Exploiting an Adversary’s Intentions in Graphical Coordination Games

Brandon C. Collins and Philip N. Brown

Abstract—How does information regarding an adversary’s intentions affect optimal system design? This paper addresses this question in the context of graphical coordination games where an adversary can indirectly influence the behavior of agents by modifying their payoffs. We study a situation in which a system operator must select a graph topology in anticipation of the action of an unknown adversary. The designer can limit her worst-case losses by playing a security strategy, effectively planning for an adversary which intends maximum harm. However, fine-grained information regarding the adversary’s intention may help the system operator to fine-tune the defenses and obtain better system performance. In a simple model of adversarial behavior, this paper asks how much a system operator can gain by fine-tuning a defense for known adversarial intent. We find that if the adversary is weak, a security strategy is approximately optimal for any adversary type; however, for moderately-strong adversaries, security strategies are far from optimal.

I. INTRODUCTION

Many engineered and social systems consist of a collection of agents making decisions based on locally available information. For example, a group of unmanned vehicles performing surveillance in a hostile area may use a distributed control strategy due to communication constraints; social systems are intrinsically distributed, as individuals make decisions based on personal objectives and in response to the behavior of friends and acquaintances. For example, the decision to adopt a recently released technology may depend both on the quality of the item itself and on friends’ choices [1].

In a multiagent system, whether engineered or social, an adversary may be able to influence the behavior of individual agents to indirectly influence the overall behavior of the system. In principle, a defender can compute and deploy a security strategy, which effectively plans for a maximally-malicious adversary. Security strategies allow the defender to guarantee a minimum level of system performance, but as worst-case defenses, they may be suboptimal against a real adversary. In this paper, we investigate the cost of employing security strategies against poorly-characterized adversaries, in particular studying how information regarding an adversary’s intent and strength can be used to fine-tune defensive measures.

A popular theoretical framework in this area is to model the agents in a multiagent system as players in a game, endow the players with a distributed learning algorithm (such as the well-studied log-linear learning rule [2], [3]), and analyze the emergent behavior resulting when the agents execute their algorithm. This has prompted much recent interest the question: how can adversarial manipulation alter the emergent behavior of distributed decisionmaking [4]–[7]?

Work in this area has characterized how an adversary’s capabilities and intelligence impacted their ability to influence behavior [5], [6], as well as investigated simple questions of defense [7].

In this paper, we ask how a system operator can leverage information about an adversary’s goals (or intent) to select an effective defensive posture. Here, we suppose that the system operator must select a defensive posture without knowing which type of adversary she will face. That is, a central question of this paper is this: how should a system operator implement effective defenses if the goals of the adversary are unknown? Naturally, one option is that the planner could adopt a pessimistic view and plan for the worst-case adversary type; however, we demonstrate that doing so can be quite costly in some circumstances.

We study two models of adversary intent:

- A malicious adversary wishes to harm the system operator as much as possible, and
- An advertiser adversary has a vested interest in some specific agent action.

We investigate the system operator’s defensive decision problem as affected by the adversary’s strength (measured by how many friendly agents the adversary is able to influence) and the adversary’s goals. We find that there are distinct regimes of adversary strength: when the adversary is “weak” (only able to influence a small set of agents), a security strategy is nearly optimal. On the other hand, if the adversary is moderately powerful, then security strategies are always far from optimal, and can only garner a maximum of a fraction of the total available welfare for an advertiser.

II. MODEL

A. Model of agent and adversarial behavior

To study adversarial influence we use the n-player graphical coordination game [8]–[10]. The foundation of a graphical coordination game is a simple two agent coordination game [11], [12], where each agent must choose one of two conventions, \{x, y\}, with payoffs depicted by the following payoff matrix:

|   | \(x\)  | \(y\)  |
|---|---|---|
| \(x\) | \(1 + \alpha\), \(1 + \alpha\) | 0, 0 |
| \(y\) | 0, 0 | 1, 1 |

Funding support provided by Colorado State Bill 18-086, the UCCS Undergraduate Research Academy, and the UCCS Committee on Research and Creative Works.

Brandon C. Collins (corresponding author) and Philip N. Brown are with the Department of Computer Science at the University of Colorado Colorado Springs, {bccollin3,philip.brown}@uccs.edu.
The game is played on undirected graph \( g = (N,E) \) between agents \( N = \{1, 2, 3, ..., n\} \) using edge set \( E \). Agent \( i \) plays the two-player coordination game with agent \( j \) if \((i,j) \in E \) and \( i \neq j \). We term \( \alpha \in (0,1) \) as the payoff gain that ensures the \( x \) convention has an intrinsic coordination benefit. An agent’s total payoff is the sum of payoffs it receives in the two-player games played with its neighbors \( N_i = \{j \in N : (i,j) \in E\} \), i.e., for a joint action \( a = (a_1, ..., a_n) \in A \) the utility function of agent \( i \) is

\[
U_i(a_1, ..., a_n) = \sum_{j \in N_i} u(a_i, a_j) - \frac{1}{p} \text{ if } a_i = x, 0 \text{ if } a_i = y
\]

where \( u(\cdot) \) is chosen according to payoff matrix \( [\mathbb{1}] \). Here, the parameter \( p \in (0, \alpha) \) indicates that each agent experiences some small personal cost for selecting the \( y \) convention. Joint actions \( \vec{x} := (x, x, ..., x) \) and \( \vec{y} := (y, y, ..., y) \), where either all players choose \( x \) or all players choose \( y \), are Nash equilibria of the game for any graph; other equilibria may also exist depending on the structure of graph \( g \).

Suppose agents in \( N \) interact according to the graphical coordination game above, specified by the tuple \( (g, \alpha, p) \). The system operator’s objective is that the agents maximize the system welfare, or the sum of nominal agent utilities:

\[
W(a) = \sum_{i \in N} U_i(a).
\]

An adversary seeks to influence the system’s emergent behavior by modifying agent utility functions; the adversary accomplishes this by selecting a set of agents and posing as a neighbor whose action is fixed at \( \vec{y} \). Effectively, the adversary spreads “impostor” agents throughout the network, attaching each impostor to a specific friendly agent. We write \( K = \{i_j\}_{j=1}^{k} \) to represent the adversary’s selection, where \( i_j \) denotes the index of the friendly agent connected to adversary vertex \( j \); we write \( K \) to denote the set of all feasible adversary selections. That is, each impostor vertex connects to one and only one friendly agent, but note that we allow multiple impostor vertices to connect to a single friendly agent. Agents’ utilities, \( \vec{U} : A \times K \to \mathbb{R} \), are now a function of adversarial and friendly behavior, defined by:

\[
\vec{U}_i(a, K) = U_i(a) + \{ \{ j : i_j \in K \} \} \mathbb{1}_{a_i=y}.
\]

That is, in addition to the ordinary payoff from coordinating with its neighbors, if an agent selects \( y \), it receives an additional payoff of 1 for each impostor node that it is connected to.

Note that the payoffs in this game define an exact potential game [13]. Given action profile \( a \), let \( E_x(a) (E_y(a)) \) be the set of edges connecting \( x \)-playing (\( y \)-playing) agents, and let \( N_y(a) \) be the set of \( y \)-playing agents, and let \( N_{Ky}(a) \) be the number of total adversarial attacks that are successfully converting an agent to \( y \) in \( a \). Then it can be shown that the game’s potential function is given by

\[
\Phi(a) = (1+\alpha)|E_x(a)| + |E_y(a)| - \frac{1}{p}N_y(a) + |N_{Ky}(a)|.
\]

### B. Learning and Emergent Behavior

Now, suppose that agents in \( N \) update their actions over time according to some stochastic process such as the well-studied log-linear learning algorithm. Log-linear learning is known to select a potential function maximizing state with high probability in the long run; it is said that the potential function maximizers of the game are the stochastically stable states of log-linear learning. For reasons of space, we omit the full details, which can be found in [5], [6], [14].

Accordingly, as we assume the agents are employing log-linear learning and thus selecting potential-maximizing states, Definition [\ref{def:emergent}] gives our central solution concept:

Definition 1: Given adversarial influence \( K, \alpha \in (0,1) \), and \( p \in (0,\alpha) \), action profile \( a \in A \) is called an emergent state with respect to \( K \), written \( a^* (g, \alpha, p, K) \), if for each component of graph \( g \), all agents in that component are selecting the same action \( \vec{y} \) (either \( x \) or \( y \)) and that action maximizes the component’s potential function.

Suppose component \( c \) with \( m_c \) edges and \( n_c \) nodes is being attacked by \( k_c \) impostor nodes. Writing \( a_c \) to denote the action profile restricted to \( c \), we can write the potential function of this component as

\[
\Phi_c(a_c, k_c) = \begin{cases} m_c + m_c\alpha & \text{if } a_c = \vec{x} \\ m_c - n_c p + k_c & \text{if } a_c = \vec{y} \end{cases}
\]

Let \( k^*(c) \) be defined as the minimum \( k_c \) required such that the emergent state of component \( c \) is weakly \( \vec{y} \).

\[
k^*(c) = \min\{ k_c \mid \Phi_c(\vec{y}, k_c) \geq \Phi_c(\vec{x}, k_c) \}.
\]

It can be shown that \( k^*(c) = [m_c\alpha + n_c p] \).

In this paper, we consider two distinct types of adversary, which we term advertiser \( A(k) \) and malicious \( M(k) \). An advertiser adversary can be thought of as a promoter for the \( y \) action (perhaps a marketing manager for a competing product or a political rival), and arranges its attacks in an attempt to maximize the number of agents selecting \( y \) in an emergent state. That is, the advertiser desires to maximize the objective function

\[
U^A(a^*, K; g) = |\{i : a^*_i = y\}|
\]

A malicious adversary desires to maximize the objective function

\[
U^M(a^*, K; g) = -\sum_{i \in N} U_i(a^*)
\]
equivalently minimizing the system’s total welfare.

In this paper, we study a situation in which the system operator selects a network topology \( g \) in anticipation of an adversary attack. Thus, we nominally have a Stackelberg game in which the operator acts as leader and the adversary acts as follower. For any network topology \( g \), we assume that the adversary plays a best response to \( g \); for adversary type \( T \in \{A(k), M(k)\} \), we write

\[
K_T(g) = \arg \max_{K \in \mathcal{K}} U_T(a^g, K; g)
\]

(11)
to denote the adversary’s best response to \( g \) at an emergent state (breaking ties in favor of the adversary). For convenience, we write

\[
W(g, T, \alpha, p) \triangleq W(a^g(g, \alpha, p, K_T(g))),
\]

(12)
or simply \( W(g, T) \) when the dependence on \( \alpha, p \) is clear. The system operator’s nominal goal is to select a graph topology \( g \) which maximizes \( \mathcal{W} \). For adversary type \( T \in \{A, M\} \), we write

\[
g^*(T) \triangleq \arg \max_g W(g, T, \alpha, p).
\]

(13)

For simplicity, we assume throughout that the system contains exactly twice as many agents as graph edges; writing \( m := |E| \), we have that \( 2m = n \). We note that \( n > 2m \) leaves a quantity of nodes that a system designer can never interact with. Conversely, for \( 2m < n \) the possible edge topologies are a strict subset of the case when \( m = 2n \). We often utilize several specific graph topologies, defined as:

- **SPARSE**: A graph with exactly \( m \) 2-node components and no isolated nodes.
- **LINE**: A graph with a single sparse giant component described by a line graph containing \( m + 1 \) nodes; this graph contains \( m - 1 \) isolated nodes.
- **COMPLETE**: A graph with a single dense giant component described graph containing all edges and the minimum-possible number of nodes.

In this paper, we study decision making in the face of strategic uncertainty; accordingly, we ask the following: if the system operator is uncertain about the adversary’s goal (i.e., type), how should she select graph topology? To begin to address this question in the context of network design for graphical coordination games, in this paper we study the cost of mischaracterizing the adversary type for a variety of situations. That is, for \( T, T' \in \{A(k), M(k)\} \), we define the relative regret associated with type \( T \) as

\[
R(T) = \frac{W(g^*(T'), T', \alpha, p) - W(g^*(T), T', \alpha, p)}{W(g^*(T), T', \alpha, p)},
\]

(14)

which captures the fraction of welfare that is lost by planning for type \( T \) when the actual realized adversary type is \( T' \).

### III. OUR CONTRIBUTIONS

We begin with a result which expresses a threshold on \( k \) below which the operator can guarantee a completely-coordinated graph. For adversary type \( T \in \{M, A\} \), we define \( k_{\max}(T(k)) \) as the largest value of \( k \) below which the operator has a graph \( g \) such that \( a^g(g, \alpha, p, K_T(g)) \) is fully coordinated on \( x \) actions. That is, if \( k < k_{\max}(T) \), then the operator can be sure that a strategic adversary of type \( T \) will not cause any coordination on \( y \) in any emergent state, even if isolated nodes are playing \( y \). Note that the operator always prefers to avoid \( y \) coordination if possible (see Lemma 3).

In the following, \( d \) is defined to be the minimum integer that satisfies \( m \leq \binom{d}{2} \). We introduce the following technical definition:

**Definition 2**: A triple \((m, \alpha, p)\) is said to be ordinary if there exists some graph \( g = (N, E) \) such that every component \( c \) in \( g \) satisfies \( n_c < k^*(c) \).

**Theorem 1**: We find \( k_{\max}(\cdot) \) for malicious adversary \( M \) and advertiser adversary \( A \).

\[
k_{\max}(M) = \lceil m\alpha + mp + p \rceil,
\]

\[
k_{\max}(A) = \begin{cases} 
2m - 2d + \lceil m\alpha + dp \rceil & \text{if } (m, \alpha, p) \text{ is ordinary} \\
k_{\max}(M) & \text{otherwise.}
\end{cases}
\]

The proof of Theorem 1 appears in the Appendix. The first result of the Theorem comes directly from Lemma 3.

We begin deriving the second result by seeking an upper bound on \( R(M) \) for \( k < k_{\max}(M) \). We present lower bound on \( W(g^*(M(k)), A(k)) \) as

\[
W(g^*(M(k)), A(k)) \geq W(g^*(M(k)), M(k)) \geq W(LINE, M(k)) = W(LINE, A(k)).
\]

Note the last equality holds as \( k_{\max}(M) \) was derived in Theorem 1 via a LINE graph, thus leaving any adversary with trivial choices while \( k < k_{\max}(M) \). This allows Lemma 2 to directly apply to the numerator of \( R(M) \) producing upper bound

\[
R(M) \leq \frac{\mathcal{O}\left(\sqrt{m}\right)}{W(LINE, M(k))}.
\]

When \( k < k_{\max}(M) \), it can be observed that \( W(LINE, M(k)) = 2m + 2m\alpha - p\min\{k, m - 1\} = \Omega(m) \).

Thus

\[
R(M) = \Theta\left(\frac{1}{\sqrt{m}}\right)
\]

(16)
as desired.

**Theorem 3**: When \( k \in [k_{\max}(M), k_{\max}(A)] \), we have that the maximum achievable welfare in the presence of an advertiser is significantly higher than that available in the
Fig. 1. We demonstrate that \( k_{\text{max}}(M) = 117 \) and \( k_{\text{max}}(A) = 183 \), realized by LINE and COMPLETE respectively, serve as important breakpoints in optimal decision making. In this case \( \alpha + 2p > 2 \), so SPARSE dominates for either adversary after their respective \( k_{\text{max}} \) value.

![Graph showing welfare vs k for different algorithms](image)

### B. Malicious Characterization

We once again are able to readily play LINE if \( k < k_{\text{max}}(M) \) by similar arguments. However once \( k \) exceeds \( k_{\text{max}}(M) \) we must chose between potentially all 3 graphs.

By similar arguments COMPLETE will at least weakly dominate LINE for all \( k \geq k_{\text{max}}(M) \). Thus we will define \( k_{\text{int}} \) as the intersection between SPARSE and COMPLETE and use it as decision point between the two.

For our first case where \( \alpha + 2p > 2 \) we find

\[
g^*(A(M)) = \begin{cases} 
\text{LINE} & k < k_{\text{max}}(M) \\
\text{SPARSE} & k_{\text{max}}(M) \geq k.
\end{cases}
\]

If \( \alpha + 2p \leq 2 \) we simply use \( k_{\text{int}} \) as a break point where SPARSE switches to COMPLETE.

\[
g^*(A(M)) = \begin{cases} 
\text{LINE} & k < k_{\text{max}}(M) \\
\text{SPARSE} & k_{\text{max}}(M) \leq k < k_{\text{int}} \\
\text{COMPLETE} & k_{\text{int}} \leq k.
\end{cases}
\]

We showcase the results with sample parameters for \( \alpha + 2p > 2 \) in Figure 1 and \( \alpha + 2p \leq 2 \) in Figure 2.

### References

[1] G. E. Kreindler and H. P. Young, “Rapid innovation diffusion in social networks,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 111, pp. 10881–10888, 2014.

[2] L. E. Blume, “The Statistical Mechanics of Strategic Interaction,” *Games and Economic Behavior*, vol. 5, no. 3, pp. 387–424, 1993.

[3] D. Shah and J. Shin, “Dynamics in Congestion Games,” *Proc. ACM SIGMETRICS’10*, vol. 38, no. 1, pp. 107–118, 2010.

[4] H. P. Borowski and J. R. Marden, “Understanding the Influence of Adversaries in Distributed Systems,” in *IEEE Conference on Decision and Control (CDC)*, pp. 2301–2306, 2015.

[5] P. N. Brown, H. P. Borowski, and J. R. Marden, “Security Against Impersonation Attacks in Distributed Systems,” *IEEE Transactions on Control of Network Systems*, vol. 6, no. 1, pp. 440–450, 2019.

[6] B. Canty, P. N. Brown, M. Alizadeh, and J. R. Marden, “The impact of informed adversarial behavior in graphical coordination games,” in *57th IEEE Conference on Decision and Control*, pp. 1923–1928, 2018.

[7] K. Paarporn, M. Alizadeh, and J. R. Marden, “Risk and Security Tradeoffs in Graphical Coordination Games,” in *58th IEEE Conference on Decision and Control*, pp. 4409–4414, 2019.

[8] M. Kearns, M. L. Littman, and S. Singh, “Graphical Models for Game Theory,” *Proceedings of the 17th conference on Uncertainty in artificial intelligence*, no. Owen, pp. 253–260, 2001.

[9] H. P. Young, *Strategic Learning and its Limits*, vol. 9780199269. 2011.
A. Notation

For component $c$ in some graph we define $\Delta W(c) := 2m_c + n_c \pi$ as the loss in welfare when component $c$ switches from being stable on $x$ to $y$.

We define several sets

- $C(g) = \{c_1, c_2, c_3, \ldots, c_n\}$ such that $c_i \in C(g)$ is a component in graph $g$
- $C_e(g) = \{c_1, c_2, c_3, \ldots, c_n\}$ such that $c_i \in C_e(g)$ is an edged component in graph $g$
- $C_l(g) = \{c_1, c_2, c_3, \ldots, c_n\}$ such that $c_i \in C_l(g)$ is a node with degree 0
- $C_x(g|T(k)) = \{c_1, c_2, c_3, \ldots, c_n\}$ such that $c_i \in C_x(g|T(k))$ is a component with emergent state $\vec{x}$
- $C_y(g|T(k)) = \{c_1, c_2, c_3, \ldots, c_n\}$ such that $c_i \in C_y(g|T(k))$ is a component with emergent state $\vec{y}$

We use

$$W(g, T(K)) = 2m + 2ma - \sum_{c \in C_v(g, A(k))} \Delta W(c) \quad (22)$$

as a convenient way to denote system welfare given topology and the adversary.

We term some component $c$ advertiser resistant if $c$ satisfies $n_c < k^*(c)$.

B. Proof of Theorem 7

First we derive $k_{\text{max}}(M)$. Let $g$ be an arbitrary graph and $q \in \arg \min_{c \in C_e(g)} k^*(c)$. Lemma 3 gives the malicious adversary strictly prefers $C_y(g|M(k)) \subseteq C_e(g)$. Therefore we can simply seek to design $g$ to maximize $k^*(c)$. First,

$$k_{\text{max}}(M) = \max_{n_q, n_p} \left[ m_q \alpha + n_q \pi \right] = \left[ m \alpha + (m + 1)p \right] \quad (23)$$

simply by problem parameters. Now we seek $k_{\text{max}}(A)$. We begin with the case where in the triple $(m, \alpha, p)$ is not ordinary. This implies all graphs contain a nonempty set $S$ of non advertiser resistant components and we take $c = \arg \min_{s \in S} k^*(s)$. Thus at $k = k^*(c)$, $n_c \geq k$ and intuitively the advertiser may weakly improve $U^a$ by taking component $c$ over lone nodes. This behavior is similar to the malicious adversary and we conclude $k_{\text{max}}(A) = k_{\text{max}}(M)$ in this case.

For brevity we present an outline of the case where $(m, \alpha, p)$ is ordinary.

- Given $(m, \alpha, p)$ is ordinary, let all components in graph $g$ be advertiser resistant
- If there are plentiful lone nodes, advertiser resistance gives the advertiser prefers to attack the lone nodes.
- Once all lone nodes are stable on $y$ the advertiser may gather excess attacks to make up the difference in $U^a(\cdot)$ that occurs taking some non-trivial component.
- It can be shown that the minimal $k$ where this happens may be weakly increased by removing two components in $g$ and creating a new component with fewer nodes.
- We finish optimizing the graph with a single non-trivial component by making making it as near complete as possible.

**Lemma 1:** We introduce an upper bound of

$$W(g^*(M), M) \leq 2m(1 + \alpha) - \frac{k}{3} \left(2\alpha + 2p\right)$$

for $k \in [k_{\text{max}}(M), k_{\text{max}}(A))$.

**Proof:** Observe the cost to take every component in the graph is given by

$$k_r(g) = \sum_{c \in C(g)} k^*(c) \quad (24)$$

Noting $\max_{\alpha, \pi} k^*(c) = n_c + m_c$ and

$$\sum_{c \in C(g)} m_c = m \land \sum_{c \in C} n_c = 2m$$

by problem parameters we may obtain $\max_{\alpha, \pi, g} k_r(g) = 3m$ across all possible graphs.

Now we examine a SPARSE design with $\alpha + 2p > 2$. We compute $k^*(s) = [\alpha + 2p] = 3$ for $c \in C_e(SPARSE)$ so we obtain $k_r(SPARSE) = 3m$.

Let $g'$ be some graph such that $g' \neq SPARSE$. $C_i(g')$ must then be nonempty and by Theorem 7, we know for $\exists c \in C_i(g')$ such that $c \in C_y(g')$. Thus $g'$ features two unique $\frac{\Delta W(c)}{k_r(c)}$ for $c \in C_l(g')$ and $c \in C_e(g')$. By Lemma 2 we know we know $\frac{\Delta W(s)}{k_r(s)} > k^*(c)$ for $s \in C_e(SPARSE)$. Knowing $W(g', M(0)) = W(SPARSE, M(0))$ and $W(g', M(3m)) = W(SPARSE, M(3m))$, we deduce then that $W(c) > \frac{\Delta W(s)}{k_r(s)}$ for $c \in C_e(g')$. Thus $W(SPARSE, M(k)) \geq W(g', M[k])$ for $k \geq k_{\text{max}}(M)$ and $\forall g'$, thus making SPARSE optimal.

Now we define SPARSE* to be a fictional graph identical with SPARSE except it satisfies $k^*(c) = 3$ for $c \in C_e(SPARSE^*)$ for all $\alpha, p$. So $k_r(SPARSE^*) = 3m$ and we may then follow similar arguments that $k_r(g')$ is maximally $3m$ and $g'$ must contain some component satisfying $\frac{\Delta W(c)}{k_r(c)} > \frac{\Delta W(s)}{k_r(s)}$ for $c \in C_e(g'), s \in C_e(SPARSE^*)$. Thus $W(SPARSE^*, M) \geq W(g', M)$ by the same arguments as sparse $g', \alpha, p$.

So we may present the welfare function of SPARSE* as

$$W(g^*(M), M(k)) \leq 2m(1 + \alpha) - \frac{k}{3} \left(2\alpha + 2p\right) \quad (26)$$

**Lemma 2:** When $k < k_{\text{max}}(M)$, the welfare benefit of $g^*(A(k))$ over LINE is $O(\sqrt{m})$.

$$B(k) = W(g^*(A), A(k)) - W(LINE, A(k)) = O(\sqrt{m})$$

**Proof:** First we may appreciate the implications of Lemma 3 such that any candidate optimal graph $g$ must
satisfy \( C_e(g) \cap C_y(g; A(k)) = \emptyset \).

And by Theorem 1, \( k_{\max} \) values were obtained via single nontrivial component graphs. However even the most sparse of these designs, the line graph, leaves \((m - 1)\) nodes unconnected. That is,

\[
W(LINE, A(k)) = 2m(1 + \alpha) - p \min\{k, m - 1\},
\]

(28)

So if we can satisfy the first requirement with a many-tree component graph, we could increase welfare.

Observe that any tree component will cover \( m_e + 1 \) nodes and any component with at least one cycle will cover strictly less. Therefore we can let \( n_e = m_e + 1 \). Note that \( \sum_{e \in C_l(g)} m_e + 1 = m + |C_e(g)| \) thus forest graph \( g \) contains \( |C_l(g)| - |C_l(LINE)| = |C_e(g)| - 1 \) less lone nodes. Thus, an optimal graph \( g \) in this regime is composed entirely of tree components and lose none to an advertiser. That is, it holds that

\[
W(g, A(k)) \leq 2m(1 + \alpha) - p \min\{k, m - 1\} + p(|C_e(g)| - 1).
\]

(29)

This is due to the fact that each additional component protects at most one additional node, potentially improving welfare relative to (28) by an amount \( p \) for some \( k \). Combining (28) and (29), we obtain upper bound.

\[
B(k) \leq (|C_e(g)| - 1)p.
\]

(30)

Consider a situation such that forest graph \( g^* \) has \( |C_e(g^*)| \) non trivial components and \( m - |C_e(g^*)| \) lone nodes. Let \( c \in \arg \min_{g \in C_e(g^*)} k^*(c) \). When \( m - |C_e(g^*)| < k < m - |C_e(g^*)| + |m_e + n_e|p \), suppose the advertiser takes all lone nodes but no component. Then we have

\[
B(k) = 2m(1 + \alpha) - (m - |C_e(g^*)|)p - (2m(1 + \alpha) - kp - m - |C_e(g^*)|)p
\]

\[
= [m, m_e + n_e, p, m - |C_e(g^*)|]p.
\]

(31)

Combining (30) and (31) and noting \( g^* \) is a forest, we have

\[
B(k) \leq p \cdot \min\{|C_e(g^*)| - 1, m_e + \alpha + n_e + p\}.
\]

(32)

Recall that \( |C_e(g^*)| \leq \frac{m}{m_e} \). Then if \( m_e > \sqrt{m} \),

\[
|C_e| \leq \frac{m}{m_e} \leq \frac{m}{\sqrt{m}} = |C_e|.
\]

(33)

That is, either \( m_e > \sqrt{m} \) or \( |C_e(g^*)| \leq \sqrt{m} \). Thus it is clear from (32) that

\[
B(k) = O(\sqrt{m}),
\]

as desired.

**Lemma 3:** If \( c \in C_e(g) \) and \( c' \in C_l(g) \) then \( \Delta W(c) / k^*(c) > \Delta W(c') / k^*(c') \) or equivalently \( \Delta W(c) > k^*(c)p \). Thus we can guarantee system designers have a strong preference that \( C_y(g | T(k)) \subseteq C_l(g) \).

**Proof:** First note that

\[
\Delta W(c') / k^*(c') = p / [0 \cdot \alpha + p] = p,
\]

and

\[
\Delta W(c) / k^*(c) = 2m_e + 2m_e - (2m_e - n_e)p = 2m_e + n_e p.
\]

Thus, we wish to show \( p < \frac{2m_e + n_e p}{m_e - n_e p} \) or equivalently \( |m_e - n_e p| < 2m_e + n_e p \).

Let \( R = |m_e - n_e p| > |m_e - n_e p| \) such that \( R \in [0, 1] \). So we may obtain \( m_e + n_e p, m_e + n_e p \) which may be simplified further into \( R < m_e + n_e p(1 - \alpha) + m_e \alpha (1 - p) \). We may further observe that \( m_e + n_e p(1 - \alpha) + m_e \alpha (1 - p) \) and clearly \( p \geq R \). Thus \( p < \alpha \implies R < m_e + n_e p(1 - \alpha) + m_e \alpha (1 - p) \).

**Lemma 4:** When \( k < k_{\max}(M) \), a line graph is within \( 2p \) of optimal against a malicious adversary for \( k < k_{\max}(M) \):

\[
W(g^*(M(k)), M(k)) - W(LINE, M(k)) \leq 2p.
\]

(34)

and it holds that

\[
W(g^*(M(k)), M(k)) \geq 2m(1 + \alpha) - p \min\{k, m - 1\}.
\]

(35)

**Proof:** First we seek conditions on arbitrary graph \( g \) such that it may satisfy

\[
W(g|M(k)) > W(LINE|M(k))
\]

for some \( k \in [0, k_{\max}(M)] \). Theorem 1 and Lemma 3 give that for \( g \in \arg \min_{g \in C_e(g)} k^*(c) \)

\[
k^*(g) > k
\]

(37)

must hold or (36) will not be satisfied.

We may also deduce that to satisfy (36) there must exist some \( k \) where \( C_y(g|M(k)) < C_y(LINE|M(k)) \) knowing \( C_y(g|M(k)) \leq C_l(g|M(k)) \), thus we can conclude the following two conditions must hold for \( g \) to satisfy (36):

\[
|C_l(g)| < |C_l(LINE)| = m - 1 \quad \text{and} \quad k > |C_l(g)|.
\]

(38)

That is, graph \( g \) has fewer lone nodes than LINE, so there may exist a \( k \) where \( g \) will feature less \( \bar{y} \) lone nodes, and \( k \) must be large enough to be in this zone. Let \( k_g \) be the minimal such \( k \) to satisfy (39) on graph \( g \).

Now we note that \( k^*(g) \) may be increased and \( k_g \) may be decreased by following a simple algorithm where all components are destroyed and replaced with tree components with corresponding edge counts. So we seek forest graphs satisfying \( k^*(g) > k_g \).

On forest graphs we may easily compute \( k_g = 2m - m - |C_e(g)| + 1 = m - |C_e(g)| + 1 \) as tree components always satisfy \( n_e = m_e + 1 \). We may once again optimize forest graph \( g \) then by removing all components except \( q \) and producing as many copies of \( q \) as possible. However, the process will not be feasible if \( m_q < m \), so we use \( m_q \) as an upper bound on \( |C_e(g)| \) while maintaining a given \( k^*(q) \) thus producing lower bound

\[
k_g := m - m_q + 1 + m(m_q - 1)/m_q + 1 \leq k_g.
\]

(40)

Since (40) is independent of \( \alpha, p \), we take \( a = p = 1 \) to deduce upper bound \( k^*(q) = 2m_q + 1 \geq |m_e \alpha + n_e p| = k^*(q) \).

We now show \( k_g \geq k^*(q) \) holds for \( m_q \in \{2, 3, 4, ..., \lfloor \frac{m}{p} \rfloor \} \) by showing \( k_g \geq k^*(q) \). It follows algebraically that

\[
k_g \geq k^*(q) \iff m \geq 2m_q^2/(m_q - 1),
\]

(41)
and the right hand statement is always true since

\[
2 \frac{m_q^2}{m_q - 1} \leq 2 \frac{m_q^2 + m_q - 2}{m_q - 1} = 2m_q + 4 \leq m, \quad (42)
\]

where the inequalities follow from \(2 \leq m_q \leq m/3\) and \(m \geq 12\). Thus (37) never holds for graphs satisfying \(2 \leq m_q \leq m/3\).

If \(m_q = 1\) (i.e., \(g = \text{SPARSE}\)), it can be shown that \(W(\text{SPARSE}, M(K)) - W(\text{LINE}, M(k)) \leq 2p\). If \(m_q > m/3\), the resulting \(g\) can have at most 2 nontrivial components, and it can be shown that \(W(g, M(K)) - W(\text{LINE}, M(k)) \leq p\). \(\blacksquare\)