MB-CIM: A Multi-round Budgeted Competitive Influence Maximization

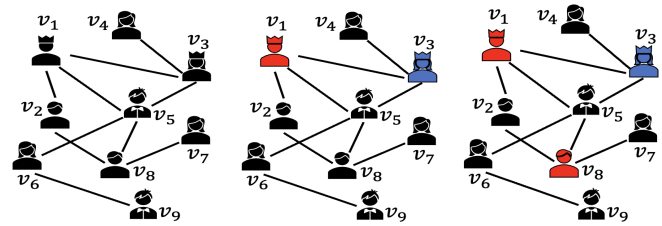
Nadia Niknami and Jie Wu
Department of Computer and Information Sciences
Temple University, Philadelphia, PA, USA
(nadia.niknami, jiewu)@temple.edu

***Abstract*—Social networks often serve as a medium for the diffusion of ideas or innovations. The maximizing influence spread through a social network has attracted significant research interest recently. Influence maximization is trying to select a small set of seed users in the social network to maximize the spread of influence. An individual's decision to adopt a product or innovation will be highly dependent on the choices made by the individual's peers or neighbors in the social network. Competitive Influence Maximization (CIM) addresses the competition where multiple competing sources propagate in the same network. Competitors need to decide which nodes in the given social network would be an influential one and how many resources should be allocated to the potential social network member so that identifying the best algorithm for the influence maximization under budget constraint has become a demanding task. Understanding, predicting, and controlling social influence and its diffusion is an exciting topic in social network analysis. Most previous works on CIM focus on the same budget allocation for different seed nodes. Also, they consider a single-shot game where competitors select potential members in one round. We are interested in multi-round CIM where each competitor needs to decide the location and the amount of budget to invest in the most influential members simultaneously and repeatedly under a given total budget. The object of competitors is maximizing the total number of activated nodes. This paper proposes a tree-approximate game-theoretical framework and introduces the new measurement as a dynamic node weight. We demonstrate through simulation that our approach works well in a multi-round and learning-based CIM problem.**

***Index Terms*—budget allocation, game theory, reinforcement learning, social networks, multi-round influence maximization.**

I. Introduction

Viral marketing [1] is one of the most effective marketing tactics in advertising. Facebook and YouTube are two social networks that help promote products [2]. Influence of social networks among individuals plays an essential role in viral marketing. The growth of online social networks has enabled them to spread quickly [3]. Influence maximization is one of the most fundamental algorithms in social influence analysis. Over the last decade, significant effort has been put into the development of efficient algorithms for influence maximization [4]–[10]. The main objectives in the IM problem are to discover which potential members, seed set, to select and how many resources to allocate to these potential members to maximize the competitors' influence. In the IM problem, all the nodes in seed set ξ are activated directly, and the remaining nodes are inactive. According to some probability distribution,



(a) $S = \{v_1, v_2, v_3\}$ (b) No winner in v_3 (c) Red wins v_3

Fig. 1: Budget allocation in case of tie in the CIM.

when a node is activated at timestamp i, it may activate its out-neighbors at timestamp $i+1$. When no node can activate any other node, the influence propagation ends. In the real world, there are many competitors at the same time implementing their strategies to find a considerable influence on the same social network. Actually, each rational player tries to spread its influence as maximal as possible and make its opponents as minimal as possible. That is why Competitive Influence Maximization (CIM) [11]–[15] has received a lot of attention recently. A CIM problem involves selecting the most effective seeds based on decisions made by other competitors in order to maximize their influence. The CIM model allows the influences of each player to cascade simultaneously throughout the social network, which can interfere with each other.

Considering a competitive game with two competitors, Red and Blue, in the given social network $\mathcal{G}(V, E, P)$, where V is the vertex set, and E is the edge set. P is a set of edge propagation probabilities, where $p(u, v)$ represents the influence probability of the edge between u and v, where $\sum_u p(u, v) < 1$. When there is no edge between u and v, $p(u, v) = 0$. The given social network is defined as a network of connections and interactions among entities. Nodes can take on one of the following states: activated by Red, activated by Blue, and inactive. First, competitors identify the nodes with the most influence. They compete for only these influential nodes based on the amount of budget each of them allocates to each node. After activation of a node, its influence propagates with a certain probability to their not yet activated neighbors. At each step t, each node u activated at step $t-1$ activates its neighbor v with probability $p(u, v)$. Once activated, they stay activated. Influence maximization under both *independent cascade* (IC) [4] and *linear threshold* (LT) [16], [17] models are NP-hard. These propagation models satisfy

TABLE I: Main notations

Symbol	Meaning
B_1/B_2	Total budget of player $1/2$
$B_1(u)/B_2(u)$	Allocated budget of player $1/2$ on node u
T	Total number of rounds
$N(u)$	Neighbor set of node u
V	Set of nodes in the network
V^1/V^2	Set of activated nodes by player $1/2$
$w(u)$	Weight of node u
$w'(u)$	Estimated total influence weight of node u
$p(u, v)$	Influence probability of edge between u and v
$R(u, v)$	Influence value of the MRIP between u and v
ξ	Seed set
s	State of network in reinforcement learning
$\pi(s)$	Policy in state s
r_t	Reward in round t in reinforcement learning
a_1/a_2	Player $1/2$'s action
$\mathcal{V}(s)$	Value of state s in reinforcement learning

two important properties, submodularity, and monotonicity, in terms of their influence spread function. We will use IC in this paper. The key characteristic of this model is that diffusion events associated with every edge in the given social graph are mutually independent, and the success of the seed node u to influence one of its inactive neighbors v only depends on the propagation probability of the edge from u to v.

Consider the social network in Fig. 1(a). Players Red and Blue compete over the nodes of this network. These players select v_1, v_3 , and v_8 as the most influential members in this network. Red player allocates $(\$2, \$2, \$2)$ and Blue player allocates $(\$1, \$3, \$2)$ on members v_1, v_3 , and v_8, respectively. The winning probability is proportional to the budget allocation of two parties. Red player wins v_1 with the probability of $2/(1 + 2) = 2/3$. Blue player wins v_3 with the probability of $3/(3 + 2) = 3/5$. Players have the same budget allocation on v_8. Fig.1(b) presents the result of this competition until this step. If player Red adds more money, say \$1 extra on his investment over v_8, his chance to win this node will be $3/2 + 3 = 3/5$. By doing so, he wins v_8. After finding these seed nodes, the propagation process which is based on the influence probability of relations or links between seed nodes and their friends in the given network will start. The player finding the maximum number of influenced members would be the winner of this game.

Such a scenario can be modeled by the multi-round Competitive Influence Maximization. The goal of each player in the competitive environment is to find an optimal combination of strategies to utilize their budget efficiently. The idea is to take a more realistic and practical setup, rather than selecting seeds only in the first round. In a multi-round CIM, players keep selecting seed nodes according to the current network state and the expected reactions of other players within given rounds. In addition, each player can spend a limited amount of budget in all rounds on seed nodes. Nodes with the greatest influence in a given network are selected according to different strategies. In each round, players choose a seed node ξ_t, decide the amount of budget that should be allocated to this seed node, then wait until the end of the propagation process. This assumption can be extended to multiple seed nodes in each round. Note that during each round, players take action simultaneously, but there are sequence rounds (Fig. 2). As influence maximization is NP-hard, we introduce a new notation of Most Reliable Influence Path (MRIP) as an approximation.

The value of influencers varies, and competitors want to find the best value for their overall social advertising budget. It is obvious that an equal budget at each round does not sufficiently model the willingness to choose a cost-efficient seed set. Indeed, we see that the choice to use a fraction of the budget for round t is crucial: a too large budget allocation translates into a waste of budget, and a too small budget allocation translates into a waste of time (a whole round is used to influence only a few users). To circumvent this issue, instead of a budget per round, in our framework, we allow the agent to have the competition of the most influential nodes at each round under an overall budget constraint. In this paper, compared to the conference version [18], we make a set of extensions in the case of explaining the approach and evaluation. We evaluate our proposed approach under different parameters such as different amount of total budget, various network structures, different densities, and different competition strategies. The contributions of this paper are summarized as follows:

- We define a new measurement called dynamic weight for nodes. Considering both fixed and dynamic weights in selecting seed nodes helps players have a more accurate selection.
- We discuss the influence spread in the social network by considering the Most Reliable Influence Paths (MRIP) for each node in the process of seed selection as an approximation. MRIP is inspired by the notion of a critical path in the scheduling community.
- We consider three new features maximum weight of inactive nodes, the ratio of budget, and the weight of nodes, in case of reachability to describe the state of the network in reinforcement learning.
- We propose a CIM model which selects the winner of the node in case of breaking tie based on the budget proportion, rather than randomly. Players can compete on the given node by increasing their investment in this node to increase their chance.
- We evaluate the effect of our model experimentally using real datasets and some synthetic ones.

Organization. The remainder of the paper is organized as follows. Section II briefly surveys the related works. In Section III, we describe some preliminaries. Section IV presents details of seed selection, budget allocation, and our proposed algorithm. Section V demonstrates experiment results on the proposed model in the case of different important parameters. Finally, Section VI offers conclusions and some directions for future work.

II. Related work

In this section, we review related research efforts on the CIM problem, which analyzes the implications of competing products interfering with each other. In addition, we review some reinforcement approaches in the CIM problem.

A. Competitive Influence Maximization

Competitive IM aims at finding strategies that maximize one's influence while minimizing his opponents' influence in a social network [19] [20]. There are different extensions of the IC model and the LT model to accommodate multiple competing ideas in social networks instead of focusing on spreading a single ideas [9] [11] [21]. Li *et al.* [22] consider a model for competitive IM. According to a graph $\mathcal{G}$ and diffusion model, the strategy space comprises all IM algorithms that players can adopt. For each player, the objective is to find a Nash equilibrium strategy that maximizes his own influence. In [23], authors addressed a multi-stage version of the Influence Maximization problem. They provided a new formulation and compared their approaches in terms of accuracy and computation run time.

B. Reinforcement Learning

An important line of work that uses RL to solve NP-hard optimization problems on graphs is [24] [25]. Lin *et al.* in [26] model a multi-party CIM problem and propose a different model with the help of RL and based on the Multi-Round CIM method. Authors in [27] propose a novel deep RL-based framework to tackle the MRCIM problem considering the network community structure under a quota-based ϵ-greedy policy. K. Ali *et al.* [28] propose a deep reinforcement learning-based model to tackle the CIM on unknown social networks. In [29], by using automatically learned node and graph representations that encode important network structural properties, H. Kamarthi *et al.* propose a RL framework for discovering effective network sampling heuristics. K. Ali *et al.* in [30] propose a novel RL-based framework that is built on a nested Q-learning algorithm. They derive the optimal solution in both budget allocation and node selection that results in the maximum profit with time constraints.

C. Resource Allocation Against Opponents

Parties in a competitive influence maximization problem perform like a player in a Colonel Blotto game. Colonel Blotto games (CBG) are a class of two-player zero-sum games, in which both players need to allocate limited resources over several objects simultaneously. Authors in [19] focused on competitive influence when players need to decide on resource allocation against their opponents. They proved that competition's price is unbound in such a Colonel Blotto game. Authors in [31] address the budget allocation scenario in maximization influence problem. Companies can allocate different budgets to nodes in the network, and nodes will be attracted to companies whose products offer a higher value. In this case, companies compete by allocating a certain amount of budget to each node in the network. A Nash equilibrium-based model is proposed by Masucci *et al.* [13] to compete for obtaining more customers in online social networks. Unlike most of the existing works, in this paper, we study the problem of Multi-round Competitive Influence Maximization within budget constraints and while considering the remaining budget of opponents. We consider a different approach from the Blotto game for budget allocation strategy. There is a dependency between targets, and players can continue their investment in case of tie-breaking. In addition, there is propagation after any activation. In comparison with ML approaches, we consider new features to describe the state of the network.

Fig. 2: A Multistage game with two competitive players.

III. Preliminaries

A social network can be modeled with a weighted and directed graph $\mathcal{G}(V, E, P, W)$, where we define W as a set of weights associated with each vertex in V. Activating a node u in $\mathcal{G}$ means accepting an idea from the player i. Once a node u accepts the idea of being occupied by a player i, it cannot change occupation to another party. If the given node does not accept any idea, it means that the state of the node u is inactive.

A. Competitive Influence Maximization.

In a multi-stage CIM problem, competitors need to select seed nodes simultaneously in each of the sequence stages. Suppose that there is a CIM game with two players, 1 and 2, and n nodes in a social network $\mathcal{G}$. Player 1 has a budget of size B_1, and player 2 has a budget size of B_2. Each node u has a value, $W(u) > 0$, which can be regarded as the reward of taking this node for players. The total value of n nodes in this social network is $W = \sum_{u \in V} W(u)$. The winner of this game would be the player who can obtain the most reward by influencing the more important nodes. Players have competition with the amount of budget they allocate in seed nodes (the most influential nodes).

In this game, three types of competition can occur. The first competition is players' competition on seed nodes by the amount of allocated budget, which can be called Node-Node competition. The second one is Link-Link, which is the competition of influence when two different links with different influences try to activate the given node in their favor. The last one is Node-Link. This will happen when one of the competitors allocates some budget on the given node, and the influence of another competitor reaches this node by the influence of the link.

1) Nod-Node influence competition: Considering a Node-Node competition on the node u. Suppose that $B_1(u)$ and $B_2(u)$ are the amount of budget that players 1 and 2 have

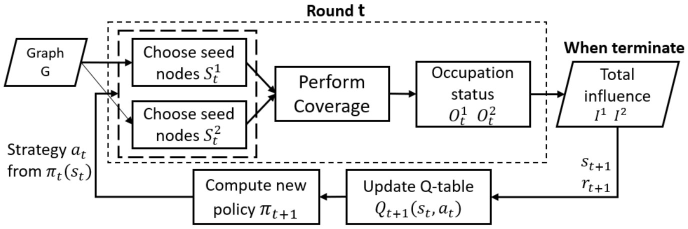


Fig. 3: Approach during training process.

allocated to node u. The winning probability of player 1 for this competition is as follows:

$$\frac{B_1(u)}{B_1(u)+B_2(u)} \tag{1}$$

2) Link-Link influence competition: Link-Link influence competition will happen after the budget allocation process and determining the winner of this stage in the case of taking the given seed node. During the propagation process, suppose that node u has the influence of player 1 from one of its neighbors with $p_1 = p(v,u)$. In addition, node u has influence of player 2 from another neighbor, node w, with $p_2 = p(w,u)$. The probability that node u would be activated by player 1 is as follows:

$$\frac{p_1}{(p_1+p_2)} \times (1-p_1p_2), \tag{2}$$

where $(1-p_1p_2)$ considers the probability of activation of node u by at least one of the players. The probability that node u would be activated by player 2 is as follows:

$$\frac{p_2}{(p_1+p_2)} \times (1-p_1p_2) \tag{3}$$

3) Node-Link influence competition: In a multistage competition, competitors are able to allocate a budget at the same moment at the beginning of each stage rather than during the stage. At the beginning of each stage, competitors decide on their budget allocation, then influence propagation starts. At the end of the propagation, competitors can start the next stage and make a decision about new budget allocation. Therefore, there is node-node competition at the beginning of each stage and link-link competition during each stage. Consequently, we will avoid considering the link-node competition for the multistage CIM problem.

B. Multi-agent Reinforcement Learning.

In sequential games, players need to look forward and reason back to find the best decision. In simultaneous games, players look for the best response when they cannot see the other side's strategy. Therefore, players need to learn more about the strategies of opponents. Reinforcement learning (RL) is a subfield of machine learning that addresses the problem of learning optimal decisions over time. In RL, the agent keeps interacting with the environment to find the optimal policy π to maximize his expected accumulated rewards [32]. The goal of an RL is to learn a policy $\pi(s)$ to determine which action to take given a specific environment represented by state s.

The reward obtained by an agent should reinforce his behavior. Reward reflects the success of the agent's recent activity and not all of the successes achieved by the agent so far. The agent's objective is to learn the policy that maximizes the expected value of the return. The return is the measure of future cumulative reward during the rounds.

$$r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + ... = \sum\nolimits_{k=0}^{\infty} \gamma^k r_{t+k+1}. \tag{4}$$

RL formulates the expected accumulated rewards of a state which is called the $\mathcal{V}$ function. Also, it formulates the expected accumulated rewards for each state-action pair which is called the Q function. Q function estimates how efficient the policy π is at maximizing the accumulated reward r_t. The $\mathcal{V}$ function $\mathcal{V}^\pi(s)$ associated with a policy π tells the agent how good the policy is. The state-value function is defined as:

$$\mathcal{V}^\pi(s) = E_\pi\{r_t|s_t = s\} = E_\pi\{\sum\nolimits_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t = s\}, \tag{5}$$

where γ is the discount factor. The action-value function $Q(s,a)$ is expected return starting from action a in state s, and then following policy π:

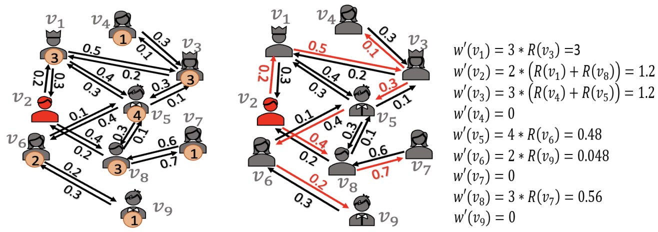
$$\begin{aligned} Q^\pi(s,a) =& E_\pi\{r_t|s_t = s, a_t = a\} \\ =& E_\pi\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}|s_t = s, a_t = a\}. \end{aligned} \tag{6}$$

The state value and action value in equations (5) and (6) can be learned through the interaction of agents with the environment. The optimal policy $\pi(s)$ can be obtained given the Q function and find the maximum value. Fig. 3 displays the details of RL for a multi-round CIM. According to this diagram, at the end of each round, players can see the result of the competition in terms of reward and the current state. Then, they update their learning, compute new policy against the opponent's strategy, and select a new seed set. In algorithm 1, we can see this process step by step. In the case of multi-agent RL, agents learn the policies through experience in the environment and interaction with each other. We assume that there are only two parties that compete with each other. We need first to define the environment, the reward, the action, and the state.

Algorithm 1 RL

```
Q(s, a) ← initial value
while training is not terminal do
    s_t ← s_0
    while s_t is not a terminal state do
        Determine Q_t(s_t, a_t)
        Take strategy a_t based on Q_t(s_t, a_t)
        Simulate opponent's action
        Propagate influence to obtain reward r_{t+1}
        Compute next state s_{t+1} based on network features
        Update Q(s_t, a_t)
        s_t ← s_{t+1}
```

In the given social network, considering V^1 and V^2 as the total number of activated nodes by players 1 and 2 respectively. The multi-round CIM can be considered as a zero-sum game



(a) Original graph (b) Constructing $\mathcal{T}_{v_2}$ (c) Calculating w'

Fig. 4: Computing shortest paths and influence weights.

TABLE II: Computing R(v) from source node v_2

A	N(A)	R(s)* p(s,v)	R(v)
$\{v_2\}$	v_1	$1 * 0.2 = 0.2$ **1*0.4=0.4**	$R(v_8)$
$\{v_2, v_8\}$	v_1 v_5 v_7	$1 * 0.2 = 0.2$ $0.4 * 0.1 = 0.04$ **0.4*0.7=0.28**	$R(v_7)$
$\{v_2, v_8, v_7\}$	v_1 v_5	**1*0.2=0.2** $0.4 * 0.1 = 0.04$	$R(v_1)$
$\{v_2, v_8, v_7, v_1\}$	v_1 v_5	$0.4 * 0.1 = 0.04$ **0.2*0.5=1** $0.2 * 0.4 = 0.08$	$R(v_3)$
$\{v_2, v_8, v_7, v_1, v_3\}$	v_1 v_5	$0.4 * 0.1 = 0.04$ $0.2 * 0.4 = 0.08$ $1 * 0.1 = 0.1$ **1*0.3=0.3**	$R(v_5)$
$\{v_2, v_8, v_7, v_1, v_3, v_5\}$	v_4 v_6	$1 * 0.1 = 0.1$ **0.3*0.4=0.12**	$R(v_6)$
$\{v_2, v_8, v_7, v_1, v_3, v_5, v_6\}$	v_4 v_9	**1*0.1=0.1** $0.12 * 0.2 = 0.024$	$R(v_4)$
$\{v_2, v_8, v_7, v_1, v_3, v_5, v_6, v_4\}$	v_9	**0.12*0.2=0.024**	$R(v_9)$

for players 1 and 2 since $(V^1 - V^2) + (V^2 - V^1) = 0$, where $V^1 - V^2$ and $V^2 - V^1$ are the goals of players 1 and 2, respectively. In such a game, the Nash equilibrium is guaranteed to exist with mixed strategies. The MINMAX theorem would be useful to find the equilibrium [33].

IV. Methodology

Traditional RL has been successful in dealing with multi-round CIM [26]. Nevertheless, this approach did not address the effect of budget on player seed selection strategy. In our approach, we integrate seed selection and budget allocation into the RL model. In the budget allocation phase, we consider convincing influential nodes to act as seeds, as well as selecting seed nodes. The player in this framework learns how to maximize the value of accumulated rewards by choosing the optimal policy π. The first step is identifying influential nodes within the network. Players then compete over only the selected nodes, rather than the entire network, depending on the budgets they allocate to each influential node. During each round t, the agent observes a set of features representing the network state $s_t \in S$, and selects one of the legal actions from the set a_t. In each round, the agent selects a seed set, $\xi_t \subset V$, based on its past observations. Note that ξ_t is the seed set selected by π at round t. The goal for the agent is to follow a learning policy π maximizing the total number of activated nodes. When no budget remains or no node can be added to the seed set ξ, the algorithm terminates.

Algorithm 2 Finding seed set by MRIP

```
S ← ∅
for all u ∈ V do
    w′(u) ← 0
for u ∈ V do
    Construct T_u via Alg. 3
    for each leaf v in reverse T_u do
        z ← parent(v)
        while v ≠ u do
            Compute w′(z) = w′(z) + R(v) × w(z)
            v ← z
            z ← parent(v)
new seed ← arg max_{u∈V/ S} w′(u)
S ← S ∪ {new seed}
V_A ← Activated nodes by new seed node
Constructing G′ with vertex set V − V_A
Recalculate T and w′ in G′
```

Algorithm 3 Computing $\mathcal{T}_u$

```
Require: G(V, E, P), source node u
A = {u}, R(u) = 1
while A ≠ V do
    Find node v ∈ N(A) and v ∈ V − A such that
    R′(v) = max_{(s,v):s∈A,v∈V−A} R(s) × p(s, v)
    R(v) = R′(v)
    A = A ∪ {v}
    Set s as the parent of v in spanning tree T_u
return T_u
```

Definition 1. (Budgeted Multi-round CIM) Given the network $\mathcal{G}$, each player chooses seed nodes in turn, and then influence propagation is performed at round t. Players compete based upon the budgets they allocate to the most influential nodes in order to win these nodes as seed sets. The objective of each player is to maximize its overall relative influence V^i after T rounds, where V^i is the difference among activated nodes of different players.

A. Selecting Seed Nodes and Propagation Model

Consider a static social network $\mathcal{G}$ and B_1 and B_2 for players as their fixed budget. Each round, one seed node will be chosen. The goal of each player is to reach and activate as many nodes as possible within their total budget. Each player can decide to implement the specific strategy to maximize its overall influence in $\mathcal{G}$. The strategy refers to how the player spends their budget on selecting the seed nodes at each round. Maximal influence with a spanning tree restricts node u's influence diffusion to a local tree structure rooted at u. The influence of a node in a tree can be calculated efficiently and precisely. Note that the conflict rule is slightly different from other works. In contrast to other approaches, which prioritize one of the players or select the winner of conflicting randomly, our approach allows players to increase their investment in case of tie-breaking. The winner will be determined with the help of budget proportion.

B. Most Reliable Influence Path (MRIP)

Since influence maximization is NP-hard, we use the idea of the critical path in the scheduling community. Following the style of Dijkstra and Prim's greedy algorithm, an inactive node will get a chance to become active only through the shortest path from the initially active nodes. In order to find the shortest path in a maximum influence problem, we can consider the maximum influence probability of edges. The distance between node u and v can be computed as the logarithm of the inverse of the influence probability of edge (u, v). Influence propagates through the most probable paths, and the notion of the Most Reliable Influence Path (MRIP) can be considered as an approximation. It is helpful to estimate the local influence of nodes for seed selection. The influence of each node when considering the most reliable paths that originate from the given node can be regarded as a new measurement for ranking nodes. This paper calls this value the weighted influence of each node u, $w'(u)$. Considering $R(v)$ as the influence value of the most reliable path on node v originated from the source u, we construct a spanning tree $\mathcal{T}$ with the most reliable paths helps us to find w' for all nodes.

In fact, Prim's algorithm allows us to determine the spanning-tree $\mathcal{T}_v$ rooted at v such that each node is reached from the source node v via MRIP. The value of $R(v)$ for any two nodes u and v in V is the value of the shortest path from node u to v, where $\mathcal{P}_{u,v} = (u = u_1, u_2, ..., u_m = v)$, and there is no duplicate nodes. The probability that node v is activated by u through the path $\mathcal{P}_{u,v}$ is calculated as $\prod_{i=1}^{m-1} p(u_i, u_{i+1})$. All nodes along the path from u to v need to be successfully activated, then node v would be activated. As an extension, to more efficiently compute the increased influence spread within the tolerance of error, we can use an influence threshold to filter out the insignificant maximal influence paths whose values are less than due to having a very small impact on the influence spread computation.

For all $v \in V$ in the $\mathcal{G}$, we need to find $\mathcal{T}$. For simplicity, we explain the process of computing w' just by considering node u as the tree's root node. Suppose that $R(u) = 1$, among all of the neighbors of node u finding the edge (u, v) with maximum $R(u) \times p(u, v)$ is the first step. This is a greedy algorithm. In each step, we consider all of the edges that the source of them is in the explored node set A, and its destination is in $V - A$. We continue this process until A includes all of the nodes in V. Algorithm 3 represents these steps in detail. After constructing the spanning-tree $\mathcal{T}_u$, we compute the influence weight of each node by traversing this tree reversely. For each node v, the parent of v is u, and the weight of node u, which is illustrated as the weighted influence, will be measured by:

$$w'(u) = \sum\nolimits_{\forall v \in V} R(v) \times w(u), \tag{7}$$

where $w(u)$ is the weight of node u and $R(v)$ presents the value of shortest originated from u to v. If we consider $w(u)$ as the *fixed weight* for node u, which can be show the importance of node in the case of degree or centrality, $w'(u)$ can be called *dynamic weight* of this node. Considering the example in Fig. 4, for any pair of nodes u and v, we need to find the maximum influence path from u to v and construct a spanning tree $\mathcal{T}$. Fig. 4 shows the process for node v_2. Table II presents some early steps of finding $R(v)$ for each nodes when v_2 is the source node. Using the calculated $R(v)$ and reverse traversing the $\mathcal{T}_{v_2}$ in Fig. 4(b), the influence weights of all nodes are shown in Fig.4(c). The intuition behind the proposed algorithm comes from Dijsktra's algorithm. We can prove the proposed algorithm can find the most reliable path correctly.

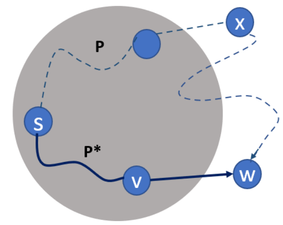


Fig. 5: Correctness of MRIP algorithm.

Theorem 1. If $\mathcal{T}_s$ is the spanning tree selected by MRIP's algorithm for source node s in the social network $\mathcal{G} = (V, E, P, W)$, then $\mathcal{T}_s$ is a most reliable influence tree rooted in s in $\mathcal{G}$ and $R(v)$ for each node $v \in V$ shows the influence value of the most reliable path on node v.

Proof: In Fig. 5, the gray area includes the explored nodes. Suppose that w is the next vertex added to $\mathcal{T}$ and P^* be the path from source s to destination w through node v. Considering any other path P from s to w, node x be the first node on path outside $\mathcal{T}$. Path P is already as long as P^* as soon as it reaches x by greedy choice. Thus, $R(w)$ is the length of the most reliable path from s to w. This completes the proof. ∎

Algorithm 2 presents the processes of selecting the seed node based on the influence weight of nodes. After finding $\mathcal{T}_v$ for each node v in algorithm 3, by considering $w(v)$ of nodes as the weight of node or ranking measurement in the case of the importance of node and $R(v)$ as the value of the most reliable path, the influence weights $w'(v)$ of all of the nodes can be calculated. The node with the highest w' would be selected as the seed node in each round. After choosing a seed node and propagating its influence, the next step is to recalculate $\mathcal{T}$ and the weighted influence of nodes in the graph $\mathcal{G}'$ with $V - V_A$ nodes, where V_A is the set of activated nodes. Therefore, after selecting any seed node and the propagation process, there are new w's for nodes. That is because we called this weight as *dynamic weight*.

The time complexity of Dijkstra's algorithm is $O(|E| \cdot log|V|)$, but here we need to find the shortest paths for all pairs of nodes. Now, the time complexity becomes $O(|E|^2 \cdot log|V|)$. After selecting a seed node, we need to remove the activated nodes, V_A, from $\mathcal{G}$ and consider a new social network $\mathcal{G}'$ including the set of nodes $V - V_A$ to recalculate $\mathcal{T}_v$ for each

TABLE III: Q-table for strategies

State	Budget	Q-Value
33311001	Unit	0.7
33311001	All	0.2

(a) Budget-allocation

State	Seed	Q-Value
333110010	Degree	0.26
333110010	Weight	0.24
333110010	MRIP	0.3
333110010	Compete	0.6
333110011	Degree	0.26
333110011	Weight	0.24
333110011	MRIP	0.3
333110011	Compete	0.6

(b)Seed-selection

node v as well as new weight w'. Therefore, the total number of nodes in these paths should be considered in the algorithm's time complexity as well.

C. Reinforcement Learning Settings.

As we are considering a multi-round scenario, the opponent's past decisions can be taken into account, but the opponent's future decisions are not known. There are several parameters we need to define in order to implement reinforcement learning. The propagation of influence is treated as an environmental effect, whereby activated nodes spread their influence to their neighbors and activate new ones. The reward we receive after T steps is the number of nodes that have been influenced in the entire graph. Through Q-function updates, rewards are propagated back to previous states.

Action. Players can allocate different amounts of budget to nodes in $\mathcal{G}$. Competition is based on how much budget each player allocates to each node. The possible actions are allocating budget on new seed nodes or feeding an activated seed node to increase its influence on neighbors. We use the idea of meta-learning [26] [30] in RL. We consider the following actions: (1) Selecting a new seed node and (2) feeding a node in case of tie. Selecting seed nodes can include Max-degree, Max-weight, Centrality, Randomly, Voting, and learning-based strategies. In case of investment, we consider investing \$1 or all of the remaining budget.

State. In order to represent the network and environment status, we must model the state. The design of features will reflect both the current status of the network and the current occupation status. Correlations with rewards, the choice of actions, and the condition of networks require certain features. Below are the features we have designed:

1) Number of inactive nodes
2) Summation of degrees of all inactive nodes
3) Maximum degree among all inactive nodes
4) Summation of the weight of the edges for which both vertices are inactive
5) Summation of the inactive out-edge weight for nodes which are the neighbors of player i
6) Maximum sum of the inactive out-edge weight of a node among all nodes
7) Ratio of budgets
8) Weight of nodes in case of reachability

Features 1 to 5 help players find the condition of network in terms of the status of nodes as well as the weight of edges.

TABLE IV: Social Networks

Name	**Nodes**	**Edges**	**Description**
Facebook	4,039	88,234	Facebook social network
Ca-HepTh	9,877	51,971	Arxiv High Energy Physics
Cit-HepPh	620	827	Paper citation network
Synthetic	100	500	Randomly generated network

TABLE V: Evaluation of different features

Dataset	Reward		Dataset	Reward	
Facebook	OPT-F6	%49	Synthetic	OPT-F6	%45
	OPT-F7	%52		OPT-F7	%53
	OPT-F8	%55		OPT-F8	%50
	OPT-F6F7	%58		OPT-F6F7	%48
	OPT-F7F8	%56		OPT-F7F8	%58
	OPT-F6F8	%58		OPT-F6F8	%51
	OPT	%65		OPT	%68

Features 6, 7, and 8 are new ones to describe the states of the network. These features help players to learn more about the environment, as well as the opponent's strategy. As a result of the dependence between some features, not all combinations of states are possible. There is a correlation between the candidate strategies we use to choose our actions and these features. The player continually updates both Q-tables, that is, seed-selection, and budget-allocation Q-tables, during the training. Meanwhile, it updates its policy throughout the training in order to find an optimal policy for budget utilization from budget-allocation and seed-selection Q-tables.

V. Experiments

We conducted experiments to evaluate the efficiency of the proposed models in terms of influence spread to other algorithms. Also, we evaluate our algorithm for different datasets with different densities.

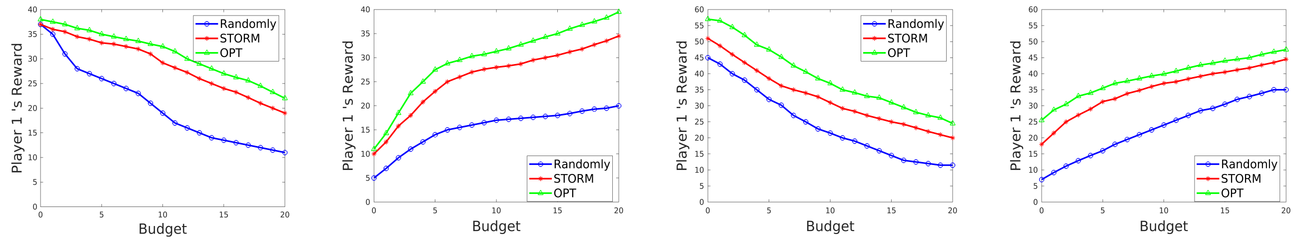
A. Experiment Setup

We used the igraph Python library to represent the graphs and the shortest path calculations. The datasets consist of two real-world social networks and two synthetic ones. We used the IC as the diffusion model. The edge weights are set randomly in a range between 0 and 1. In order to check the impact of influence propagation, we consider normal distribution, with the same $\mu = 0$ and different σ^2. We train the model by doing 1000 runs and then selecting the best result as the final result of the model. We use the random tree generation algorithm, as discussed in experimental settings.

B. Comparison Methods

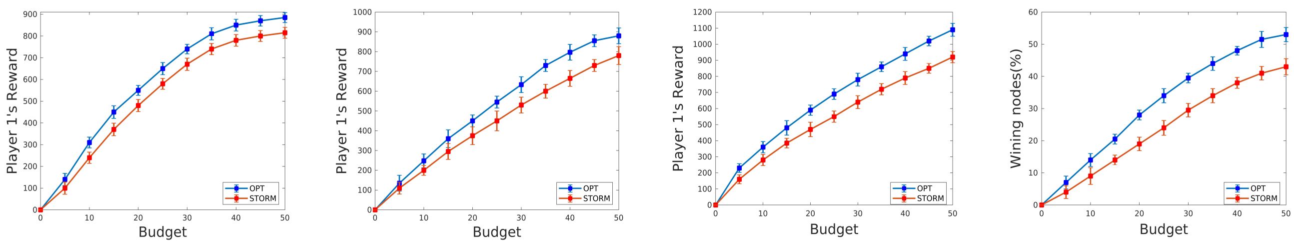
To find the performance of our approach, we consider different baseline IM methods and the state-of-the-art multi-round competitive approach, which is called *STORM* [26]. *OPT* is the name of the current paper's approach, which selects seed nodes based on both the fixed and dynamic weight of nodes. We consider the following approaches:

- STORM: a reinforcement learning-based algorithm that finds an optimal seed selection using Q-learning.



(a) Fixed player 1 budget 10 (b) Fixed player 2 budget 10 (c) Fixed player 1 budget 10 (d) Fixed player 2 budget 10

Fig. 6: Evaluation of player 1's reward with varying budget setting in synthetic dataset with different height.



(a) Facebook (b) Ca-HepTh (c) Cit-Hepph (d) Average all networks

Fig. 7: Evaluation of player 1's reward with varying budget setting in real datasets.

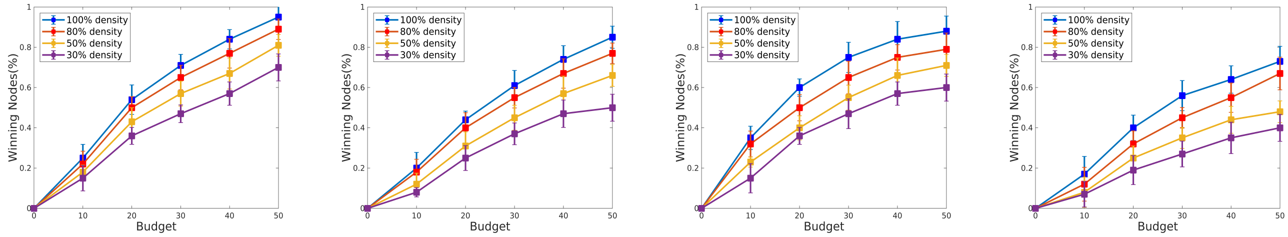
- Max-Degree: traditional influence maximization strategy as the algorithm selects nodes with the highest degree in the network as seed nodes.
- Centrality: this strategy select seed nodes based on the location of nodes in the network.
- Max-Weight: one of the baseline methods that finds the seed node based on the maximum summation of out-edge weights.
- Random: this strategy is a baseline algorithm that randomly chooses one of the seed selection methods.
- Voting: this method lets the other three strategies vote for a node as the seed node.
- MRIP: the algorithm selects seed nodes based on both the node's fixed and dynamic weight.
- OPT: our proposed learning-based approach

C. Experiment Results

We compare the influence spread of different algorithms on real-world datasets. Table IV shows the details of these real datasets, which are accessible from [34]. Each round is defined as players choosing a seed node and propagating influence. The number of active nodes after the diffusion process is used to evaluate the effectiveness of influence maximization algorithms. We consider the evaluation of our approach in the cases of different budgets, network structures, competing strategies, and ranges for the weight of the edges. Table V shows the evaluation of approaches in the case of different combinations of features. *OPT-F6* is the approach that we do not consider features 7 and 8. Similarly, others show the approaches with different features. It can be seen from the table V with the three features 6, 7, and 8 there is the best result in real datasets. We call our approach as *OPT*.

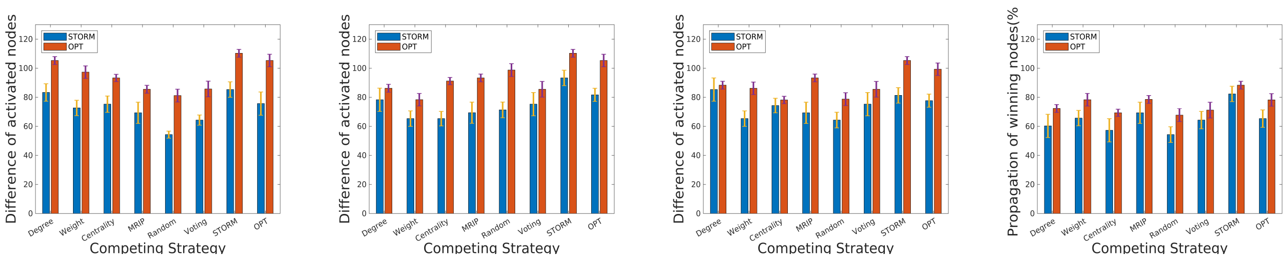
1) Evaluation on Budget Setting: In the first experiment, we examine the effectiveness of the proposed models' performances in terms of reward by assuming players have a different budget. We consider a fixed budget for one of the players, then analyze the result of competition with a varied amount of budget for the opponent side. Clearly, the larger the budget, the more the increase of spread. Parts (a) and (b) in Fig. 6 show the result of this experiment for the network with a topology that is like a tree. Parts (c) and (d) display the result in a network with a fat-tree topology. Figs. 6(a) and (c) illustrate the effect of varying budget for player 2 when player 1 has a fixed budget of \$10 for three algorithms of *Random*, *STORM*, and *OPT*. Moreover, Figs. 6(b) and (d) present the effect of varying budget for player 1 while player 2 has a fixed budget of \$10 on the spread of player 1's influence. It should be noted that we have trained the models by assuming both parties have the same budget. It can be seen from the figures that *OPT* achieves better performance in comparison with other models.

2) Evaluation Based on Different Topologies: We examine the effectiveness of the proposed models' performances on different networks in terms of reward by assuming players have different budgets. We consider a fixed budget for one of the players, then analyze the competition result with a varied budget for the opponent side. Clearly, the larger the budget, the more the spread increases. It should be noted that we have trained the models by assuming both parties have the same budget. It can be seen from the figures that *OPT* achieves better performance in comparison with other models. Also, we illustrate the performance of the proposed framework on networks with different structures. It can been seen from Fig. 7 that in different real datasets with different topologies, *OPT* has better results than *STORM*. In addition, *OPT* can find more rewards when a player has a higher amount of budget.



(a) Edge weights in (0.4.0.7) (b) Edge weights in (0.1,0.4) (c) Edge weight U(0,0.2) (d) Edge weight U(0,1)

Fig. 8: Evaluation of player 1's reward with different influence distributions in average all network.



(a) Facebook (b) Ca-HepTh (c) Cit-Hepph (d) Average all networks

Fig. 9: Evaluation of player 1' s reward with competing strategies in case of different budget.

3) Evaluation on Edge-weight Setting: We analyze the effect of different edge-weight settings on the proposed model. We consider the weight of the edges in the range of $[0.1, 0.4]$ and $[0.4, 0.7]$. In addition, the weight for edges are randomly sampled from the normal distribution of $U(0, 0.2)$ and $U(0, 1)$. In addition, we consider different densities for the network to evaluate the performance of the approach in the case of the sparsity of the network. From Fig. 8 can observe that the influence will diffuse more nodes when there are higher weights for edges. That happens because seed nodes can affect mode nodes. Also, the results show that *OPT* performs better if there is a high-density network.

4) Evaluation on Different Competing Strategies: We evaluate our approach for player 1 against a competitor with a different strategies such as *Degree*, *Weight*, *MRIP*, as well as the learned-based strategy *STORM*. In this part of experiment, we consider some baseline strategies such as *Degree*, *Weight*, *MRIP*, as well as the learned-based strategy *STORM*. For example, in the second to last column in Fig. 9, the blue one shows the result of the competition when players 1 and 2 have *STORM* approach, and the red one shows the result of the competition when player 1 uses *OPT* and player 2 uses *STORM* approaches. The blue one in the last column in Fig. 9 shows the results when player 1 uses *STORM* and 2 uses *OPT* approach. The red one shows the result when players 1 and 2 has *OPT* approaches. We can conclude from Fig. 9 that *OPT* has the best performance against all the competing strategies, even against the *STORM* which is the learned-based model. According to the result of this experiment, based on the network structure, there are different results with baseline competing strategies.

In summary, according to the results of experiments considering the new extra features to describe the state of the environment when there is a budget constraint for the players would be helpful to find better final rewards. In the case of different datasets with different topologies and different numbers of nodes and edges, MRIP helps players have maximum influence against the opponent's propagated influence. Considering the total amount of budget, density of dataset, weight distribution of edges, and competitions strategies as the different parameters of simulations illustrate this learning approach is helpful in different network structures, influence probabilities distributions, and different amounts of budget for players.

VI. Conclusion

In this work, we propose a reinforcement learning framework to tackle the multi-round CIM problem considering budget ratio for players. A large body of related research did not focus on the impact of different budgets for players in a CIM problem. We look into identifying the set of seed nodes to maximize the spread by considering opponents' capabilities. In fact, our framework considers the combination of seed-selection and budget-allocation strategies to invest the budget efficiently to achieve better rewards considering budget constraints. To summarize, our main contribution is designing and evaluating a budgeted learned-based framework that handles the multi-round CIM. Our experimental results show that our approach successfully increases the influence on the given network compared to some known baseline approaches and a learned-based CIM approach. One possible future research is to investigate whether it is possible to accelerate the process of learning and study which parameters have significant in fact in the speed of learning. We also plan to study the partial-observed MDP (POMDP) algorithms to handle the players' partial information about the environment, opponents' strategies, and diffusion process.

ACKNOWLEDGMENTS

This research was supported in part by NSF grants CNS 2128378, CNS 2107014, CNS 1824440, CNS 1828363, and CNS 1757533.

REFERENCES

[1] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 57–66.
[2] H. Li, J.-T. Cui, and J.-F. Ma, "Social influence study in online networks: A three-level review," *Journal of Computer Science and Technology*, vol. 30, no. 1, pp. 184–199, 2015.
[3] N. Ohsaka, "The solution distribution of influence maximization: a high-level experimental study on three algorithmic approaches," in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, 2020, pp. 2151–2166.
[4] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *In Proc. of ACM SIGKDD*, 2003, pp. 137–146.
[5] H. Nguyen and R. Zheng, "On budgeted influence maximization in social networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 6, pp. 1084–1094, 2013.
[6] K. Liontis and E. Pitoura, "Boosting nodes for improving the spread of influence," *arXiv preprint arXiv:1609.03478*, 2016.
[7] H. Zheng and J. Wu, "Friend recommendation in online social networks: Perspective of social influence maximization," in *In Proc. of ICCCN*. IEEE, 2017, pp. 1–9.
[8] X. Wang, X. Tong, H. Fan, C. Wang, J. Li, and X. Wang, "Multi-community influence maximization in device-to-device social networks," *Knowledge-Based Systems*, vol. 221, p. 106944, 2021.
[9] G. Tong, W. Wu, and D.-Z. Du, "Distributed rumor blocking with multiple positive cascades," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 2, pp. 468–480, 2018.
[10] X. Molinero and F. Riquelme, "Influence decision models: From cooperative game theory to social network analysis," *Computer Science Review*, vol. 39, p. 100343, 2021.
[11] S. Bharathi, D. Kempe, and M. Salek, "Competitive influence maximization in social networks," in *International workshop on web and internet economics*. Springer, 2007, pp. 306–311.
[12] H. Wu, W. Liu, K. Yue, W. Huang, and K. Yang, "Maximizing the spread of competitive influence in a social network oriented to viral marketing," in *In Proc. of Web-age information management*. Springer, 2015, pp. 516–519.
[13] A. M. Masucci and A. Silva, "Advertising competitions in social networks," in *In Proc. of ACC*. IEEE, 2017, pp. 4619–4624.
[14] H. Huang, Z. Meng, and H. Shen, "Competitive and complementary influence maximization in social network: A follower's perspective," *Knowledge-Based Systems*, vol. 213, p. 106600, 2021.
[15] S. Shirazipourazad, B. Bogard, H. Vachhani, A. Sen, and P. Horn, "Influence propagation in adversarial setting: how to defeat competition with least amount of investment," in *In Proc. of Information and knowledge management*, 2012, pp. 585–594.
[16] D. Kempe, J. Kleinberg, and E. Tardos, "Influential nodes in a diffusion model for social networks," in *International Colloquium on Automata, Languages, and Programming*, 2005, pp. 1127–1138.
[17] M. Granovetter, "Threshold models of collective behavior," *American journal of sociology*, vol. 83, no. 6, pp. 1420–1443, 1978.
[18] N. Niknami and J. Wu, "A budged framework to model a multi-round competitive influence maximization problem," in *In Proc. of IEEE International Conference on Communications (ICC)*, 2022.
[19] A. M. Masucci and A. Silva, "Strategic resource allocation for competitive influence in social networks," in *In Proc. of Communication, Control, and Computing (Allerton)*. IEEE, 2014, pp. 951–958.
[20] H. Zhang, A. D. Procaccia, and Y. Vorobeychik, "Dynamic influence maximization under increasing returns to scale," in *In Proc. of Autonomous Agents and Multiagent Systems*, 2015, pp. 949–957.
[21] L. Sun, W. Huang, P. S. Yu, and W. Chen, "Multi-round influence maximization," in *In Proc. of ACM SIGKDD*, 2018.
[22] H. Li, S. S. Bhowmick, J. Cui, Y. Gao, and J. Ma, "Getreal: Towards realistic selection of influence maximization strategies in competitive networks," in *In Proc. of ACM SIGMOD*, 2015, pp. 1525–1537.
[23] I. Rahaman and P. Hosein, "On the multi-stage influence maximization problem," in *In Proc. of LA-CCI*. IEEE, 2016, pp. 1–6.
[24] S. Zhu, I. Ng, and Z. Chen, "Causal discovery with reinforcement learning," *arXiv preprint arXiv:1906.04477*, 2019.
[25] Y. Wei, L. Zhang, R. Zhang, S. Si, H. Zhang, and L. Carin, "Reinforcement learning for flexibility design problems," *arXiv preprint arXiv:2101.00355*, 2021.
[26] S.-C. Lin, S.-D. Lin, and M.-S. Chen, "A learning-based framework to handle multi-round multi-party influence maximization on social networks," in *In Proc. of ACM SIGKDD*, 2015.
[27] T.-Y. Chung, K. Ali, and C.-Y. Wang, "Deep reinforcement learning-based approach to tackle competitive influence maximization," in *Proc. of MLG workshop*, 2019.
[28] K. Ali, C.-Y. Wang, M.-Y. Yeh, and Y.-S. Chen, "Addressing competitive influence maximization on unknown social network with deep reinforcement learning," in *In Proc. of ASONAM*. IEEE, 2020, pp. 196–203.
[29] K. Harshavardhan, V. Priyesh, W. Bryan, R. Balaraman, and T. Milind, "Influence maximization in unknown social networks: Learning policies for effective graph sampling," in *In Proc. of Autonomous Agents and MultiAgent Systems*, 2020.
[30] K. Ali, C.-Y. Wang, and Y.-S. Chen, "A novel nested q-learning method to tackle time-constrained competitive influence maximization," *IEEE Access*, vol. 7, pp. 6337–6352, 2018.
[31] T. Maehara, A. Yabe, and K. Kawarabayashi, "Budget allocation problem with multiple advertisers: A game theoretic view," in *In Proc. of Machine Learning*. PMLR, 2015, pp. 428–437.
[32] R. S. Sutton, A. G. Barto *et al.*, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 135.
[33] O. Morgenstern and J. Von Neumann, *Theory of games and economic behavior*. Princeton university press, 1953.
[34] [Online]. Available: http://snap.stanford.edu/data/

Nadia Niknami received her B.S. degree in Computer Science from University of Isfahan, Iran, in 2011, and MSc degrees From Tarbiat Modares University, Tehran, Iran in 2015. She is currently pursuing the Ph.D. degree in the Department of Computer and Information Sciences, Temple University, Philadelphia. Her current research focuses on security, privacy-preserving, attack-defense scenarios, Markov chain, social network, game theory, Software Defined Network.

Jie Wu is the Director of the Center for Networked Computing and Laura H. Carnell professor at Temple University. He also serves as the Director of International Affairs at College of Science and Technology. He served as Chair of Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and Associate Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Mobile Computing, IEEE Transactions on Service Computing, Journal of Parallel and Distributed Computing, and Journal of Computer Science and Technology. Dr. Wu was general cochair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, ACM MobiHoc 2014, ICPP 2016, and IEEE CNS 2016, as well as program chair/cochair for IEEE INFOCOM 2011, CCF CNCC 2013, and ICCCN 2021. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a Fellow of the AAAS and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.