# Game theory Answer

Enbo lyu

May 2024

## 1 Part 1

1. For certainty, we know exactly what the consequence of our choice will be; for uncertainty, we don't know the consequences of our choices will be: for every possible choice, there are multiple possible consequences, each with an attached probability.

2. A preference relation is a binary relation $\succeq \subseteq \Omega \times \Omega$, which is required to satisfy:

   (1) Reflexivity:

   $\omega \succeq \omega$ for all $\omega \in \Omega$

   (2) Completeness:

   for all $\omega, \omega' \in \Omega$ we have either $\omega \succeq \omega'$ or $\omega' \succeq \omega$

   (3) Transitivity:

   for all $\omega, \omega', \omega'' \in \Omega$, if $\omega \succeq \omega'$ and $\omega' \succeq \omega''$ then $\omega \succeq \omega''$

   Reflexivity means every option is at least as good as itself. Completeness means with any two options A and B, you either prefer A or indifferent between A and B, or prefer B or indifferent between them. Transitivity make sure logical relationship is consistent, if you prefer A over B, prefer B over C, the you must also prefer A over C.

3. If both $\omega \succeq \omega'$ and $\omega' \succeq \omega$ then we say you are indifferent between $\omega$ and $\omega'$, and we write $\omega \sim \omega'$.

4. Preferences are not observable, choices are revealed preferences. $\omega \succeq \omega'$ means that if you have a choice between them, you will choose $\omega$. If you have a choice between two option, one of which will result in $\omega$ and the other result in $\omega'$, you will choose the one resulting in $\omega$.

5. A utility function $u : \Omega \to \mathbb{R}$ is said to represent a preference relation $\succeq$ iff we have:

   $$u(\omega) \geq u(\omega') \quad \text{iff} \quad \omega \succeq \omega'$$

6. For every preference relation $\succeq \subseteq \Omega \times \Omega$ there is a utility function $u : \Omega \to \mathbb{R}$ that represents $\succeq$.

7. No. You make the choice because $\omega \succ \omega'$, the utility values are just chosen to reflect this. But if you choose picked the number right, then you behave like you are maximising the utility.

8. $\Sigma$ represents a set of strategies that are aviliable to our decision maker, such as choices, actions, etc.

9. An outcome function is $g : \Sigma \to \Omega$.

10. Decision making under certainty is given by a quad

    $$\langle \Omega, u : \Omega \to R, \Sigma, g : \Sigma \to \Omega \rangle$$

11. The decision maker selects the strategy $\sigma*$ that

$$\sigma^* \in argmax_{\sigma \in \Sigma} u(g(\sigma))$$

12. A lottery over S is a probability distribution over S.

13. $EU(l) = \sum_{\omega \in \Omega} u(\omega) P(\omega, l)$

14. A preference relation $\succeq \subseteq$ Lott$(\{\mathcal{W}, \mathcal{L}\}) \times$ Lott$(\{\mathcal{W}, \mathcal{L}\})$ over win-lose lotteries needs to satisfy completeness, reflexivity, transitivity and continuity, iff there exists a utility function $u : \mathcal{W}, \mathcal{L} \to R$ such that $l_1 \succeq l_2$ iff $EU(l_1) \geq EU(l_2)$ where $EU(l) = \sum_{\omega \in W,L} u(\omega) P(\omega, l)$.

15. (a) The Equivalence Axiom: The structure of a lottery is irrelevant.
    (b) The Monotonicity Axiom: If you prefer $l_1$ over $l_2$, then you will prefer to maximise the probability of getting $l_1$ over $l_2$.
    Suppose $l_1 \succ l_2$, then $p \geq q$ iff $pl_1 + (1-p)l_2 \succeq ql_1 + (1-q)l_2$
    (c) The Archimedean Axiom: We can quantify our preferences over lotteries.
    If $l_1 \succeq l_2 \succeq l_3$, then there is some $p \in [0, 1]$ such that $l_2 \sim pl_1 + (1-p)l_3$
    (d) The Independence Axiom(subsitution): We can freely substitute lotteries that we are indifferent between.

16. A preference relation $\succeq \subseteq lottery(\Omega) \times lottery(\Omega)$ statisfy the Von Neumann axioms iff there exist $u : \Omega \to R$ such that $l_1 \succeq l_2$ iff $EU(l_1) \geq EU(l_2)$ where $EU(l) = \sum_{\omega \in \Omega} u(\omega) P(\omega, l)$

17. Proof of Von Neumann and Morgensterm's Theorem

18. $\langle \Omega, u : \Omega \to R, g : \Sigma \to Lott(\Omega) \rangle$

19. The decision maker selects $\sigma^*$ that maximise the expected utility:

$$\sigma^* = argmax_{\sigma \in \Sigma} EU(u(g(\sigma))) = argmax_{\omega \in \Omega} P(\omega, g(\sigma)) u(\omega))$$

20. If the value of $l_1 \geq l_2, l_4 \geq l_3$, but you have preferences $l_1 \succ l_2$ and $l_4 \succ l_3$, then you don't satisfy the Von Neumann and Morgonstern's Theorem.

21. 1) single-peaked preferences; 2) dichotomous preferences; 3) lexigraphical preferences.
    (a) Single-peaked preferences: this is a preference relation that exist a most preferred strategy $\omega^*$, and the preference that is closer to $\omega^*$ would be preferred over those that are far away.
    (b) dichotomous preferences: This is a preference relation that only has two outcomes, Win or Loss. Formally, there exists outcome $\mathcal{W} \subseteq \Omega$ and $\mathcal{L} \subseteq \Omega$ such that:
    - $\mathcal{W} \cup \mathcal{L} = \Omega$
    - $\mathcal{W} \cap \mathcal{L} = \emptyset$
    - $\forall \omega_1, \omega_2 \in \mathcal{W}, \omega_1 \sim \omega_2$
    - $\forall \omega_1, \omega_2 \in \mathcal{L}, \omega_1 \sim \omega_2$
    - $\forall \omega_1 \in \mathcal{W}, \omega_2 \in \mathcal{L}, \omega_1 \succ \omega_2$
    (c) Preferences are lexicographic if outcomes can be characterised by an ordered set of attributes.

22. $\Phi = \{x_1, ..., x_l\}$, where each variable takes value $\top or \perp$ that represent True or False.

23. (a) The propositional formula representation for dichotomous boolean preferences is complete, ie. any dichotomous boolean preferences can be represented by a propositional formula.
    (b) The propositional formula representation can be exponentially more compact than the naive representation.

(c) There exist dichotomous preference relations that the smallest propositional representation is of size exponentially in $|N|$ and hence no better than the naive representation.

24. A **weighted formula** or **rule** is a pair $(\varphi, x)$ where $\varphi$ is a propositional formula (logical formula) and $x \in \mathbb{R}$. Sometimes this is written as $\varphi \to x$. The **rule base** $\mathcal{R} = \{(\varphi_1, x_1), \dots, (\varphi_k, x_k)\}$ defines the utility function. The utility function $u_{\mathcal{R}}(v)$ is defined as:

$$u_{\mathcal{R}}(v) = \sum_{\substack{(\varphi_i, x_i) \in \mathcal{R} \\ v \models \varphi_i}} x_i$$

25. (a) It is a complete representation for utility functions $u : 2^N \to R$.
    (b) It can be exponentially more compact than naive representation.
    (c) There exist utility functions for which even the weighted formula representation required exponentially many rules.

26. (a) The problem of determining whether there exist a valuation $v$ such that $u_R(v) \geq k$ with a target value $k$ and a rule $\mathcal{R}$ is NP-complete.
    (b) The problem of finding the optimal solution $v^*$ satisfying $v^* \in argmax_v \sum u_R(v)$ is $\mathrm{FP}^{\mathrm{NP}}$-complete.

27. $u_i : \Sigma_1 \times ... \times \Sigma_n \to R$, note that $u_i(\sigma_1, \dots, \sigma_n)$ is shorthand for $u_i\left(g\left(\sigma_1, \dots, \sigma_n\right)\right)$

28. 
    - The game has a set $N = 1, ...n$ players.
    - Each player $i$ simultaneously chooses a strategy from their set of pure strategies $\Sigma_i$
    - $u_i : \Sigma_1 \times \cdots \times \Sigma_n \to \mathbb{R}$ is the utility function for agent $i \in N$. The utility $i$ gets depends not on only her actions, but on the actions of others, and similarly for other agents.

29. A strategy profile $\vec{\sigma}$ is a tuple of strategies, one for each player.

$$\vec{\sigma} = (\sigma_1, \dots, \sigma_i, \dots, \sigma_n) \in \Sigma_1 \times \cdots \times \Sigma_i \times \cdots \times \Sigma_n$$

30. $(\vec{\sigma_{-i}}, \sigma_i') = (\sigma_1, \dots, \sigma_i', \dots, \sigma_n)$

31. A strategy $\sigma$ with the property that no matter what other players' strategies are, the best response would be to choose $\sigma$. $\sigma_i \in \Sigma_i$ is a dominant strategy if: $\forall \vec{\sigma}$ and $\forall \sigma_i' \in \Sigma_i$, we have $u_i(\vec{\sigma_{-i}}, \sigma_i) \geq u_i(\vec{\sigma_{-i}}, \sigma_i')$

32. Dominant strategy equilibrium is a strategy profile that everyone has chosen a dominant strategy.

33. A strategy profile is a Nash equilibrium if no player would change its strategy assuming others not changing. $\vec{\sigma_i}$ is a Nash equilibrium if there is no player $i \in N$ and strategy $\sigma_i' \in \Sigma_i$ such that $u_i(\vec{\sigma_{-i}}, \sigma_i') > u_i(\vec{\sigma_i})$. Nobody can benefit by deviating from NE.

34. A player's best response to a strategy profile $\vec{\sigma}$ is the strategy that gives the highest utility.

$$BR_i(\vec{\sigma}) = argmax_{\sigma_i \in \Sigma_i} u(\vec{\sigma_{-i}}, \sigma_i)$$

35. $BR(\vec{\sigma}) = BR_1(\vec{\sigma}) \times ... \times BR_n(\vec{\sigma})$

36. $s \in S$ is a fixed point of a function $f : S \to S$ if $s = f(s)$. $s \in S$ is a fixed point of a function $f : S \to \mathbf{2}^S$ if $s \in f(s)$.

37. $\vec{\sigma} \in NE(G)$ iff $\vec{\sigma} \in BR(\vec{\sigma})$. The NE of a game

$$NE(G) = \{\vec{\sigma} \in \Sigma_1 \times ... \times \Sigma_n | \vec{\sigma} \in BR(\vec{\sigma})\}$$

38. $\sigma_i \in \Sigma_i$ is a dominant strategy for player i iff $\sigma_i \in \bigcap_{\vec{\sigma} \in \Sigma} BR_i(\vec{\sigma})$

39. A strategy profile $\vec{\sigma}$ is Pareto efficient if making one player better requires making another player worse.

40. Max utilitarian: $\vec{\sigma^*} = argmax_{\vec{\sigma}} \sum_i u_i(\vec{\sigma})$

41. Max egalitarian: $\vec{\sigma^*} = argmax_{\vec{\sigma}}(min_i u_i(\vec{\sigma}))$

42. Every dominant strategy equilibrium is NE, reverse need not be the case.

43. Dominant strategy equilibrium and NE need not be Pareto efficient, maximised utilitarian or egalitarian social welfare.

44. The outcome that max utilitarian social welfare is Pareto efficient, the converse need not be the case.

45. Prisoner's dilemma: DD is dominant strategy equilibrium, NE, all except DD is Pareto optimal, CC maximise utilitarian social welfare.

46. Game of Chicken: No dominant strategy equilibrium, CD and DC are NEs, all but DD is Pareto optimal, all but DD maximise utilitarian social welfare.

47. Focal point: Sometimes outcomes in games have features that make them stand out, independently of the utility structure in games. Evolutionary approaches: If we have time, we learn to coordinate.

48. Stag Hunt

49. Hawk Dove

50. Matching Pennies

51. A game is dominance solvable if after Iterated Elimination of Strictly Dominated Strategies(IEDS), there is only a single outcome remains.

52. If a game G is dominance-solvable, then the unique outcome of the game according to IEDS is the unique pure-strategy Nash equilibrium of G.

53. Potential games. And congestion games are an important class of potential games.

54. If there exist a function $P : \Sigma_1 \times ... \times \Sigma_n \rightarrow R$... We have

$$u_i(\vec{\sigma_{-i}}, \sigma_i) - u_i(\vec{\sigma_{-i}}, \sigma_i') = P_i(\vec{\sigma_{-i}}, \sigma_i) - P_i(\vec{\sigma_{-i}}, \sigma_i')$$

55. Every finite potential game has a pure Nash Equilibrium.

56. **Theorem: Complexity of Boolean Games**

    (a) It is co-NP-complete to check whether an outcome forms a NE in a Boolean game.

    To prove this problem is co-NP-complete, we show the complement problem: some player has a beneficial deviation - is NP-complete.

    - Membership of NP: Guess a player $i$ and a strategy $\vec{\sigma_i'}$ for $i$, verify that $i$ does better with $\sigma_i'$ than their component of $\vec{\sigma}$. This verification can be done in polynomial time ( evaluate the utility of player i with the current strategy profile $\vec{\sigma}$; evaluate the utility of player i with the modified strategy profile $(\vec{\sigma_{-i}}, \sigma_i')$; then compare the two utilities to see which one is greater)
    - NP Hardness: Reduce from a known co-NP-Complete problem: SAT. Given a SAT instance $\phi$, define 1-player game with $\gamma_1 = \phi \wedge z$ where $z$ is a new variable. Define a strategy $\sigma_1$ which sets all variables to False, and check whether $\phi$ is satisfiable.
      **If $\phi$ is satisfiable**, then there exist strategy profile that can make $\phi$ True by changing all False to True. This means there is a beneficial deviation from $\sigma_1$, therefore $\sigma_1$ is not a NE.
      **If $\phi$ is not satisfiable**, this means there is no assignment of variables that can make $\phi$ True, no matter how the player changes their strategies, which implies no deviation, therefore $\sigma_1$ is a NE.

(b) It is $\Sigma_2^P$-complete to check whether a Boolean game has a NE.

- Membership in $\Sigma_2^P$: A Boolean game has a NE can be expressed in Quantified Boolean Formula: $\exists\vec{\sigma}\ \forall$ player $i$, there is no beneficial deviation. This is a QBF with two quantifiers $\exists, \forall$, which is an instance of $QBF_{2,\exists}$, whose truth can be checked in $\Sigma_2^P$.
- $\Sigma_2^P$ Hardness: Reduce from a known $\Sigma_2^P$-complete game: $QBF_{2,\exists}$ in form $\exists X \forall Y \psi(X,Y)$ to a 2-player Boolean game.
  Define a Boolean game,
  - $\Phi_1 = X \cup \{x\}$
  - $\Phi_2 = Y \cup \{y\}$
  - $\gamma_1 = \psi(X,Y) \vee (x \leftrightarrow y)$
  - $\gamma_2 = \neg\psi(X,Y) \wedge \neg(x \leftrightarrow y)$

  **Assume** $\exists X \forall Y \psi(X,Y)$ **is True**, then player 1 can choose value for $X$ such that $\psi(X,Y)$ is True for any $Y$, then $\gamma_1$ will always be True, and player 2 cannot make it False by deviating. Therefore NE.
  **Assume** $\exists X \forall Y \psi(X,Y)$ **is False**, then $\exists X \forall Y \neg\psi(X,Y)$ is True. For any choice of $X$, there is a $Y$ that can make $\psi(X,Y)$ False. Therefore if $\gamma_1$ is satisfied, player 2 has a beneficial deviation to make $\gamma_1$ False and make $\gamma_2$ True. And if $\gamma_2$ is True, player 1 then would have initiative to deviate to make $\gamma_1$ True. Therefore no NE.

Threfore, the game is in $\Sigma_2^P$ because it can be expressed with QBF, whose truth can be checked in $\Sigma_2^P$, it is $\Sigma_2^P$-Hard becuase it can be reduced from a known $\Sigma_2^P$-complete problem.

57. **Nash Theorem:** Every finite game has a Nash Equilibrium in mixed strategies.

58. The **support** of a mixed strategy $\mu_i : \Sigma_i \to [0,1]$ is the set of pure strategies played with positive probability in $\mu_i$: $supp(\mu_i) = \{\sigma | \mu_i(\sigma) > 0\}$. A mixed strategy $\mu_i$ is fully mixed if $supp(\mu_i) = \Sigma_i$, ie. all pure strategies are played with positive probability.

59. (a) $EU_1(T,q) = q \times v_1^1 + (1-q) \times v_2^1$

60. Then $EU_1(T,q) = EU_1(B,q), EU_2(L,p) = EU_2(R,p)$

61. We have $EU_i(\sigma_i, \vec{\mu}_{-i}) = EU_i(\sigma_j, \vec{\mu}_{-i})$

62. The Support Enumeration Method (SEM)

63. $BR_i : MS_1 \times \cdots \times MS_n \to 2^{MS_i}$ where $BR_i(\vec{\mu}) = \arg\max_{\mu_i \in MS_i} EU_i(\vec{\mu}_{-i}, \mu_i)$, and the best response function of the game is: $BR(\vec{\mu}) = BR_1(\vec{\mu}) \times \cdots \times BR_n(\vec{\mu})$

64. (a) Zemelo's algorithm terminates, the root labelled with a payoff profile that would be obtained by a NE strategy profile.
    (b) The algorithm runs in a polynomial time in the size of the game tree.

65. (a) Every extensive form game with perfect information and no chance moves has a NE in pure strategies.
    (b) Pure strategies NE in extensive form game can be computed in polynomial time with Zermelo's algorithm.
    (c) If no two leaf nodes have the same utility for any player, then the NE is unique.

66. A strategy profile $\vec{\sigma} = (\sigma_1, ..., \sigma_n)$ is a subgame perfection NE if it is a NE in each subgame $G'$ of G.

67. (a) Every extensive form game with perfect information and no chance moves has a SPNE.
    (b) SPNE for extensive form games can be computed in polynomial time using Zemolo's algorithm.

68. A game is a **Imperfect Information Game** if $[v] = [v']$, then the decision maker does not know whether she is in $v$ or $v'$, and $Action(v) = Action(v')$

69. A strategy in Imperfect Information game is a function that assigns an action to each information set: $\sigma_i : \mathcal{I}_i \to A_i$

70. 1) Mixed strategies; 2) Behavioral strategies

71. Behaviour Strategies involve randominzing at each decision node independently, they can choose different probabilities for each action at each point when he has to make a decision.

    (a) A behaviour strategy $\beta_i : V_i \to \Delta A_i$, such that $supp(\beta_i(v)) \subseteq A(v)$. ($\Delta A_i$ represents the set of all possible probability distributions over the actions available to player i. $supp()$ if the support of a probability distribution is the set of actions that have a non-zero probability of being chosen, here $supp(\beta_i(v))$ includes all actions that player i might choose at node v.

    Mixed strategy will at the start of the game, give the same decision eg. exist or not, regardless of which decision point it is currently in because it cannot distinguish. But behavioural strategies randomised ( give a probability distribution over actions at each decision node independently).

72. In extensive form game with perfect recall,

    (a) For every mixed strategies there exists a behavioural strategy that yield the same probability distribution over outcomes.

    (b) For every behavioural strategies there exists a mixed strategy that yield the same probability distribution over outcomes.

73. $value = lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T} u_i(\omega_t)$. This does not always converge, but if players use **automata strategies**, this always converges.

74. 1) ALLD; 2) ALLC; 3) GRIM; 4) TIT-FOR-TAT; 5) TAT-FOR-TIT.

75. Finite machine strategies playing against each other will generate a run with finite non-repeating sequence and finite sequences that repeats infinitely often. Note that the non-repeating sequence may be empty, and the repeating sequence may be length one.

76. The value of infinite run will just be the average of the finite repeating sequence.

77. The player $i$'s **security value** in a game G is the best utility that it can guarantee for itself, no matter what the other players do. Practical: the min(max) of the utility of all strategies of each player.

78. In an infinitely repeated game, every outcome in which every player gets at least their security value can be sustained as a Nash Equilibrium.

79. Discounted Sum used a **discount factor** $0 < \sigma < 1$ to discount the value of future rounds. The value of infinite run is then: $\sum_{u\in N} \delta^u u_i(\omega_u)$, the core identity is $\sum_{n=1}^{\infty} \delta^n = \frac{1}{1-\delta}$.

80. iterated boolean

81. **Zero sum games** are games in which for every $\omega \in \Omega$ we have $\sum_{i\in N} u_i(\omega) = 0$. It is strictly competitive, best outcome of me is the worst outcome of you.

82. $\bar{v} = max_{\sigma_1\in\Sigma_1} min_{\sigma_2\in\Sigma_2} u_1(\sigma_1, \sigma_2), \underline{v} = min_{\sigma_2\in\Sigma_2} max_{\sigma_1\in\Sigma_1} u_2(\sigma_1, \sigma_2)$

83. Because we know what every row we choose, the column player will choose the smallest value.

84. Then $\bar{v} = \underline{v}$. ie. In zero-sum game, NE and minimax/maximin coincide. The value for player 1 is called the **value of the game**.

85. (a) For player 2, the goal is to minimise $U_1^*$ subject to

$$\sum_{\sigma_2^k\in\Sigma_2} u_1(\sigma_1, \sigma_2^k) p_2^k \le U_1^*, \forall \sigma_1 \in \Sigma_1$$

$$\sum_{\sigma_2^k\in\Sigma_2} p_2^k = 1, p_2^k \ge 0$$

(b) For player 1, the goal is to maximise $U_1^*$ subject to

$$\sum_{\sigma_1^j \in \Sigma_1} u_1(\sigma_1^j, \sigma_2) p_1^j \geq U_1^*, \forall \sigma_2 \in \Sigma_2$$

$$\sum_{\sigma_1^j \in \Sigma_1} p_1^j = 1, p_1^j \geq 0$$

86. In extensive form win-lose games, **Zermolo's algorithm** tells us which player can force a win; Zermolo's algorithm are the optimal strategies for all players, and would give the winning strategies for the relevent player that can force the win. As a corollary, finite extensive form win-lose games are determined: one of the players can force a win.

87. P-complete.

88. (a) P-complete
    (b) PSPACE-complete
    (c) EXPTIME-complete
    (d) EXPSPACE-complete
    (e) high complexity

## 2 Part 2

1. A congestion game is defined by n players, a set of resources $E$ (shown as edges in the network), cost or latency functions $c_e : [n] \to R$; and strategies of each player $i$: $S_i \subseteq 2^E$ (which can be think of as all paths from a node $s_i$to $t_i$.
2. $C_i(P) = \sum_{e \in P_i} c_e(n_e(P))$, where $n_e(P)$ is the number of strategies that include resource e.
3. This tuple defines a game in which each player wants to select a strategy to minimize their own cost.
4. The cost of using a resource depends only on the number of times it's being used.
5. A game $G\left(n, A = A_1 \times \cdots \times A_n, (C_i)_{i=1}^n\right)$, with $n$ players, strategies $A_i$ for player $i$, and cost function $C_i$ is an exact potential game if there is a function $\Phi : A \mapsto \mathbb{R}$ such that:
$$C_i(a_i, a_{-i}) - C_i(a_i', a_{-i}) = \Phi(a_i, a_{-i}) - \Phi(a_i', a_{-i})$$
6. A joint strategy is a local minimum of the exact potential iff it is a pure Nash Equilibrium.
7. True.
8. Every congestion game is a potential game.
9. Every congestion game has at least one NE.
10. $PoA = \frac{\text{Cost of worst NE}}{\text{optimal non-selfish cost}}$
11. Linear congestion games are those with cost functions $c_e(k)$ as a linear function: $c_e(k) = a_e k + b_e$, for some non-negative constants $a_e$ and $b_e$.
12. The Price of Anarchy of linear congestion game is 5/2.
    - Upper bound
    Consider a linear congestion game with a cost function $c_e(k) = a_e k + b_e$, with a Nash equilibrium $\vec{P} = (P_1, \ldots, P_n)$, and $\vec{P^*} = (P_1^*, \ldots, P_n^*)$ is the strategy profile with the optimal non-selfish cost. Then the total cost of $\vec{P}$ is:
    $$\begin{aligned}\sum_i C_i(\vec{P}) &= \sum_i \sum_{e \in P_i} c_e(n_e(\vec{P})) \\ &= \sum_e c_e(n_e(\vec{P})) \cdot n_e(\vec{P}) \\ &\leq \sum_e c_e(n_e(\vec{P^*})) \cdot n_e(\vec{P^*}) \\ &\leq \sum_e c_e(n_e(\vec{P}) + 1) \cdot n_e(\vec{P^*})\end{aligned}$$
    Given that: for all non-negative $x, y : 3(x+1)y \leq x^2 + 5y^2$,
    $$\begin{aligned}\sum_e c_e(n_e(\vec{P}) + 1) \cdot n_e(\vec{P^*}) &\leq \frac{1}{3} \sum_e c_e(n_e(\vec{P})) \cdot n_e(\vec{P}) + \frac{5}{3} \sum_e c_e(n_e(\vec{P^*})) \cdot n_e(\vec{P^*}) \\ &= \frac{1}{3} \sum_i C_i(\vec{P}) + \frac{5}{3} \sum_i C_i(\vec{P^*})\end{aligned}$$
    Therefore:
    $$\begin{aligned}\frac{2}{3} \sum_i C_i(\vec{P}) &\leq \sum_i C_i(\vec{P^*}) \\ \sum_i C_i(\vec{P}) &\leq \frac{5}{2} \sum_i C_i(\vec{P^*})\end{aligned}$$

- Lower bound

13. **Routing games** are essentially congestion games with infinity many players. They both have latency function $c_e(\cdot)$, in routing games every path $P_{i,j}$ as has a rate of flow $r_{i,j}$.

14. The Price of Anarchy of linear routing games is 4/3.

15. A **load balancing game** is defined by:
    - m identical machines
    - n tasks/ players with weights $w_1, ..., w_n$
    - The strategy of each player $i$ to select a machine $a_i$
    - a load function of a machine $j$: $l_j = \sum_{i:a_i=j} w_i$, and the cost of player $i$ is $l_{a_i}$.

16. The objective of load balancing games is to minimize the highest load on each machine, balance the load as evenly as possible across all the players.

17. The PoA of pure Nash equilibrium of balancing games of m identical machines when the social cost is the makespan is $2 - \frac{2}{m+1}$.

18. (a) Fixed a pure NE and a player $i$ in the machine that has the maximum load.
    (b) Since player $i$ prefers machine $a_i$ over $j$: $l_{a_i} \leq l_j + w_i$
    (c) Summing for all $j \neq a_i$, get $(m-1)l_{a_i} \leq \sum_{j\neq a_i} l_j + (m-1)w_i \to ml_{a_i} \leq \sum_j l_j + (m-1)w_i$
    (d) Since the optimal makespan OPT must be at least the load of the heaviest task, $OPT \geq max(w) \geq w_i$; also, OPT must also be at least the average load per machine because if the load is perfectly banlanced, each machine would have a load of $\frac{\sum_j l_j}{m}$, therefore $OPT \geq \frac{\sum_j l_j}{m}$. Therefore:
    $$OPT \geq max(w_i, \frac{\sum_j l_j}{m})$$
    (e) Substitute what we have from step (d) into the equation of step (c):
    $$ml_{a_i} \leq \sum_j l_j + (m-1)w_i \leq mOPT + (m-1)OPT = (2m-1)OPT$$
    (f) $COST = l_{a_i} \leq 2 - \frac{1}{m}$

19. **Price of Stability** is the optimistic version of the Price of Anarchy. PoA measures the inefficiency of equilibrium in a game, it captures how much worse the system performs when players act selfishly compared to if they acted in a coordinated way to achieve the optimal outcome, if higher the PoA, the more inefficient the equilibrium; PoA measures the best-case inefficiency of equilibrium, if $PoS = 1$, means the best equilibrium is as good as the optimal outcome, indicating perfect efficiency in the best equilibrium, if $PoS > 1$, means even the best equilibrium is worse than the optimal outcome. **PoA measures how bad things can get when players act selfishly, PoS measures how good things can get when player act selfishly.**

20. $PoS = \frac{\text{cost of best equilibrium}}{\text{optimal non-selfish cost}}$

21. $c_e(k) = \frac{a_e}{k}$

22. The unique NE in the fair allocation game is: every player i selects the direct edge $(s_i, t)$, with the social cost $1 + \frac{1}{2} + ... + \frac{1}{n} = H_n$

23. The optimal solution is for all player to go through node v, with total cost $1 + \epsilon$

24. For $\epsilon \approx 0$, $PoS \approx H_n/1 \approx lnn$

25. The PoS of linear congestion game is $1 + \frac{\sqrt{3}}{3} \approx 1.577$

26. ToDo: Proofs in congestion games

# 3 True False Questions

1. False. The point here is dominant strategy also includes **weakly dominant strategy**, therefore even one strategy is weakly dominate, other strategy can also be a NE.
2. True
3. True
4. True
5. True
6. True
7. False
8. True
9. False: The intuition is that pareto optimal only says you cannot improve your state without harming others. It doesn't say anything about how much you improve vs how much you harm others.
10. True
11. False