Assignment #1

Game Theory and Its Applications

Requirement 1-1 & 1-2

Contents

- Simulating the execution of graph games based on your student ID

Student ID mod 6	Game to simulate
0	Multi-Domination Game
1	k-Domination Game
2	Maximal Independent Set (MIS) Game (Symmetric)
3	Asymmetric MIS Game
4	Weighted MIS Game
5	MIS-based IDS Game

Student ID mod 2	Game to simulate
0	Symmetric MDS-based IDS Game
1	Asymmetric MDS-based IDS Game

Multi-Domination Game [YC14]

- Players: node set $\{p_1, p_2, \ldots, p_n\}$
- Strategies: $c_i \in \{0 \text{ (OUT)}, 1 \text{ (IN)}\}$ for all p_i
- Utility functions (C: strategy profile)

$$u_i(C) = \begin{cases} \left(\sum_{p_j \in M_i} g_j(C)\right) - \beta & \text{if } c_i = 1 \\ 0 & \text{otherwise,} \end{cases}$$

$\beta > 0$: constant
M_i: closed neighbors of p_i

where

$$g_j(C) = \begin{cases} \alpha, & \text{if } v_j(C) \leq k_j \\ 0, & \text{otherwise} \end{cases}$$

$\alpha > \beta$: constant

where

$$v_j(C) = \sum_{p_k \in M_j} c_k$$

The number of nodes that dominate p_j

k-Domination Game [YC14]

- Players: node set $\{p_1, p_2, \ldots, p_n\}$
- Strategies: {0 (OUT), 1 (IN)}

If $|N_i| < k$, c_i must be 1

$$u_i(C) = \begin{cases} \boxed{\alpha \quad \text{if } |N_i| < k \text{ and } c_i = 1} \\ \sum_{p_j \in N_i} g_j(C) - \beta & \text{if } |N_i| \geq k \text{ and } c_i = 1 \\ 0 & \text{otherwise} \end{cases}$$

where

$$g_i(C) = \begin{cases} \alpha, & \text{if } c_i = 1 \text{ and } v_i(C) \leq k \\ 0, & \text{otherwise} \end{cases}$$

$\boxed{\alpha > \beta > 0}$

where

$$v_i(C) = \sum_{p_j \in N_i} c_j$$

N_i (not M_i): p_i's open neighbors (p_i excluded)

不含P

Maximal Independent Set (MIS) Game (Symmetric) [YHT16]

- Players: nodes p_i's
- Strategies: $c_i \in \{1\ (\text{IN}), 0\ (\text{OUT})\}$
- Utility functions:

$$u_i(C) = \sum_{p_j \in N_i} \omega(c_i, c_j) + c_i$$

N_i: p_i's open neighbors

where

$$\omega(c_i, c_j) = -\alpha c_i c_j$$

$\boxed{\alpha > 1\text{: constant}}$

$$BR_i(c_{-i}) = \begin{cases} 0, & \text{if } \exists p_j \in N_i, c_j = 1 \\ 1, & \text{otherwise.} \end{cases}$$

Asymmetric MIS Game [YHT16]

- Player's utility

$$u_i(C) = \sum_{p_j \in L_i} \omega(c_i, c_j) + c_i$$

L_i: p_i's neighbors that have equal or higher priority

where

$$\omega(c_i, c_j) = -\alpha c_i c_j$$

$\boxed{\alpha > 1\text{: constant}}$

$$BR_i(c_{-i}) = \begin{cases} 0, & \text{if } \exists p_j \in L_i, c_j = 1 \\ 1, & \text{otherwise.} \end{cases}$$

Players only care neighbors that have priority equal to or higher than theirs

Weighted MIS Game [YHT16]

- Each node has a weight and we want to maximize the total weight in the MIS
- One approach: using priority
- Possible priority functions:

$$\frac{W(p_i)}{\deg(p_i)+1} \qquad\qquad \frac{W(p_i)}{\mathrm{W}(p_i)+\sum_{p_j \in N_i} W(p_j)}$$

MIS-based IDS Game [YS18]

- p_i's utility:

$\alpha > 1$

$$u_i(C) = c_i\left(1 - \alpha \sum_{p_j \in L_i} c_j\right)$$

L_i: set of p_i's neighboring node p_j with $\deg(p_j) \geq \deg(p_i)$.

prefer nodes with higher node degrees

- Best response of p_i

$$BR_i(c_{-i}) = \begin{cases} 0, & \text{if } \exists p_j \in L_i, c_j = 1 \\ 1, & \text{otherwise.} \end{cases}$$

Symmetric MDS-based IDS Game [YS18]

- Let $M_i = N_i \cup \{p_i\}$. Define $$v_i(C) = \sum_{p_j \in M_i} c_j$$
- Let $\alpha > 1$ be a constant. Define $g_i(C)$ as
$$g_i(C) = \begin{cases} \alpha & \text{if } v_i(C) = 1 \\ 0 & \text{otherwise,} \end{cases}$$
- Let $\gamma > n\alpha$ be a constant. Define
$$w_i(C) = \sum_{p_j \in N_i} c_i c_j \gamma,$$
- Let $0 < \beta < \alpha$. p_i's utility:

gain of dominance

penalty of violating independence

$$u_i(C) = \begin{cases} \boxed{\left(\sum_{p_j \in M_i} g_j(C)\right)} - \beta - \boxed{w_i(C)} & \text{if } c_i = 1 \\ 0 & \text{otherwise,} \end{cases}$$

Asymmetric MDS-based IDS Game [YS18]

- Let $M_i = N_i \cup \{p_i\}$. Define

$$v_i(C) = \sum_{p_j \in M_i} c_j$$

- Let $\alpha > 1$ be a constant. Define $g_i(C)$ as

$$g_i(C) = \begin{cases} \alpha & \text{if } v_i(C) = 1 \\ 0 & \text{otherwise,} \end{cases}$$

- Let $\gamma > n\alpha$ be a constant. Define

$$w_i(C) = \sum_{\boxed{p_j \in L_i}} c_i c_j \gamma.$$

only care neighbors with higher degrees

- Let $0 < \beta < \alpha$. p_i's utility:

$$u_i(C) = \begin{cases} \left(\sum_{p_j \in M_i} g_j(C)\right) - \beta - w_i(C) & \text{if } c_i = 1 \\ 0 & \text{otherwise,} \end{cases}$$

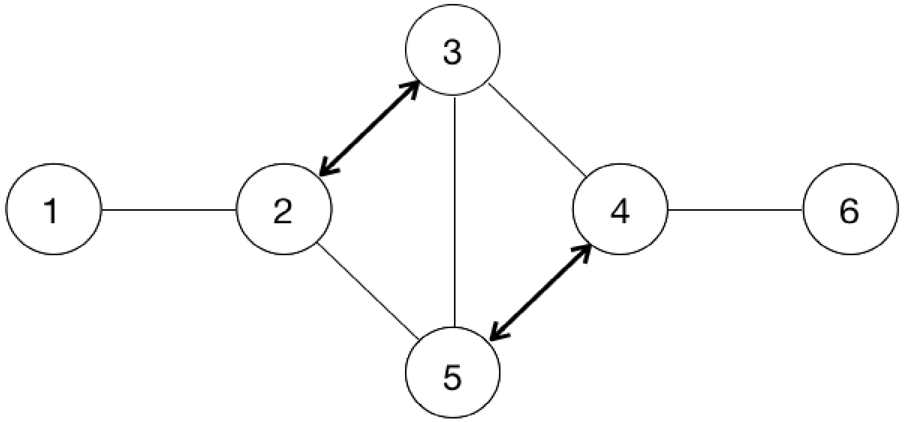
Requirement 2

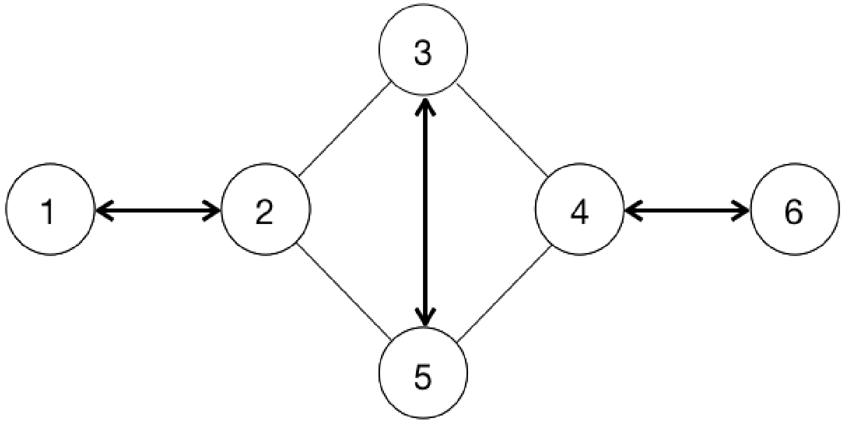
Contents

- In this assignment, you could reuse your codes for homework#1
- You have to define a utility function by yourself for the maximal matching problem
  https://en.wikipedia.org/wiki/Matching_(graph_theory)

Maximal Matching

- Both are maximal matching (one with 2 matched pairs and the other with 3 matched pairs)





Matching Game

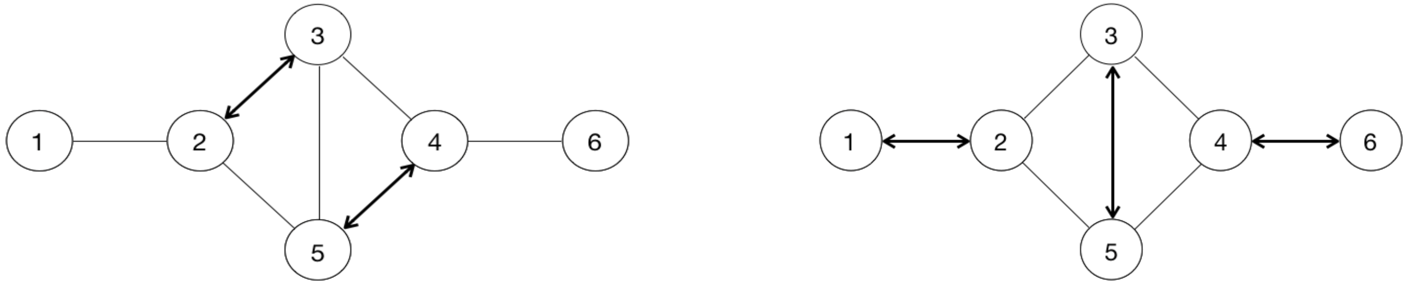
- Given a graph $G = (P, E)$, where P is the vertex set (with n vertices) while E is the edge set
- Each player p_i is a node in P
- Strategy set of each player $p_i \in P$: $N_i \cup \{null\}$, where N_i is p_i's open neighbors and $null$ indicates unmatched
- Let c_i be p_i's strategy and $C = (c_1, c_2, \cdots, c_n)$ be a strategy profile. (p_i, p_j) is a *matched pair* if and only if $c_i = p_j$ and $c_j = p_i$.
- p_i is *unmatched* if $c_i = null$

Design Goal

- Define a utility function $u_i(C)$ for each player $p_i \in P$ such that
  - Starting from any strategy profile, ensure that any player's strategy change as his best response eventually ends up with a NE
  - Ensure that every NE is a maximal matching
    - It's a matching because ① $c_i = p_j$ whenevr $c_j = p_i$ ② $c_i = null$ otherwise
    - It's maximal because there exists no two players that are unmatched but could match with each other

Design Goal (Optional)

- A heuristic to increase the number of matched pairs is to give high priority to nodes with few neighbors in matching
- Example



- Try to integrate this concept into utility definition
- This part is not mandatory (only for those who are interested)

Pseudo Code for Game Simulation

```
randomize initial game state
move_count = 0
while the game does not reach NE
  randomly pick up one player who can improve its utility
  change this player's strategy to its best response
  move_count++
end while
verify that the game state is a valid solution
output game state and move_count
```

Performance Measurements of The Result

- Except weighted MIS game, the quality of the result can be measured by the number of elements in the set
  - We want to minimize the number of elements in a dominating set
  - We want to maximize the number of elements in an independent set
- We want to maximize the total weight in the weighted MIS game
- Besides, we want to minimize the number of player's movements (i.e., `move_count`)

Pseudo Code for Performance Evaluation

Topology: the WS model ($n = 30, k = 4$)
Adjustable parameter: p_r (0 to 0.8 step 0.2)

```
repeat every adjustable topology parameter
  repeat 100 times
```

Code for Game Simulation

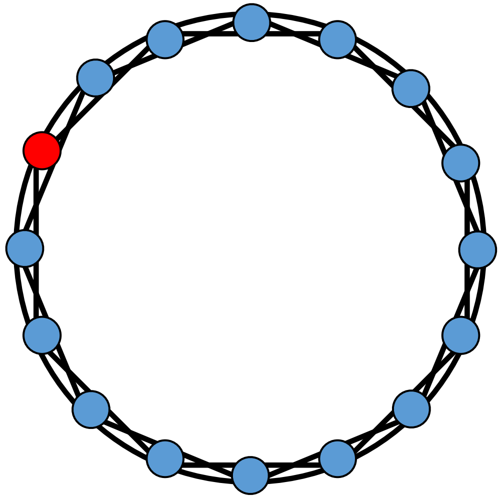
```
  calculate the averaged set cardinality and move_count
plot the results using x-y figures
```

Simulation Environment

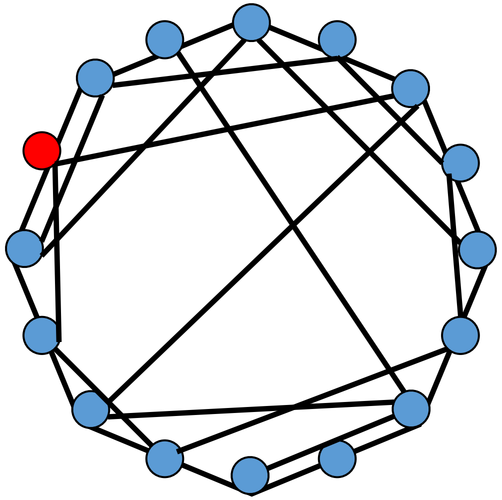
The WS Model [WS98]

- an n-node regular graph is first formed
  - each node has k edges connecting to its k nearest neighbors

$n = 16$
$k = 4$

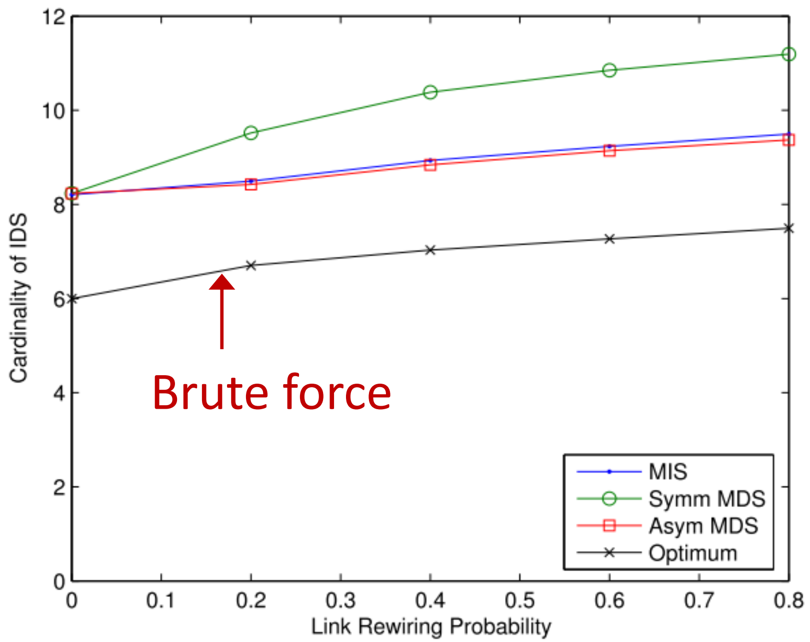


- rewire every edge to a randomly selected node with probability p_r



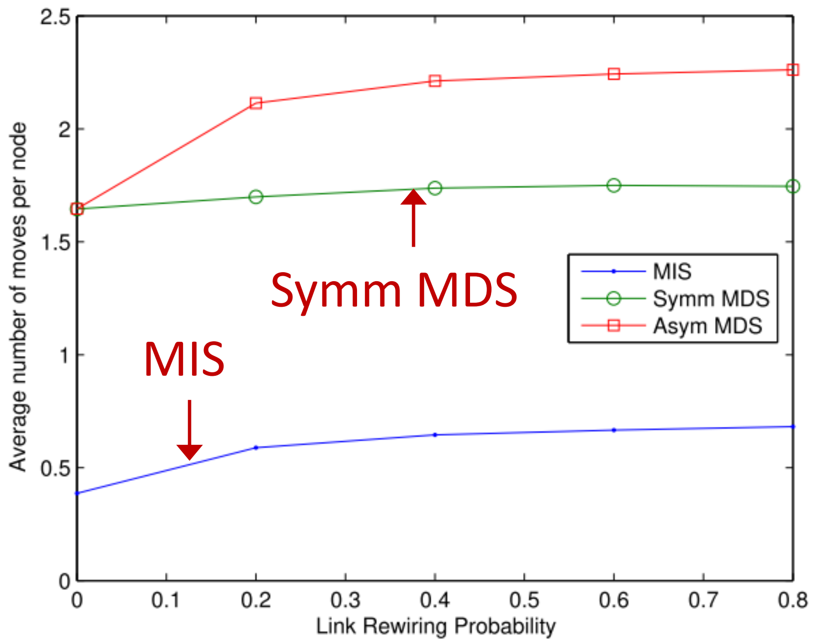
Sample Result (for Requirement 1-1 & 1-2)

Avg. size of IDS



Link rewiring prob.

Avg. #moves per node



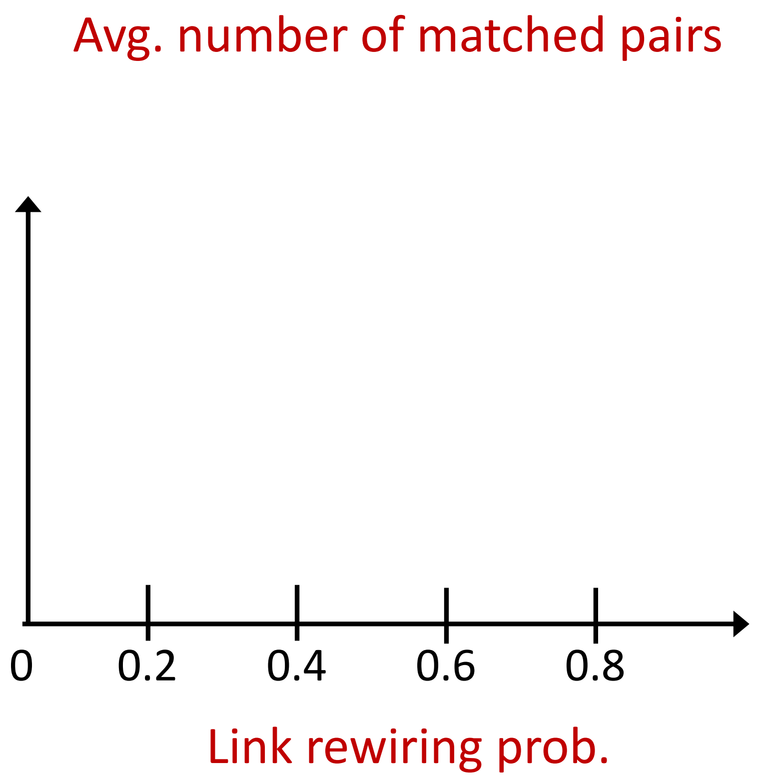
Link rewiring prob.

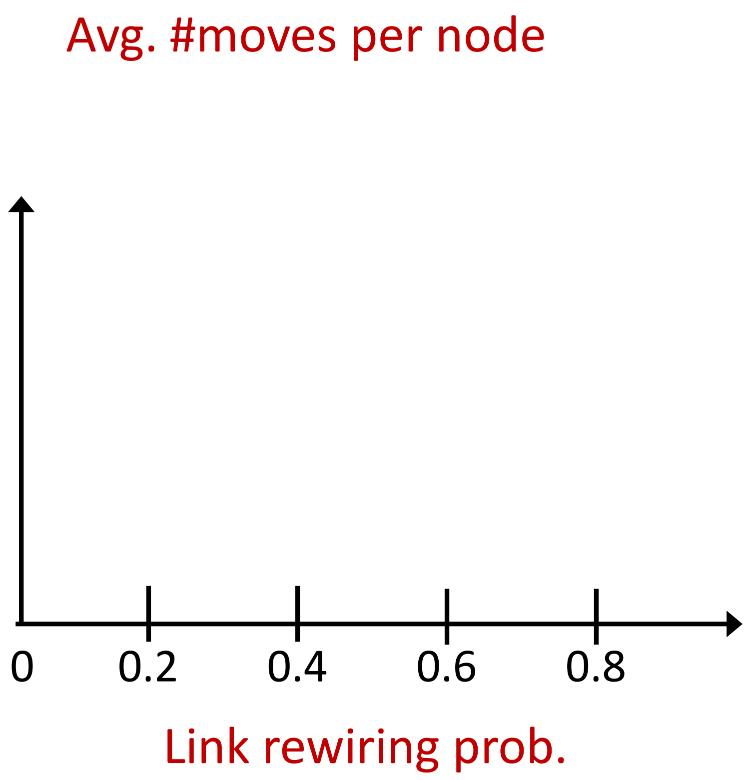
$n = 30, k = 4$

References (Requirement 1-1 & 1-2)

- [YC14] L.-H. Yen and Z.-L. Chen, “Game-theoretic approach to self-stabilizing distributed formation of minimal multi-dominating sets,” *IEEE Trans. on Parallel and Distributed Systems*, 25(12): 3201-3210, Dec. 2014.
- [YHT16] L.-H. Yen, J.-Y. Huang, and V. Turau, “Designing self-stabilizing systems using game theory,” *ACM Trans. on Autonomous and Adaptive Systems*, 11(3), Sept. 2016.
- [YS18] L.-H. Yen and G.-H. Sun, “Game-theoretic approach to self-stabilizing minimal independent dominating sets,” *The 11th Int'l Conf. on Internet and Distributed Computing Systems* (IDCS 2018), Tokyo, Japan, Oct. 2018.
- [WS98] D.J. Watts and S.H. Strogatz, “Collective Dynamics of ‘Small-World’ Networks,” *Nature*, vol. 393, pp. 440-442, June 1998.

Sample Result (for Requirement 2)





$n = 30, k = 4$

References (Requirement 2)

- Maximal Matching: https://en.wikipedia.org/wiki/Matching_(graph_theory)
- [WS98] D.J. Watts and S.H. Strogatz, "Collective Dynamics of 'Small-World' Networks," *Nature*, vol. 393, pp. 440-442, June 1998.