A Budged Framework to Model a Multi-round Competitive Influence Maximization Problem

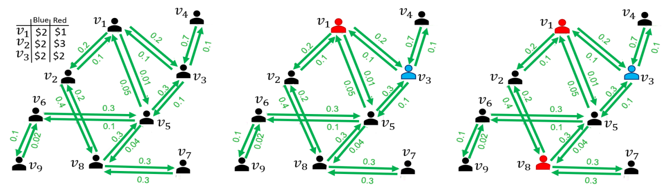
Nadia Niknami and Jie Wu
Department of Computer and Information Sciences
Temple University, Philadelphia, PA, USA

***Abstract*—Competitive Influence Maximization has attracted great research interest recently. A key challenge in this regard is modeling the users' information forwarding behaviors from nodes just activated to their not yet activated neighbors, especially when some competitors try to maximize their influence over the network. Competitors need to simultaneously decide how many resources should be allocated to a potential member of a social network. The main objectives are to discover which potential members to select and how many resources to allocate to these potential members to the maximize of the competitors' influence. Most previous works on competitive influence focus on the single-shot game without considering the effect of budget allocation. We are interested in multi-round competitive influence where each competitor needs to decide the location and amount of budget to invest simultaneously and repeatedly under a given total budget to maximize the total number of activated nodes. In this paper, we propose a tree-approximate game-theoretical framework and introduce the new measurement as a dynamic 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

In the real world, there are many competitors at the same time implementing their strategies to find a large influence on the same social network. Actually, each rational player tries to make its influence spread as maximal as possible and make that of its opponents as minimal as possible. That is why Competitive Influence Maximization (CIM) [1]–[3] has received a lot of attention recently. The CIM problem aims to select the best seeds in response to the other competitors' decisions with the goal of maximizing their influence. 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, $p(u, v) = 0$. Nodes can take on one of the following states: activated by Red, activated by Blue, and inactive. First, competitors identify the most influential nodes of the network. Then they compete over only these influential nodes by the amount of budget they allocate 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) [5] models are NP-hard.



(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.

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 $p(u, v)$. Consider the social network in Fig. 1(a). Players Red and Blue compete over the nodes of this network. They consider v_1, v_3 , and v_8 as seed nodes. The table shows the amount of budget allocation from these players over seed nodes. Player Red wins v_1 with the probability of $2/(1 + 2) = 2/3$. Player Blue wins v_3 with the probability of $3/(3 + 2) = 3/5$. Players have the same budget allocation on v_8. The winning probability is proportional to the budget allocation of two parties.

We model such a scenario by the multi-round CIM where each player would prefer to find an optimal combination of strategies to utilize their budget efficiently in a competitive environment. Instead of selecting seeds only in the first round, we consider a more realistic and practical setup in which the players might keep taking action by selecting seed nodes, based on 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. The most influential nodes are selected according to different strategies based on the current state of the given network. In each round, players select a seed node s_t, make a decision about 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. As in-

This research was supported in part by NSF grants CNS 1629746 and CNS 1651947.

fluence maximization is NP-hard, we introduce a new notation of Most Reliable Influence Path (MRIP) as an approximation.

The contributions of this paper are summarized as follows: 1) We define new measurement called dynamic weight for nodes. Considering both fixed and dynamic weights in the processes of selecting seed nodes helps players to have a more accurate selection. 2) 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 critical path in the scheduling community. 3) 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. 4) We propose a CIM model which selects the winner of the node in case of breaking tie based on the budget proportion.

II. Related work

Competitive Influence Maximization. The competitive IM aims to find a strategy for the competitors in a social network such that one's influence is maximized while his opponents' influences are minimized [6]. Instead of focusing on the propagation of a single idea in social networks, there are various extensions of the IC model and the LT model for multiple competing ideas [7]. Li *et al*. [8] consider a model for competitive IM. Given a graph $\mathcal{G}$ and a diffusion model, the strategy space consists of all IM algorithms that may be adopted by players. The objective is to find a Nash equilibrium strategy for each player such that his own influence $\pi(s_i)$ is maximized. In [9], 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.

Reinforcement Learning. An important line of work that uses RL to solve NP-hard optimization problems on graphs is [10] [11]. In [12], Lin *et al*. 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 [13] 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*. [14] propose a deep reinforcement learning-based model to tackle the CIM on unknown social networks. K. Ali *et al*. in [15] 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.

Resource Allocation Against Opponents. Parties in a CIM 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 simultaneously allocate limited resources over several objects. Authors in [16] address the budget allocation scenario in maximization influence problem. Companies can allocate different values of their budget to a node in the network, and nodes will prefer the product of the company that has offered a higher value. Here, companies compete with each other through the amount of budget they allocate to each node in the network.

TABLE I: Main notations

Symbol	Meaning
B_1/B_2	Total budget of player $1/2$
T	Total number of rounds
$N(u)$	Neighbor set of node u
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

Different from most of the existing works, in this paper, we study the problem of Multi-round CIM 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.

III. Preliminary

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. In $\mathcal{G}$, a node u can be activated if it accepts the idea of the player i. Once a node u accepts the idea of a player i, it cannot switch its occupation status to other parties. If the given node does not accept any idea, it means that the state of the node u is inactive.

Competitive Influence Maximization. In a multi-stage CIM problem, competitors need to make decisions about selecting seed nodes in each of sequence stages simultaneously. 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 budget of size B_1 and player 2 has 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 player who can obtain the most reward by influencing the more important nodes would be the winner of this game. 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 the competition of players on seed nodes by 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 influence try to activate the given node by 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 to this node by influence of link.

1) Node-Node Influence Competition: Considering a Node-Node competition on the node u. Suppose that x_1 and x_2 are the amount of budget that players 1 and 2 have allocated

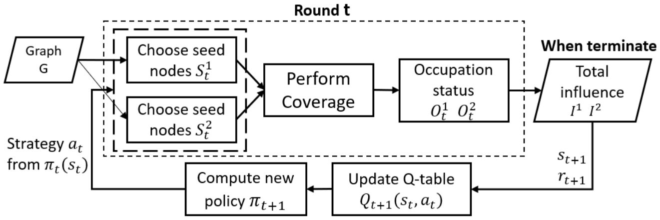


Fig. 2: Approach during training process.

to node u. The winning probability of player 1 for this competition is $x_1/(x_1+x_2)$.

2) Link-Link Influence Competition: Link-Link influence competition will be happened after the process of budget allocation and determining the winner of this stage in the case of taking the given seed node. During the process of propagation, 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 $(p_1/(p_1+p_2)) \times (1 - p_1p_2)$, 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 $(p_2/(p_1+p_2)) \times (1 - p_1p_2)$.

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

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 what is the strategy of the other side. 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 the learning of optimal decisions over time. In RL, the agent keeps interacting with the environment to find the optimal policy π in order to maximize his expected accumulated rewards [17]. The goal of an RL is to learn a policy $\pi(s)$ to determine which action to take given a certain 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.

RL formulates the expected accumulated rewards of a state and the expected accumulated rewards given a state-action pair to estimate how good the policy π is in 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:

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

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:

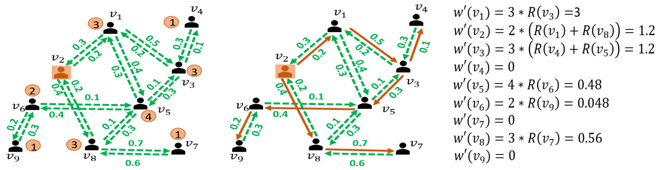
$$Q^\pi(s, a) = E_\pi\{\sum\nolimits_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\} \tag{2}$$

The state value and action value can be learned through the interacting of agents with the environment. The optimal policy $\pi(s)$ can be obtained given the Q function and find the maximum value. Fig. 2 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 and compute new policy against the opponent's strategy and select a new seed set. We assume that there are only two parties that compete with each other. We need to first define the environment, the reward, the action, and the state.

IV. Methodology

In our approach, the two phases of seed selection and budget allocation are integrated into the RL model. Convincing influential nodes to act as a seed in addition to just selecting seed nodes is what we consider in the budget allocation phase. The goal of the agent in the proposed framework is to learn the optimal policy π for a seed placement strategy that maximizes its expected accumulated rewards. The players first identify the influential nodes of the network. Then, they compete over only the selected nodes, rather than the whole network, by the value of the budget they allocate to each influential node. At each round t, the agent observes a set of features that represents the network state $s_t \in S$ and selects an action a_t from the set of legal actions. At each round, the agent first selects a seed set, $S_t \subset V$, based on its past observations. Note that S_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. The algorithm terminates when no budget remains, or no node can be added to seed set S.

Selecting Seed Nodes and Propagation Model. Consider a static social network and fixed budget at each round for seed selection, that is, a single seed selection at each round. The goal of each player is to reach and activate as many users as possible within the budget. Each player can decide to implement the specific strategy in order to maximize its overall influence in $\mathcal{G}$. A strategy denotes how a player spends its budget at each round to select the seed nodes. The main idea of the maximum influence with a spanning tree is to restrict the influence diffusion of node u to a local tree structure rooted at u. The influence of a node in the tree can be computed efficiently and exactly. Note that the conflict rule is slightly different from other works. In contrast to other approaches, which consider a priority for one of the players or select the winner of conflicting randomly, our approach allows players to



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

Fig. 3: 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** $0.4 * 0.1 = 0.04$	$R(v_1)$

increase their investment in case of tie-breaking. The winner will be determined with help of budget proportion.

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. 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 in the case of considering the most reliable paths originate from the given node can be considered as a new measurement for ranking nodes.

In this paper, we call 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 node. In fact, Prim's algorithm helps 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.. 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 root node of the tree. 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 source of them in the explored node set A, and its destination is in $V - A$. We continue this process until A

Algorithm 1 Finding seed set by MRIP

$S \leftarrow \varnothing$
for all $u \in V$ **do**
  $w'(u) \leftarrow 0$
for $u \in V$ **do**
  Construct $\mathcal{T}_u$ via Alg. 2
  for each leaf v in reverse $\mathcal{T}_u$ **do**
    $z \leftarrow parent(v)$
    while $v \neq u$ **do**
      Compute $w'(z) = w'(z) + R(v) \times w(z)$
      $v \leftarrow z$
      $z \leftarrow parent(v)$
new seed $\leftarrow \arg\max_{u \in V/\ S} w'(u)$
$S \leftarrow S \cup \{new\ seed\}$
$V_A \leftarrow$ Activated nodes by *new seed* node
Constructing $\mathcal{G}'$ with vertex set $V - V_A$
Recalculate $\mathcal{T}$ and w' in $\mathcal{G}'$

Algorithm 2 Computing $\mathcal{T}_u$

Require: $\mathcal{G}(V, E, P)$, source node u
$A = \{u\}$, $R(u) = 1$
while $A \neq V$ **do**
  Find node $v \in N(A)$ and $v \in V - A$ such that
  $R'(v) = \max_{(s,v): s \in A, v \in V-A} R(s) \times p(s, v)$
  $R(v) = R'(v)$
  $A = A \cup \{v\}$
  Set s as the parent of v in spanning tree $\mathcal{T}_u$
return $\mathcal{T}_u$

includes all of the nodes in V. Algorithm 2 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:

$$w'(u) = \sum\nolimits_{\forall v \in V} R(v) \times w(u), \quad (3)$$

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. 3, 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. 3 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. 3(b), the influence weights of all nodes are shown in Fig.3(c). The intuition behind the proposed algorithm comes from Dijisktra's algorithm. We can prove the proposed algorithm can find the most reliable path correctly.

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

TABLE III: 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

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.

This theorem can be proved by the idea behind Dijkstra's algorithm easily. Algorithm 1 presents the processes of selecting the seed node base on the influence weight of nodes. After finding $\mathcal{T}_v$ for each node v in algorithm 2, 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 selecting 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*.

Reinforcement Learning Settings. To use reinforcement learning, we need to define some important parameters. We treat the influence propagation process as the environment effect, which propagates the influence of activated nodes to its neighbors and activates new ones. The reward we get after T steps is the number of nodes influenced in the entire graph. Players can allocate different values for their budget to a node in $\mathcal{G}$. Players compete with each other by the amount of budget they allocate 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 [12] [15] 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.

We need to model the state to represent the network and environment status. We need to resort to the design of features to represent the current occupation status as well as the condition of the network. 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 and 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. 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. 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.

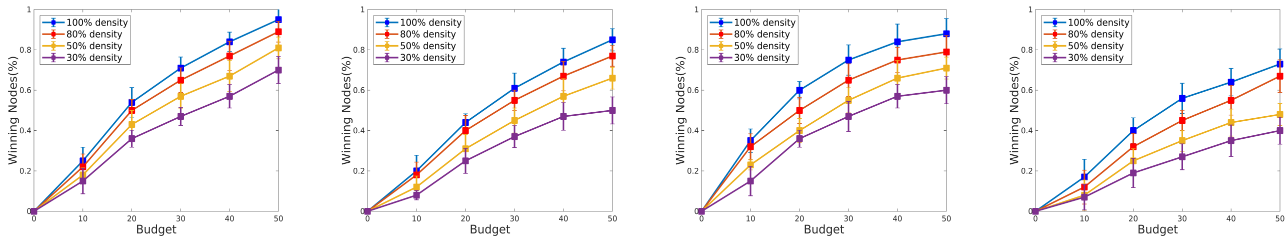
V. Experiments

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. The details of these real datasets are accessible from [18]. We used the igraph Python library to represent the graphs and for the shortest path calculations. In order 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* [12]. *OPT* is the name of the current paper's approach which selects seed nodes based on the both fixed and dynamic weight of nodes. Each round is defined as players choose a seed node and influence being propagated. The number of active nodes after the diffusion process is used to evaluate the effectiveness of influence maximization algorithms. We consider evaluation of our approach in the cases of different budgets, network structures, competing strategies, and ranges for the weight of the edges. Table III shows the evaluation of approaches in the case of different combinations of features.

1) Evaluation on Edge-weight Setting: We analyze the effect of different edge-weight settings on the proposed model. In addition, we consider different densities for network to evaluate the performance of the approach in the case of the sparsity of network. It can be observed from Fig. 4 that the influence will diffuse more nodes when there are higher weights for edges. That happens because seed nodes can have an effect on mode nodes. Also, the results show that *OPT* performs better if there is a high-density network.

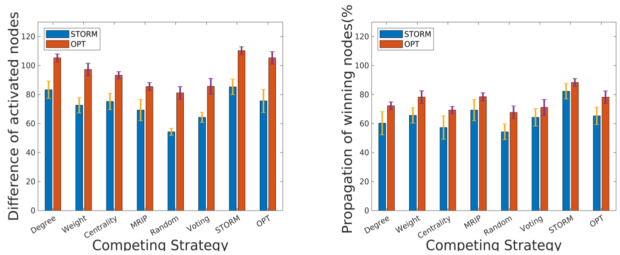
2) Evaluation on Different Competing Strategies: We evaluate our approach for player 1 against a competitor with a different strategies such as *Degree*, *Weight*, *MRIP*, as well as the learned-based strategy *STORM*. In this part, we consider a player who has a *STORM* approach or *OPT* against an opponent behaving in one of the competing strategies. We can conclude from Fig. 5 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 structure of network there are different results with baseline competing strategies.

3) Evaluation Based on Different Budgets: We examine the effectiveness of the proposed models' performances in terms of reward by assuming players have different budgets. 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



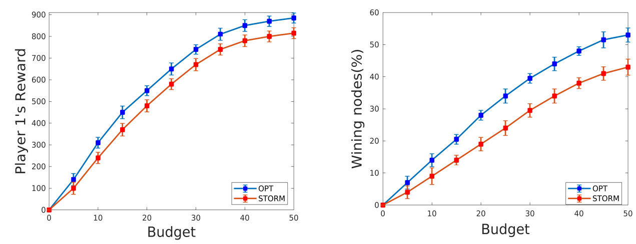
(a) Edge weights in (0.4.0.7) (b) Edge weights in (0.1,0.4) (c) Edge weight U(0,0.2) (d) Edge weight U(0,1)

Fig. 4: Evaluation of player 1's reward with different influence distributions in average all network.



(a) Facebook (b) Average all networks

Fig. 5: Player 1's reward against different competing strategies.



(a) Facebook (b) Average all networks

Fig. 6: Player 1's reward with varying budget settings.

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. 6 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.

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 the capabilities of opponents. 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 the design and evaluation of a budgeted learned-based framework that handles the multi-round CIM. Our experimental results show that our approach is successful in increasing the influence on the given network in comparison with some known baseline approaches as well as a learned-based CIM approach.

References

[1] 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.

[2] A. M. Masucci and A. Silva, "Advertising competitions in social networks," in *In Proc. of ACC*. IEEE, 2017, pp. 4619–4624.

[3] 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.

[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] 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.

[6] 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.

[7] L. Sun, W. Huang, P. S. Yu, and W. Chen, "Multi-round influence maximization," in *In Proc. of ACM SIGKDD*, 2018.

[8] 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.

[9] I. Rahaman and P. Hosein, "On the multi-stage influence maximization problem," in *In Proc. of LA-CCI*. IEEE, 2016, pp. 1–6.

[10] S. Zhu, I. Ng, and Z. Chen, "Causal discovery with reinforcement learning," *arXiv preprint arXiv:1906.04477*, 2019.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] R. S. Sutton, A. G. Barto *et al.*, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 135.

[18] [Online]. Available: http://snap.stanford.edu/data/