = \max_{a \in Act(w)} h_i^{wa}(s^{wa})$ and $s_i^w = (\bigcup_{a \in Act(w)} s_i^{wa}) \bigcup \{w \mapsto a^{'}\}$. In other words, the player i will choose the actions that gives largest expected payoff at that stage. Second, note that if no players has the same payoffs at any two terminal notes, There is only a optimal action the player i which give the player i the largest payoff since each actions leads to a "leaf nodes" with distinct payoff. More formally, the element of $\max_{a \in Act(w)} h_i^{wa}(s^{wa})$ is 1: $|\max_{a \in Act(w)} h_i^{wa}(s^{wa})| = 1$. Then using bottom up algorithm the strategy identified in this way is necessary a SPNE of the game.

(3) Figure 2 shows the example. There is only a subgame showed also showed in Figure 2 since every node in the subtree is contained in the information set that is itself entirely contained in that subtree. In this subtree there is no SPNE because the two players, player 1 and player 2, can unilaterally switch to other pure strategies to increase their own expected payoff to increase their own payoff. If $s = (s_1 : s_1(in) = b, s_2 : s_2(inb) = d)$ is SPNE, the player two can switch to action c to increase own payoff. If $s = (s_1 : s_1(in) = b, s_2 : s_2(inb) = c)$ is SPNE, player 1 can switch to action a to increase own payoff. If $s = (s_1 : s_1(in) = a, s_2 : s_2(inb) = d)$ is SPNE, player 1 can switch to action b to increase own payoff. If $s = (s_1 : s_1(in) = a, s_2 : s_2(inb) = c)$ is SPNE, player 2 can switch to action d to increase own payoff. However, there is a pure NE in this Game. If $s_3(3) = out$, No matter what pure strategy other players play, The profile $s = (s_1, s_2, s_3(\emptyset) = out)$ is a NE.

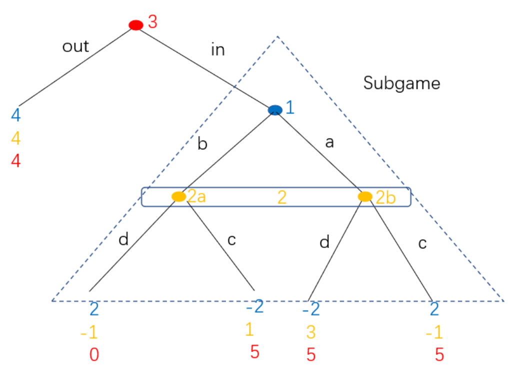

Figure 2: A finite extensive form game

2

(1)This game does not satisfy ”perfect recall”. If the game is perfect recall, if each player recalls exactly what he did in the past. More formally, on the path from the initial node to a decision node x of player i, list in chronological order which information sets of i were encountered and what player i did there. Call this list the experience $X_i(x)$ of player i in node x. The game has perfect recall if nodes in the same information set have the same experience. In the given Game, different experiences in the two nodes of information set $\{(BaC), (BaD)\}$. In the left node: $X_1((B, a, C)) = \{\ \{\emptyset\}, B, \{Ba\}, C, \{(BaC), (BaD)\}\}$. In the right node: $X_1((B, a, D)) = \{\ \{\emptyset\}, B, \{Ba\}, D, \{(BaC), (BaD)\}\}$. $X_1((B, a, C)) \neq X_1((B, a, D))$. In other words, player 1 forget what he did.

(2)$Pl_1 = \{\{\emptyset\}, Ba\}$($\emptyset$ denotes the root node), $Pl_2 = \{B, BaC, BaD\}$. First, compute the SPNE and expected payoff for both player 1 and for the subgame by transform the finite extensive form subgame 3 into strategic form:

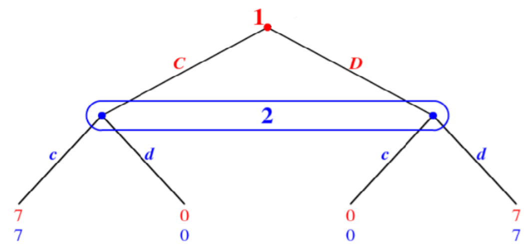

Figure 3: A subgame tree of the game

$$\begin{bmatrix} (7,7) & (0,0) \\ (0,0) & (7,7) \end{bmatrix}$$

There are two pure NEs in this strategic form game: $x^* = (x_1^* = (1,0), x_2^* = (1,0))$ and $x^* = (x_1^* = (0,1), x_2^* = (0,1))$. There is also a mixed NE in this game. By using the useful Corollary, the mixed NE is $x^* = (x_1^* = (\frac{1}{2}, \frac{1}{2}), x_2^* = (\frac{1}{2}, \frac{1}{2}))$

(i) When SPNE of the subgame is $x^* = (x_1^* = (1,0), x_2^* = (1,0))$: The expected payoff of the two player under the subgame is: $U_1(x^*) = 7 = U_2(x^*)$. By using the backward induction algorithm(Figure 4 shows the process of induction), $b_1 : b_1(\emptyset) = B, b_1(Ba) = C$, $b_2 : b_2(B) = a, b_2(BaC) = b_2(BaD) = c$. Therefore, the behavior strategies is $b = (b_1, b_2)$

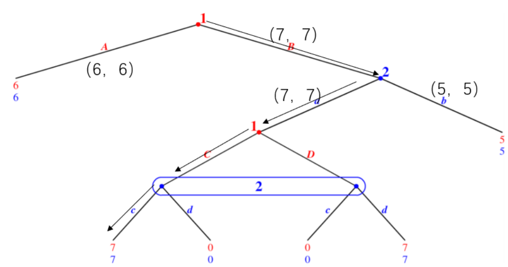

Figure 4: The process of backward induction for(i)

(ii) When SPNE of the subgame is $x^* = (x_1^* = (0,1), x_2^* = (0,1))$ The expected payoff of the two player under the subgame is $U_1(x^*) = 7 = U_2(x^*)$. By using the backward induction algorithm(Figure 5 shows the process of induction), $b_1 : b_1(\emptyset) = B, b_1(Ba) = D$, $b_2 : b_2(B) = a, b_2(BaC) = b_2(BaD) = d$. Therefore, the behavior strategies is $b = (b_1, b_2)$

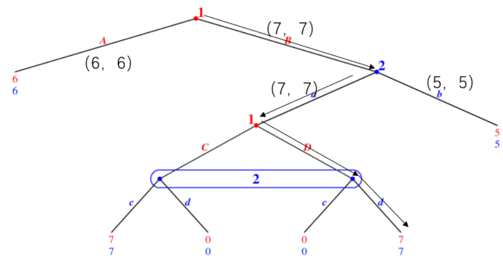

Figure 5: The process of backward induction for(ii)

(iii) When SPNE of the subgame is $x^* = (x_1^* = (\frac{1}{2}, \frac{1}{2}), x_2^* = (\frac{1}{2}, \frac{1}{2}))$: The expected payoff of the two player under the subgame is $U_1(x^*) = \frac{7}{2} = U_2(x^*)$.By using the backward induction algorithm(Figure 6 shows the process of induction), $b_1 : b_1(\emptyset) = A$, $b_2 : b_2(B) = b$.

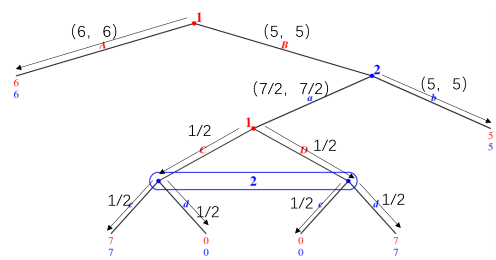

Figure 6: The process of backward induction for(iii)

(3)There is a pure NE other than SPNE still in this game. This pure NE is eliminated by backward induction. The pure NE is: $s = (s_1 : s_1(\emptyset) = A, s_2 : s_2(B) = b))$ For the rest of nodes of player i where it is player's i's turn to move, they can choose whatever actions they like. Since no player can unilaterally switch to other pure strategies to increase their own expected payoff, The validity of the pure NE is checked.

(4) Those SPNE found in (2) do not involve non-credible threats since we just find those NE by using backward induction. The SPNE just defines a NE for every subame of the game. However, In (3), the NE is Non-credible threats because player 2 "threats" the player 1 not to choose action B at the beginning of the game. If player 1 choose action B, player 2 will choose action b such that player 1 can not choose action B to increase own payoff. In fact, If player 1 choose action B, player 2 will not insist on choosing the action b because player 2 can choose other actions to increase own payoff.

3

(a) As we are working with a congestion game, we can find a pure Nash Equilibrium by starting at any pure strategy profile, and iteratively improving it until we can't. Let's say all the players take the route $s \to v_1 \to t$. For this starting point, the costs to all three players are same: 15. Then we can do iterative improvements as follow:
(i) player 1 switches to $s \to v_2 \to t$ to decrease cost to 5 and the costs to player 2 and player 3 are 10
(ii) player 2 switches to $s \to v_3 \to t$ to decrease cost to 3 and the costs to player 3 is 6 and the cost to player 1 does not change.
After the two iteration, none of those three players can switch to other pure strategies to decrease their own cost, a pure NE is found: player 1 choose $s \to v_2 \to t$, player 2 choose $s \to v_3 \to t$ and player 3 choose $s \to v_1 \to t$.
(b)However, the pure NE computed in (a) is not unique. In this game, all pure NE send one of the three players via the route $s \to v_2 \to t$ and one via the route $s \to v_3 \to t$ and one via the route $s \to v_1 \to t$. After all of the three players can choose one of the strategy, the profile is still a pure NE.
(c)The pure NE we found in (a) is the profile that maximize the social welfare. Therefore, $max_{x \in X} welfare(x) = \frac{1}{14}$. the pure NE provide the minimum social welfare is also the one we found in(a). Therefore, the the pure price of anarchy in this atomic network congestion game is $\frac{max_{s \in S} welfare(s)}{min_{s \in pure-NE} welfare(s)} = 1$

4

(a)Assume the bidders bid their true valuations, Then given these valuations, the VCG mechanism firstly picks an outcome that maximizes the sum total valuation of all the bidders, i.e., maximizes the total social welfare of the outcome. There is a outcome that maximize social welfare:Based on observation, if bidder L gets painting T_2 and the bidder S gets painting T_1, T_3, then the sum total valuation of this outcome is:$v_S(T_1, T_3) + v_L(T_2) = 42 + 21 = 63$. assume the outcomes give bidder L painting T_2 and bidder S paintings T_1, T_3. Now calculate the payments of S, L, B for that outcome: The VCG payment for S in the outcome is $max_{o \in O}(v_L(o) + v_B(o)) - (v_L(T_2) + v_B(\emptyset)) = (v_L(T_2, T_3) + v_B(T_1) - (v_L(T_2) + v_B(\emptyset)) = (43 + 20) - (21 + 0) = 42$. The payment for bidder L is: $max_{o \in O}(v_S(o) + v_B(o)) - (v_S(T_1, T_3) + v_B(\emptyset)) = v_B(T_1, T_2, T_3) - (v_S(T_1, T_3) + v_B(\emptyset)) = 62 - 42 = 20$. The payment for bidder B is: $max_{o \in O}(v_S(o) + v_L(o)) - (v_S(T_1, T_3) + v_L(T_2)) = (v_S(T_1, T_3) + v_L(T_2)) - (v_S(T_1, T_3) + v_L(T_2)) = 0$.
(b) However, the VCG outcome I have calculated in Part(a) is not unique. if bidder L gets painting T_2, T_3 and bidder B gets painting T_1, then the sum total valuation of this outcome is:$v_L(T_2, T_3) + v_B(T_1) = 43 + 20 = 63$. Assume the outcome that gives bidder L paintings T_2, T_3 and bidder B painting T_1, the payment for bidder S is: 0. The payment for the bidder L is: 42. The payment for bidder B is:20.

(c) The VCG Mechanism force the bidders to reveal their true value function by asking them to pay money and bidders declare their true value function is a (weakly) dominant strategy. If there are 20 Andy Warhol paintings, the VCG mechanism is not proper mechanism because compute c^* that maximize the total value is NP-hard problem.