Technical Appendix

Paper title: Networked Anti-Coordination Games Meet Graphical Dynamical Systems: Equilibria and Convergence.

Symbols	Definition
SACG	Sequential anti-coordination games
SyACG	Synchronous anti-coordination games
SE	Self essential
SN	Self non-essential
SE-SACG	Sequential anti-coordination games under self essential mode
SN-SACG	Sequential anti-coordination games under self non-essential mode
SE-SyACG	Synchronous anti-coordination games under self essential mode
SN-SyACG	Synchronous anti-coordination games under self non-essential mode
SDS	Sequential dynamical system
SyDS	Synchronous dynamical system
IT	Inverted threshold
(SE, IT)-SDS	Sequential dynamical system under self essential mode
(SN, IT)-SDS	Sequential dynamical system under self non-essential mode
(SE, IT)-SyDS	Synchronous dynamical system under self essential mode
(SN, IT)-SyDS	Synchronous dynamical system under self non-essential mode
$\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$	A SyDS $\mathcal{S}$ with underlying graph $G_{\mathcal{S}}$ and set of local functions $\mathcal{F}$
$\mathcal{S}' = (G_{\mathcal{S}'}, \mathcal{F}, \Pi)$	An SDS $\mathcal{S}'$ where Π is the vertex update sequence
n	The number of vertices in $G_{\mathcal{S}}$
m	The number of edges in $G_{\mathcal{S}}$
f_v	Local function of v
$\tau_1(v)$	For vertex v, the threshold value on the number of 0's for f_v to equal 1
$\tau_0(v)$	For vertex v, the threshold value on the number of 1's for f_v to equal 0
$N(v)$	The open neighborhood of v, i.e., the set of vertices adjacent to v
$N^+(v)$	The closed neighborhood of v, i.e., the set of vertices adjacent to v and v itself
$d(v)$	The degree of v
C	A configuration of $\mathcal{S}$
C'	The successor of C
C''	The successor of C'
$C(v)$	The state of v in configuration C of $\mathcal{S}$

Table 3: Symbols and Notation.

Remark. Note that $\tau_1(v)$ uniquely determines $\tau_0(v)$. In particular, $\tau_0(v) + \tau_1(v) = d(v) + 2$ for the *self essential* mode, and $\tau_0(v) + \tau_1(v) = d(v) + 1$ for the *self non-essential* mode. Based on the mapping between anti-coordination games and discrete dynamical systems, we present all proofs in the context of dynamical systems.

A Detailed Discussion of Related Work

It is known that the *self non-essential sequential* anti-coordination games are potential games, and by the argument from Monderer and Shapley (1996), such a game always has an NE. In particular, a potential game admits a potential function that increases when an agent chooses an action that yields a strictly higher payoff (Monderer and Shapley 1996). Given a potential game starting from an initial action profile, consider the sequence of profiles such that at each step, a player updates its action to obtain a strictly higher payoff (if possible). Such a sequence is called an *improvement path*, and for potential games, all the improvement paths are of finite length (known as the *finite improvement property*). More importantly, a maximal improvement path (i.e., an improvement path that goes to a maximum of the potential) always ends at an equilibrium point (Monderer and Shapley 1996). Note that this result does not immediately imply that an NE can be reached in polynomial time, as the number of possible action profiles is exponential in the number of agents. Further, we emphasize that anti-coordination games under *synchronous* update schemes are **not** potential games as limit cycles of length 2 exist.

The book by Goles and Martinez (2013) discussed the phase space properties of dynamical systems, and one can verify that such systems also model *coordination* games. In particular, they prove that for synchronous *coordination* games, the length of any limit cycle is at most 2. Their argument uses a mathematical tool called algebraic invariants, and they show that if we consider each limit cycle as a periodical sequence, then the length of such a sequence is either 1 or 2. In the same work, Goles and Martinez proposed a Lyapunov function for synchronous *coordination games* and show that the best-response dynamics of the game converges to a limit cycle of length at most 2 in a polynomial number of steps. For anti-coordination games, Barrett et al. (2003) study phase space properties of sequential dynamical systems (which includes modeling sequential AC games). Their results imply that when local functions are `nad`'s and `nor`'s (which are inverted-threshold functions), the length of a limit cycle in a *self essential* sequential anti-coordination game can be $2^{O(\sqrt{n})}$ where n is the number of agents. Later, Adam et al. (2012b) use a combinatorial approach and argue that the length of a limit cycle in a synchronous anti-coordination game is at most 2. However, they did not bound the convergence time to reach a limit cycle.

A more recent work by Ramazi, Riehl, and Cao (2016) investigates the convergence time for asynchronous *self non-essential* anti-coordination games. In their asynchronous dynamics, agents are updated at each time step in a random order, and for each agent, the number of steps between any two consecutive updates is guaranteed to be finite. Based on this scheme, they show that under the best-response dynamics, an equilibrium is always reached in finite time.

4 Equilibrium Existence and Finding

In this section, we present the detailed proofs of the results given in section 4 of the main manuscript. We start with a key observation:

Observation 4.1. *A* SyDS *and an* SDS *with the same underlying graph and the same set of local functions have the same set of fixed points.*

Consequently, SN-SyACG (SE-SyACG) and SN-SACG (SE-SACG) have the same complexity for EQE / EQF.

4.1 Intractability for the *self essential* mode

We establish that EQUILIBRIUM EXISTENCE (EQE) is **NP**-hard for *self essential synchronous* anti-coordination games (SE-SyACG), and the problem remains hard on bipartite graph. Further, the corresponding counting problem is **#P**-complete. This immediately implies that EQUILIBRIUM FINDING (EQF) is also hard for SE-SyACG. In particular, Let $\mathcal{S}$ be a (SE, IT)-SyDS that models a SE-SyACG. We show that determining the existence of a fixed point of $\mathcal{S}$ is intractable. By Observation 4.1, it follows that EQE / EQF is also hard for SE-SACG. We now proceed with the proof.

Observation 4.2. *Suppose that a vertex v of a (SE, IT)-SyDS $\mathcal{S}$ has threshold $\tau_1(v) = 1$. Then in every fixed point (if any exists) C of $\mathcal{S}$, $C(v) = 1$ and at least one neighbor of v has state 0 in C.*

Observation 4.3. *Suppose that a vertex v of a (SE, IT)-SyDS $\mathcal{S}$ has threshold $\tau_1(v) = d(v)$. Then in every fixed point (if any exists) C of $\mathcal{S}$, if $C(v) = 1$, then then all neighbors of v have state 0 in C, and if $C(v) = 0$, then at least two neighbors of v have state 1 in C.*

Lemma 4.4. *Let $\mathcal{S}$ be a (SE, IT)-SyDS whose underlying graph $G_{\mathcal{S}}$ is a complete bipartite graph of the form:*

a. The bipartitions $V(G_{\mathcal{S}}) = \{A, B\}$.

b. $|A| = |B| = 3$.

c. $\tau_1(v) = 3,\ \forall v \in V(G_{\mathcal{S}})$.

The phase space of $\mathcal{S}$ has only 2 distinct fixed points for which either (i) vertices in A are in state 1, and vertices in B are in state 0, or (ii) vice versa.

Proof. By the threshold values of vertices in A and in B, it is easy to see that the two configurations given in the Lemma are fixed points. We now argue that the system has no other fixed points. Let C be a configuration of $\mathcal{S}$ that is different from the two fixed points above. Let C' be the successor of C. We first consider the case where both A and B have at least one state-1 vertex under C. Let $v \in A$ be such a vertex in state 1 under C. Note that since B has at least one state-1 vertex, the number of state-0 vertices in v's closed neighborhood is at most $|B| - 1 < \tau_1(v)$. Thus, we have $C'(v) = 0$ and C is not a fixed point. Next, we consider the case where both A and B have at least one state-0 vertex under C. Let v and w be two state-0 vertices in A and B, respectively. Since $\tau_1(u) = 3, \forall u \in V(G_{\mathcal{S}})$, $C(v) = 0$ implies that there

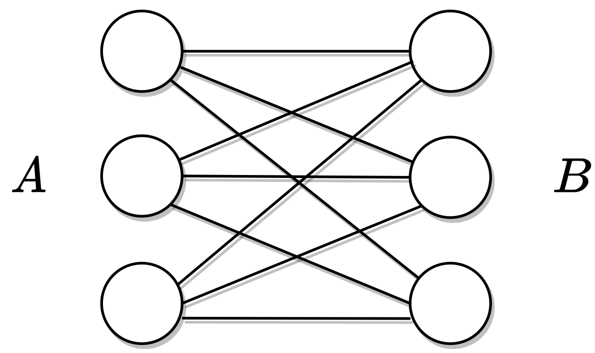

Figure 4: An example (SE, IT)-SyDS $\mathcal{S}$ for Lemma 4.4, where $G_{\mathcal{S}}$ is a complete bipartite graph with bipartitions $\{A, B\}$. The thresholds τ_1 of all vertices are 3.

exists at least two state-1 vertex in B. Similarly, $C(w) = 0$ implies that there exists at least two state-1 vertex in A. By our previous argument, it follows that C is not a fixed point. This concludes the proof. $\square$

We now present the Theorem on the intractability of EQE for SE-SyACG.

Theorem 4.4. *For SE-SyACG, the* EQUILIBRIUM EXISTENCE*(*EQE*) is **NP**-complete and the counting problem* #EQE *is **#P**-complete. Further, the problem remains hard on bipartite graphs.*

Proof. One can easily verify that the problem is in **NP**. We now establish the **NP**-hardness of EQE via a reduction from 3SAT. Let f be the formula for an arbitrary 3SAT instance. We construct a SyDS $\mathcal{S}$ such that $\mathcal{S}$ has a vertex for each literal, a vertex for each clause, and a collection of gadgets. Further, $\mathcal{S}$ is constructed so that there is a one-to-one correspondence between fixed points of $\mathcal{S}$ and satisfying assignments to the given 3SAT formula f.

Literal vertices. For each variable x_i in f, there is a positive literal vertex y_i and a negative literal vertex z_i in $\mathcal{S}$. Both these vertices have threshold $\tau_1(y_i) = \tau_1(z_i) = 3$. Under any configuration, we interpret positive literal vertex y_i having state 0 as corresponding to variable x_i being true, and negative literal vertex z_i having value 0 as corresponding to variable x_i being false. Our Gadgets (shown later) ensure that in any fixed point of $\mathcal{S}$, these two vertices have complementary values.

Clause vertices. For each clause c_j in f, there is a clause vertex w_j, $\tau_1(w_j) = 1$. Further, w_j is adjacent to each of the literal vertices that correspond to literals occurring in clause c_j. There is no other edges incident on w_j. From Observation 4.2, in every fixed point of $\mathcal{S}$, w_j has state 1, and at least one of the adjacent literal vertices has state 0.

Gadgets. We now construct the gadgets, which involve auxiliary vertices for the literal vertices, as follows. For each variable x_i in f, there are three auxiliary vertices, namely a_i, d_i and e_i. In particular,

(i) The vertex a_i is adjacent to both x_i's positive literal vertex y_i and negative literal vertex z_i. Further, $\tau_1(a_i) = 1$. From Observation 4.2, in every fixed point of $\mathcal{S}$, a_i has state 1, and at least one of the two literal vertices y_i and z_i has state 0.

(ii) The vertex d_i and e_i has threshold $\tau_1(d_i) = 1$ and $\tau_1(e_i) = 3$, respectively. Further, we introduce three auxiliary edges (e_i, d_i), (e_i, y_i) and (e_i, z_i). From Observation 4.2, in every fixed point, d_i must has state 1, thus, its neighbor e_i has state 0. Then, from an application of Observation 4.3 to vertex e_i, at least one of the two literal vertices y_i and z_i has state 1.

The combined effect of the auxiliary vertices and edges described thus far is that under every fixed point of $\mathcal{S}$, exactly one of y_i and z_i has state 0, and exactly one has state 1.

Finally, for every literal vertex (positive or negative), denoted by v, we add an auxiliary structure based on the bipartite graph in Lemma 4.4. The structure involves five auxiliary vertices: g_v^1, g_v^2, h_v^1, h_v^2, and h_v^3, each with threshold $\tau_1 = 3$ (which is the degree of each of these vertices). Consider the subsets $A_v = \{v, h_v^1, h_v^2\}$ and $B_v = \{g_v^1, g_v^2, g_v^3\}$. The structure has the following nine auxiliary edges: (v, h_v^1), (v, h_v^2), (v, h_v^3), (g_v^1, h_v^1), (g_v^1, h_v^2), $g_v^1, h_v^3)$, (g_v^2, h_v^1), (g_v^2, h_v^2), (g_v^2, h_v^3). That is, the subgraph of $\mathcal{S}$ induced on $A_v \cup B_v$ is a complete bipartite graph with bipartitions A_v and B_v.

By Lemma 4.4, that in any fixed point of $\mathcal{S}$, either all the vertices in A_v have state 1 and all the vertices in B_v have state 0, or vice versa. To see this, first suppose that in a given fixed point C, at least one of the vertices in B_v has state 1. Then, from Observation 4.3, the three neighbors of this vertex, namely the three vertices in A_v, all have state 0. Consequently, since the threshold of each vertex in B_v is 3, all three vertices in B_v have value 1. Now suppose that none of the vertices in B_v has value 1, i.e., they all have value 0. Since the threshold of each vertex in A_v is 3, all three vertices in A_v have value 1.

This completes the construction of $\mathcal{S}$ which clearly takes polynomial time. Further, the resulting graph is bipartite. We now claim that there is a one-to-one correspondence between fixed points of $\mathcal{S}$ and satisfying assignments of f.

$(\Rightarrow)$ Let α be a satisfying assignment of f. We construct a configuration C_α of $\mathcal{S}$ as follows. For each variable under α, if $\alpha[x_i] = \text{TRUE}$, then the positive literal vertex y_i has the state $C_\alpha(y_i) = 0$, and the negative literal vertex z_i has the state $C_\alpha(z_i) = 1$. On the other hand, if If $\alpha[x_i] = \text{FALSE}$, then $C_\alpha(y_i) = 1$ and $C_\alpha(z_i) = 0$. Further, every clause vertex w_j has state 1. As for the states of auxiliary vertices for each x_i, we set a_i and d_i to state 1, and e_i to state 0. Lastly, for each literal vertex v, the other two members of A_v have the same value as v, and the three members of B_v have the complement of the value of v under C. This completes the specification of C_α. By checking the number of state-0 vertices in the closed neighborhood of each vertex, it can be verified that C_α is a fixed point of $\mathcal{S}$.

$(\Leftarrow)$ Let let C be a fixed point of $\mathcal{S}$. Let α_C be the assignment to f where $\alpha_C(x_i) = 1$ iff $C_\alpha[y_i] = 0$. Since C is a fixed point, every clause vertex is adjacent to at least one literal vertex with the value 0. Thus, α_C is a satisfying assignment of f.

We now have established that determining if $\mathcal{S}$ has a fixed point is **NP**-complete, even when the underlying graph is bipartite. Further, it can be verified that $C = C_{\alpha_C}$, so the reduction is parsimonious. The

NP-hardness of EQE and the **#P**-hardness of the counting version #EQE for SE-SyACG immediately follows. This concludes the proof. □

Corollary 4.5. *For SE-SACG, the* EQUILIBRIUM EXISTENCE*(*EQE*) is **NP**-complete and the counting problem* #EQE *is **#P**-hard. Further, the problem remains hard on bipartite graphs.*

Remark. A fixed point under the SE mode is generally **not** a fixed point under the SN mode. Therefore, the above hardness result does not imply a hardness of EQE for the SN anti-coordination games.

4.2 Finding NE under the *self non-essential* mode

As pointed out in (Vanelli et al. 2020), a *self non-essential sequential* anti-coordination game (SN-SACG) always has a pure Nash equilibrium (NE). By Observation 4.1, a *self non-essential synchronous* anti-coordination game (SN-SyACG) also always has an NE. In this section, we explore beyond the existence problem and tackle the problem of finding an NE in a for SN mode under arbitrary network topology (i.e., EQF). In particular, inspired by the potential function approach developed in (Barrett et al. 2006), given a (SN, IT)-SDS $\mathcal{S}'$ (modeling a SN-SACG), we show that starting from an arbitrary configuration, a fixed point of $\mathcal{S}'$ is always reached in at most $3m - n$ steps. Since each step of $\mathcal{S}'$ can be carried out in $O(m)$ time, a fixed point of $\mathcal{S}'$ is then found in $O(m^2)$ time. Consequently, a fixed point of a (SN, IT)-SyDS $\mathcal{S}$ (modeling a SN-SyACG) can also be founded in $O(m^2)$ time.

We note that our results does not follow from (Barrett et al. 2006) since (i) we study anti-coordination games, and the results in (Barrett et al. 2006) are for *coordination* games; (ii) in the *self non-essential* mode, each vertex does not consider its own state while playing the game, whereas (Barrett et al. 2006) focuses on the case where its own state is considered by each vertex.

If $\tau_1(v) = 0$ or $\tau_1(v) = d_v + 1$ for some vertex v, then v is a vertex whose state is constant after at most one transition. Thus, we can remove v from the graph and update the threshold of neighbors correspondingly without affecting system dynamics. Without loss of generality, we assume that there are no constant vertices, that is $1 \leq \tau_1(v) \leq d(v), \forall v \in V(G_{\mathcal{S}'})$.

The potential functions and bounds

Let $\mathcal{S}' = (G_{\mathcal{S}'}, \mathcal{F}', \Pi)$ be a (SN, IT)-SDS that models a SN-SACG. Given a configuration C, We now define the potentials of vertices, edges, and the system under C.

The vertex potential. Given a vertex $u \in V(G_{\mathcal{S}'})$, the potential of u under C is defined as follows

$$\mathcal{P}(C, u) = \begin{cases} \tau_0(u) & \text{if } C(u) = 0 \\ \tau_1(u) & \text{if } C(u) = 1 \end{cases}$$

The edge potential. Given an edge $e = (u, v) \in E(G_{\mathcal{S}'})$, the potential of e under C is defined as follows

$$\mathcal{P}(C, e) = \begin{cases} 1 & \text{if } C(u) = C(v) \\ 0 & \text{if } C(u) \neq C(v) \end{cases}$$

The configuration potential. The potential of the system $\mathcal{S}'$ under C is defined as the sum of the vertex potentials and edge potentials over all vertices and edges.

$$\mathcal{P}(C,\mathcal{S}') = \sum_{u\in V(G_{\mathcal{S}'})} \mathcal{P}(C,u) + \sum_{e\in E(G_{\mathcal{S}'})} \mathcal{P}(C,e)$$

A lower bound on the configuration potential. We first establish a lower bound of the $\mathcal{P}(C,\mathcal{S}')$ for any configuration C.

Lemma 4.6. *For any configuration C of $\mathcal{S}'$, we have*

$$\mathcal{P}(C,\mathcal{S}') \geq \left(\sum_{u\in V(G_{\mathcal{S}'})} \min\{\tau_0(u), \tau_1(u)\} \right) + \gamma$$

where γ is the minimum number of edges whose endpoints have the same color in $G_{\mathcal{S}'}$, over all 2-coloring of $V(G_{\mathcal{S}'})$.

Proof. Given any configuration C, since the potential of each vertex is either $\tau_1(u)$ or $\tau_0(u)$, it immediately follows that $\sum_{u\in V(G_{\mathcal{S}})} \min\{\tau_0(u), \tau_1(u)\}$ is the lower bound on the sum of potentials from all vertices.

As for the bound on the sum of edge potentials, remark that C corresponds to a 2-coloring of $V(G_{\mathcal{S}'})$ (i.e., each state represents a color). Thus, the minimum number of edges whose endpoints have the same state under C is at most γ. Since each edge with the same-state endpoints has potential value 1, it follows that the sum of edge potential under any configuration C is at least γ. This concludes the proof. □

An upper bound on the configuration potential. Next, we present the upper bound of the configuration potential. Given an arbitrary configuration C

Lemma 4.7. *The configuration potential satisfies*

$$\mathcal{P}(C,\mathcal{S}') \leq 3m$$

Proof. Observe that under an arbitrary configuration of $\mathcal{S}'$, the sum of vertex potential satisfies

$$\sum_{u\in V(G_{\mathcal{S}'})} \mathcal{P}(C,u) \leq \sum_{u\in V(G_{\mathcal{S}'})} \max\{\tau_0(u), \tau_1(u)\} \leq \sum_{u\in V(G_{\mathcal{S}'})} d(u) = 2m$$

As for the sum of edge potential $\sum_{e\in E(G_{\mathcal{S}'})} \mathcal{P}(C,e)$, it is easy to see that the upper bound is m. The upper bound of the overall configuration potential follows immediately. □

Overall, We have shown that the gap in the configuration potential between an arbitrary initial configuration and a system's converged configuration is at most

$$3m - \sum_{v\in V(G_{\mathcal{S}})} (\min\{\tau_0(v), \tau_1(v)\}) - \gamma \leq 3m - n \tag{5}$$

Decrease of the potential after each update

We establish that starting from an arbitrary initial configuration that is not a fixed point, the configuration potential of $\mathcal{S}'$ decreases by at least 1 after each vertex switches its state. Given that the potential gap is at most $3m - n/2$, it follows that the system reaches a fixed point in at most $3m - n/2$ time steps.

Lemma 4.8. *Given a configuration C of $\mathcal{S}'$ that is not a fixed point. Let $\tilde{C}$ be a configuration that results from the state change of a single vertex u due to the dynamics of $\mathcal{S}'$. We then have $\mathcal{P}(C, \mathcal{S}') - \mathcal{P}(\tilde{C}, \mathcal{S}') \geq 1$.*

Proof. Since u is the only vertex that undergoes the state change, the overall configuration potential is affected by only the change of u's potential and the potentials of edges incident to u. Let $d_0(u)$ and $d_1(u)$ denote the number of u's neighbors in state-0 and the number of u's neighbors in state-1 under C, respectively. Without loss of generality, suppose u changes its state from $C(u) = 0$ to $\tilde{C}(u) = 1$. Subsequently, the sums of potentials of u and edges incident to u under C and $\tilde{C}$ are

$$\mathcal{P}(C, u) + \sum_{e=(u,v), v \in N(u)} \mathcal{P}(C, e) = \tau_0(u) + d_0(u)$$

and

$$\mathcal{P}(\tilde{C}, u) + \sum_{e=(u,v), v \in N(u)} \mathcal{P}(\tilde{C}, e) = \tau_1(u) + d_1(u)$$

respectively. Since $\tilde{C}(u) = 1$, it follows that $d_0(u) \geq \tau_1(u)$ and $d_1(u) \leq \tau_0(u) - 1$. Therefore, we have

$$\begin{aligned}
\mathcal{P}(C, \mathcal{S}') - \mathcal{P}(\tilde{C}, \mathcal{S}') &= \mathcal{P}(C, u) + \sum_{e=(u,v), v \in N(u)} \mathcal{P}(C, e) - \left(\mathcal{P}(\tilde{C}, u) + \sum_{e=(u,v), v \in N(u)} \mathcal{P}(\tilde{C}, e) \right) \\
&= \tau_0(u) + d_0(u) - \tau_1(u) - d_1(u) \\
&\geq 1
\end{aligned}$$

This concludes the proof. $\square$

In summary, we have shown that the potential gap between any two configurations is at most $3m - n$. Further, each change of vertex state due to the system dynamic decreases the overall potential by at least one. Since in each time step (before reaching a fixed point), at least one vertex updates its state, Lemma 4.9 immediately follows.

Lemma 4.9. *For (SN, IT)-SDS, starting from an arbitrary initial configuration, the system dynamics reaches a fixed point in at most $3m - n$ time steps.*

In each time step, we compute the local functions of all vertices to determine the successor configuration, which takes $O(m)$ time. Therefore, a fixed point of $\mathcal{S}'$ can be obtained in $O(m^2)$ time. Given that a fixed point of a (SN, IT)-SDS $\mathcal{S}'$ is also a fixed point of its twin (SN, IT)-SyDS $\mathcal{S}$, we also establish the tractability of finding an NE for (SN, IT)-SyDS. Our results for SN anti-coordination games follow.

Theorem 3.10. *For both SN-SyACG and SN-SACG, we can find a pure Nash equilibrium of the game in $O(m^2)$ time.*

4.3 Finding Nash equilibria under special cases

We have established the intractability of EQE / EQF for *self essential* (SE) anti-coordination games, whereas a *self non-essential* (SN) game admits a polynomial time algorithm for finding an NE. In this section, we identify several special classes of the problem instances such that an NE (if any) can be found in polynomial time for SE anti-coordination games. Further, We extend the results of some special classes to SN anti-coordination games such that an NE can be found in linear time. Based on the connection between anti-coordination games and dynamical systems, we present all proofs in the context of synchronous dynamical systems (and the results for sequential systems follow).

Inclination for one action over another

We consider the special case where agents are inclined to choose one action over the other. Specifically, during the game evolution, each agent u either (i) chooses the action 1 in the next time step if at least one agent in u's open/closed neighborhood chose action 0 in the previous time step; (ii) chooses the action 0 in the next time step if at least one agent in u's open/closed neighborhood chose action 1 in the previous time step.

Observe that the case (i) above corresponds to $\tau_1(u) = 1$. Further, the case (ii) implies $\tau_1(u) = d(v)$ for the SN game, and $\tau_1(u) = d(v) + 1$ for the SE game. We call the local function of a vertex u with $\tau_1(u) = 1$ a NAND function, and of u with $\tau_1(u) = d(v)$ / $\tau_1(u) = d(v) + 1$ a NOR vertex function.

Theorem 4.11. *For SE anti-coordination games and SN anti-coordination games, an NE can be found in $O(m + n)$ time if the corresponding local functions of vertices are* NAND*'s and* NOR*'s.*

Proof. We first present the result for SE-SyACG, modeled by a (SE, IT)-SyDS $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$. Observe that under any fixed point of $\mathcal{S}$, a NAND vertex must be in state 1, and a NOR vertex must be in state 0. This uniquely determines a configuration C. We can then compute C's successor C', and examine if C is a fixed point in $O(m + n)$ time. We further establish the claim below

Claim 4.11.1. *Such a configuration C is a fixed point if and only if each* NOR *vertex is adjacent to at least one* NAND *vertex and vise versa.*

For sufficiency, suppose C is a fixed point. If there exists a NOR vertex v whose neighbors are all NOR vertices. Then the number of state-0 vertices in v' closed neighborhood is $d(v) + 1$ (since C is a fixed point, all NOR vertices are in state 0) which equals to the threshold of v. Thus, $C(v) = 0 \neq C'(v) = 1$ and C is not a fixed point. An analogous argument applies to the case where v is a NAND vertex whose neighbors are all NAND vertices. Specifically, the number of state-0 vertices in v's closed neighborhood is 0. Thus, $C(v) = 1 \neq C'(v) = 0$. As for the necessity of the condition, observe that if a NOR vertex v is adjacent to at least one NAND vertex, the number of state-0 vertices in v's closed neighborhood is at most $d(v) < \tau_1(v)$, thus, $C(v) = C'(v) = 0$. Similarly, if a NAND vertex v is adjacent to at least one

NOR vertex, then the number of state-0 vertices in v's closed neighborhood is at least $\tau_1(v) = 1$, thus, $C(v) = C'(v) = 1$. This concludes the proof for SE-SyACG (and thus also SE-SACG).

We now establish the result for a SN-SyACG, modeled by a (SN, IT)-SyDS, $\bar{\mathcal{S}} = (G_{\bar{\mathcal{S}}}, \mathcal{F})$. Overall, we first construct a candidate configuration C and then modify C to make it a fixed point of $\mathcal{S}$. In particular, for each vertex $v \in V(G_{\bar{\mathcal{S}}})$, if v is a NAND vertex (i.e., $\tau_1(v) = 1$), we set $C(v) = 1$. On the other hand, if v is a NOR vertex (i.e., $\tau_1(v) = d(v)$), set $C(v) = 0$. Based on the same argument in Claim 4.11.1, it follows that C is a fixed point of $\mathcal{S}'$ if and only if each NAND vertex is adjacent to at least one NOR vertex and visa versa. If this condition does not hold, however, there must exist at least one NAND (NOR) vertex whose neighbors are all NAND (NOR) vertices. Subsequently, We further modify C as follows. First, for each NAND vertex v whose neighbors are all in state 1 under C, we set $C(v) = 0$. Let V_{nand} denote such a set of vertices. Further, for each NOR vertex v whose neighbors are all in state 0 under C, we set $C(v) = 1$. Let V_{nor} denote this set of vertices. The pseudocode is shown in Algorithm 1.

Algorithm 1: EQF_NAND_NOR_SN($\mathcal{S}'$)

Input: A (SN, IT)-SyDS $\bar{\mathcal{S}} = (G_{\bar{\mathcal{S}}}, \mathcal{F})$, where $\tau_1(v) = 1$ or $d(v)$, $\forall v \in V(G_{\bar{\mathcal{S}}})$
Output: A fixed point C of $\mathcal{S}'$

$C \leftarrow$ an initial configuration of all 0's
for $v \in V(G_{\mathcal{S}'})$ **do**
  if $\tau_1(v) = 1$ **then** $C(v) \leftarrow 1$ ▷ v is a NAND vertex
  else if $\tau_1(v) = d(v)$ **then** $C(v) \leftarrow 0$ ▷ v is a NOR vertex
end for
for $v \in V(G_{\mathcal{S}'})$ **do**
  if $\tau_1(v) = 1$ **and** $C(w) = 1$ for all neighbors $w \in N(v)$ **then** $C(v) \leftarrow 0$ ▷ $V_{nand} = V_{nand} \cup \{v\}$
  if $\tau_1(v) = d(v)$ **and** $C(w) = 0$ for all neighbors $w \in N(v)$ **then** $C(v) \leftarrow 1$ ▷ $V_{nor} = V_{nor} \cup \{v\}$
end for
return C

We now argue that the resulting configuration C is a fixed point. Let C' be the successor of C. First, observe that for each NAND vertex $u \in V(G_{\mathcal{S}}) \setminus (V_{nand} \cup V_{nor})$, we have $C(u) = 1$ (since we never change the state of vertices that are not in V_{nand} or V_{nor}). Furthermore, u must be adjacent to at least one state-0 vertex, or else u will be in V_{nand}. Similarly, for each NOR vertex $u \in V(G_{\mathcal{S}}) \setminus (V_{nand} \cup V_{nor})$, we have that $C(u) = 0$ and u is adjacent to at least one state-1 vertex. It immediately follows that $C(u) = C'(u),\ \forall u \in V(G_{\mathcal{S}}) \setminus (V_{nand} \cup V_{nor})$. We now consider the states of vertices in V_{nand} or V_{nor}. For each vertex $v \in V_{nand}$, observe that neighbors of v must all be NAND vertices, or else, v is adjacent to at least one state-0 vertex (i.e., a NOR vertex) which contradicts the fact that $v \in V_{nand}$. Furthermore, we claim that

Claim 4.11.2. *Each neighbor of v is not in V_{nand}, $\forall v \in V_{nand}$.*

For contradiction, suppose v' is a neighbor of v who is also in V_{nand}. If v' is added to V_{nand} before v, then v cannot also be in V_{nand} since v has v' as a state-0 neighbor under C. Similarly, if v is added to V_{nand} before v', then v' cannot also be in V_{nand}. This concludes the claim. Claim 4.11.2 implies that v has no state-0 neighbor under C, thus, $C(v) = C'(v) = 0$. By a similar argument, for each vertex $v \in V_{nor}$, neighbors of v must all be NOR vertices, or else, v is adjacent to at least one state-1 vertex (i.e., a

NAND vertex). Moreover, each neighbor of v is not in V_{nor}. It follows that the number of state-0 neighbors of v is $d(v)$, and $C(v) = C'(v) = 1$. This concludes the correctness for Algorithm 1. As for the time complexity, the for loop from line 2 to 5 takes $O(n)$ time, and the for loop from line 6 to 9 takes $O(m+n)$ time. Therefore, the overall running time of Algorithm 1 is $O(m + n)$. The results for SN-SyACG and SN-SACG follow. □

The underlying graph is a DAG

Under a directed graph, at each time step, an agent in the *self essential* mode considers its own state and the state of *in-neighbors*. Similarly, each agent in the *self non-essential* mode only considers the states of its *in-neighbors*.

Theorem 4.12. *For both the SE anti-coordination game and the SN anti-coordination game, an NE can be found in $O(m + n)$ time if the underlying graph is a DAG.*

Proof. We first present the result for SE-SyACG modeled by a (SE, IT)-SyDS $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F}')$. Let V' be the set of vertices with 0 indegree in $G_{\mathcal{S}}$. Note that $V' \neq \emptyset$, or else, $G_{\mathcal{S}}$ contains directed cycles. We first remark that $\mathcal{S}$ has no fixed points if there exists a vertex $v \in V'$ that is not a constant vertex (i.e., $\tau_1(v) = 1,\ \exists v \in V'$). For contrapositive, suppose $\tau_1(v) = 1$ for some $v \in V'$. Given any configuration C where $C(v) = 1$, it follows that $C'(v) = 0$. Similarly, if $C(v) = 0$, then $C'(v) = 1$ for such a vertex v. Thus, any configuration C of $\mathcal{S}$ cannot be a fixed point.

Suppose all vertices in V' are constant vertices. Our algorithm constructs a potential fixed point C as follows. For each vertex $v \in V'$, if $\tau_1(v) = 0$ (i.e., v is a constant-1 vertex), we set $C(v) = 1$, and remove v from $G_{\mathcal{S}}$. On the other hand, if $\tau_1(v) = 2$ (i.e., v is a constant-0 vertex), we set $C(v) = 0$, then decrease the threshold values of all v's out-neighbors by 1 and remove v from $G_{\mathcal{S}}$. Note that after removing v, we might set some other vertices in $V(G_{\mathcal{S}}) \setminus V'$ to have 0 in-degree. Subsequently, we add these new 0 in-degree vertices to V'. The pseudocode is shown in Algorithm 2.

If at any iteration of the algorithm, we found a vertex in V' to have threshold 1 (i.e., a non-constant vertex), the algorithm terminates and concludes that $\mathcal{S}$ has no fixed points. On the other hand, if all vertices in V' are constant vertices, it follows that the algorithm uniquely determines a fixed point C of $\mathcal{S}$. As for the running time, we may consider the algorithm as a breadth-first search process that takes $O(m + n)$ time. This concludes the proof SE-SyACG (and thus SE-SACG).

As for the SN anti-coordination games, note that a 0-indegree vertex v is always a constant vertex, irrespective of $\tau_1(v)$. Thus, a SN anti-coordination game (either SN-SyACG or SN-SACG) has a unique equilibrium which is determined by the threshold of each vertex. This concludes the proof. □

The underlying graph has no even cycles

We first introduce three characterizations of vertices.

Definition 11 (Terminal vertices). *Let G be an undirected graph with no even cycles. A vertex v is a* ***terminal vertex*** *if v is not in any cycles, and v is on the path between two cycles.*

Algorithm 2: $\texttt{EQF\_DAG\_SE}(\mathcal{S})$

Input: A (SE, IT)-SyDS $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$, where $G_{\mathcal{S}}$ is a DAG
Output: A fixed point C of $\mathcal{S}$ (if exists)

$C \leftarrow$ an initial configuration of all 0's
$V' \leftarrow$ the set of vertices with 0 indegree in $G_{\mathcal{S}}$
for $v \in V'$ **do**
    if $\tau_1(v) = 1$ **then return** NULL ▷ The system has no fixed points
    else if $\tau_1(v) = 0$ **then** $C(v) \leftarrow 1$
    else if $\tau_1(v) = 2$ **then**
        $C(v) \leftarrow 0$
        for $w \in N(v)$ **do**
            if $\tau_1(w) \neq 0$ **then** $\tau_1(w) \leftarrow \tau_1(w) - 1$
        end for
    end if
    $V' \leftarrow V' \setminus \{v\}$
    $V'' \leftarrow$ the set of new vertices with 0 in-degree
    $V' \leftarrow V' \cup V''$
end for
return C

Definition 12 (Gate vertices). *Let G be an undirected graph with no even cycles. A vertex v is a **gate vertex** if v is on at least one cycle and either (i) adjacent to a terminal vertex or (ii) adjacent to another vertex on another cycle, or (iii) on at least two cycles.*

Intuitively, *gate vertices* for a cycle $\mathcal{C}$ act as "entrances" on $\mathcal{C}$, such that a transversal from any other cycles to $\mathcal{C}$ must reach one of the gate vertices.

Definition 13 (Tree vertices). *Let G be an undirected graph with no even cycles. A vertex v is a **tree vertex** if v is not on any cycles and v is not a terminal vertex.*

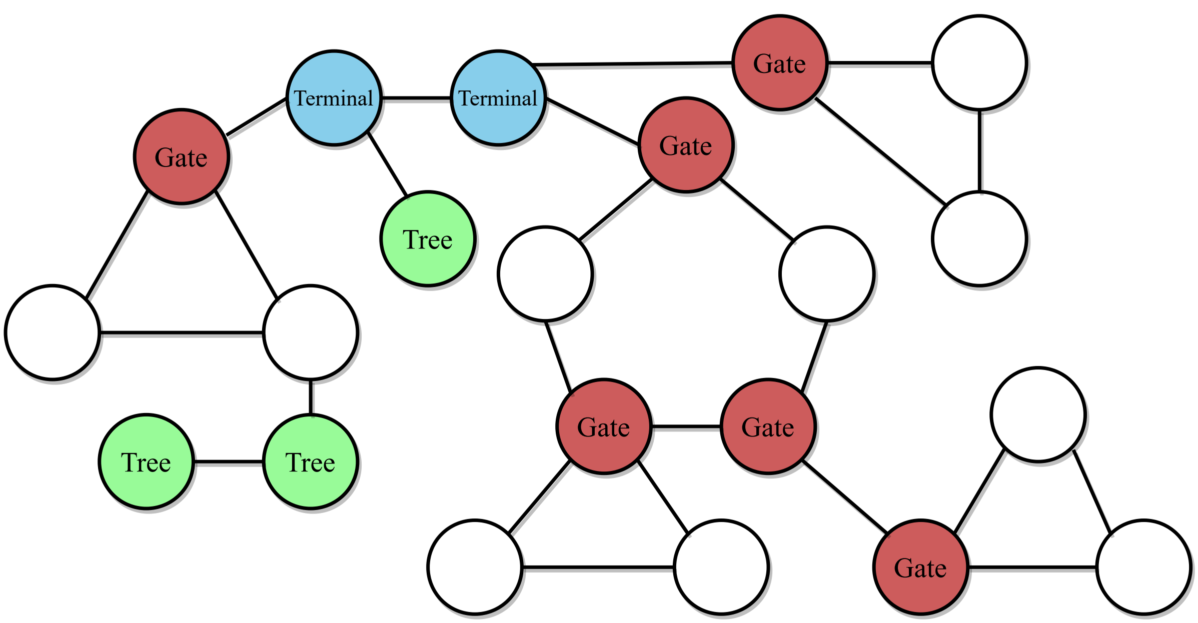

Figure 5: An example graph G where blue vertices are *terminals*, green vertices are *gates*, and red vertices are *tree* vertices.

Lemma 4.13. *Given an undirected graph G with no even cycles, all cycles in G are edge-disjoint.*

Proof. For contrapositive, suppose there exists two different odd cycles, denoted by $\mathcal{C}$ and $\mathcal{C}'$, such that they share a common path $\mathcal{L} = (v_1, ..., v_l)$ of length $l \geq 1$. Observe that the set of vertices $V(\mathcal{C}) \cup V(\mathcal{C}') \setminus V(\mathcal{L})$ form another cycle, denoted by $\mathcal{C}''$. Let $2k+1$ and $2k'+1$ be the length of $\mathcal{C}$ and $\mathcal{C}'$, respectively, $k, k' \geq 1$. It follows that the length of $\mathcal{C}''$ is $2k+1+2k'+1-2l = 2(k+k'+1-l)$ which is even. Proved by contraposition. □

Lemma 4.14. *Let G be an undirected graph with no even cycles, then there must exist a cycle $\mathcal{C}$ with at most one gate vertex.*

Proof. Given any cycle $\mathcal{C}$, let v be a gate vertex of $\mathcal{C}$. By the definition of gate vertices, remark that there exists at least one path $\mathcal{P}$ from v to another cycle[3], where $\mathcal{P}$ intersects with $\mathcal{C}$ only on v. For contradiction, suppose all cycles have at least two gate vertices. Let $\mathcal{C}_1$ be any cycle in G, and denoted by v_1 a gate vertex of $\mathcal{C}_1$. We label all vertices in C_1 as *visited*, and consider a depth-first search (DFS) process from v_1 that traverses unvisited vertices until another gate vertex is found (i.e., a new cycle is reached). During the DFS process, we label traversed vertices as *visited*. Let v_2 be the gate vertex that the DFS (from v_1) encounters, and let $\mathcal{C}_2$ be a cycle that contains v_2. We label vertices in $\mathcal{C}_2$ also as visited. Since all cycles have at least 2 gate vertices, let v_2' be another gate vertex of $\mathcal{C}_2$. Remark that there must **not** exists a path that consists of only *unvisited* vertices (excepts the two endpoints) from v_2' to any visited vertices, or else, there exists a cycle that share common edges with $\mathcal{C}_2$ which contradicts G being even-cycle free (Lemma 4.13). It follows that if we continue the DFS process from v_2' while traversing unvisited vertices, it must reach a new gate vertex v_3 on a cycle $\mathcal{C}_3$. Let v_3' be another gate vertex of $\mathcal{C}_3$. Similarly, there must **not** exists a path that consists of only *unvisited* vertices (except the two endpoints) from v_3' to any visited vertices. Overall, given a newly visited gate vertex v_k on a cycle C_k, $k \geq 2$. Let v_k' be another gate vertex of C_k. By induction, there exists a path with only unvisited vertices (except v_k' itself who is visited) from v_k' to a cycle that is different from C_i, $i = 1, ..., k-1$. However, this contradicts G being finite (and thus the number of cycles is finite). The Lemma immediately follows. □

Now consider a (SE, IT)-SyDS $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$ that models a SE-SyACG.

Lemma 4.15. *Let $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$ be a (SE, IT)-SyDS where the underlying graph has no even cycles. A tree vertex $v \in V(G_{\mathcal{S}})$ has the same state over any fixed point C.*

Proof. We first argue that if a vertex v has degree 1, then v has the same state under any fixed point C. In particular, note that the threshold $\tau_1(v)$ could be either 0, 1, 2 or 3. If $\tau_1(v) = 0$ or $\tau_1(v) = 3$, then v is a constant vertex whose state is uniquely determined by $\tau_1(v)$. On the other hand, by Observation 4.2 and 4.3, we know that $C(v) = 1$ if $\tau_1(v) = 1$ and $C(v) = 0$ if $\tau_1(v) = 2$.

[3] If v is on more than one cycle, then the path consists of only vertex v itself.

Observe that if there exists at least one tree vertex in $G_{\mathcal{S}}$, then at least one of the tree vertices has degree 1. Let v be any tree vertex with degree 1. Based on the claim above, we know that the state of v under any fixed point C is predetermined. Subsequently, we can effectively remove v from the graph and update the threshold values of v's neighbor accordingly (i.e., decrease the neighbor's threshold by one if $C(v) = 0$). By recursion, it follows that the state of any tree vertex v is the same over any fixed point of $\mathcal{S}$. $\square$

Lemma 4.16. *Let $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$ be a (SE, IT)-SyDS where the underlying graph has no even cycles. Let $\mathcal{C}$ be a cycle in $G_{\mathcal{S}}$ with at most one gate vertex. Let $u \in \mathcal{C}$ be a non-gate vertex on $\mathcal{C}$, then the state of u is the same under any fixed point.*

Proof. By Lemma 4.14, we know that such a cycle $\mathcal{C}$ with at most one gate vertex must exist. Given a non-gate vertex u on $\mathcal{C}$, observe that any of u's neighbors that are not on $\mathcal{C}$ must be tree vertices. Furthermore, by Lemma 4.15, the states of all tree vertices are predetermined under a fixed point. Thus, we can consider the graph where all the tree vertices are removed, and the threshold values of their non-tree neighbors are updated accordingly. Subsequently, all non-gate vertices on $\mathcal{C}$ have degree 2.

Claim 4.16.1. *If $\mathcal{S}$ has a fixed point, then at least one non-gate vertex on $\mathcal{C}$ does not have threshold 2.*

For contradiction, suppose all the non-gate vertices on $\mathcal{C}$ have thresholds 2. Since $\mathcal{C}$ is of odd length, it cannot be 2-colored. Thus, under any configuration C, there must exist an edge (u, v) on $\mathcal{C}$ where the state of u and v are the same. Furthermore, at least one of them is a non-gate vertex with threshold 2. W.o.l.g., let u be such a vertex. Since u has degree 2, if $C(u) = C(v) = 1$, then the number of state-0 vertices in the closed neighborhood of u is at most 1, which is less than $\tau_1(u)$. Thus, $C(u) = 1 \neq C'(u) = 0$ where C' is the successor of C. Similarly, if $C(u) = C(v) = 0$, we have $C(u) = 0 \neq C'(u) = 1$. Overall, it follows that $\mathcal{S}$ does not have a fixed point if all the non-gate vertices on $\mathcal{C}$ have thresholds 2.

Claim 4.16.1 implies that if $\mathcal{S}$ has a fixed point, then at least one non-gate vertex u on $\mathcal{C}$ has a threshold not equal to 2. We now argue that the state of u remains the same under any fixed point C of $\mathcal{S}$. In particular, if $\tau_1(u) = 0$ or $\tau_1(u) = 4$, then u is a constant vertex whose state is predetermined. On the other hand, if $\tau_1(u) = 1$, then $C(u) = 1$, and if $\tau_1(u) = 3 = d(v) + 1$, then $C(u) = 0$. It follows that the state of u is the same under any fixed point and is predetermined by $\tau_1(v)$. Thus, we may remove u from the graph and update the thresholds of u's neighbors accordingly. Remark that after removing u, all non-gate vertices on $\mathcal{C}$ become tree vertices. Subsequently, by Lemma 4.15, the state of all non-gate vertices on $\mathcal{C}$ are predetermined, and are the same over any fixed point of $\mathcal{C}$. $\square$

Theorem 4.17. *For a SE anti-coordination game, an NE can be found in $O(m+n)$ time if the underlying graph has no even cycles.*

Proof. Let $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$ be a (SE, IT)-SyDS that models a SE-SyACG. Suppose $\mathcal{S}$ has a fixed point, denoted by C. By Lemma 4.15, we can determine the state of all the tree vertices under C based on their threshold values, and remove all the tree vertices from $G_{\mathcal{S}}$. Let $\mathcal{C}_1$ be a cycle that consists of only one gate vertex. By Lemma 4.14, we can also determine the states of all non-gate vertices on $\mathcal{C}_1$ and effectively eliminate the cycle $\mathcal{C}_1$. Let $G_{\mathcal{S}1}$ be the resulting graph which is still even-cycle free. If $G_{\mathcal{S}1}$ contains other cycles, then there must exists another cycle that consists of only one gate vertex. By recursively determining the state of non-gate vertices on single-gate cycles and the states of tree vertices, it follows that we can determine the states of all vertices under C in time $O(m+n)$. Since a fixed point of $\mathcal{S}$ corresponds to an NE of the underlying SE-SyACG, we conclude that an NE can be found in $O(m+n)$ time for a SE-SyACG (and thus for SE-SACG) when the underlying graph has no even cycles. This concludes the proof. □

With the same argument, we also establish the same result for SN anti-coordination games.

Theorem 4.18. *For a SN anti-coordination game, an NE can be found in $O(m+n)$ time if the underlying graph has no even cycles.*

The underlying graph is complete

Theorem 4.19. *For a SE anti-coordination game, an NE can be found in $O(n)$ time if the underlying graph is a complete graph.*

Proof. Let $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$ be a (SE, IT)-SyDS that models a SE-SyACG. We partition the set of vertices based on their thresholds. Specifically, let k be the number of distinct thresholds τ_1. Let $\mathcal{P} = \{V_1, ..., V_k\}$ be a partition of the vertex set $V(G_{\mathcal{S}})$, such that $\tau_1(u) = \tau_1(v),\ \forall u, v \in V_i,\ i = 1, ..., k$. Furthermore, $\tau_1(u) < \tau_1(v),\ \forall u \in V_i, v \in V_j,\ 1 \leq i < j \leq k$. Remark that since $G_{\mathcal{S}}$ is a complete graph, the closed neighborhoods of all vertices are the same, which is $V(G_{\mathcal{S}})$. Thus, given any fixed point C of $\mathcal{S}$, if $C(v) = 1$ for some $v \in V_j$, then $C(u) = 1, \forall u \in V_i, i \leq j$.

Our algorithm consists of at most $k-1$ iterations, where in each iteration j, $1 \leq j \leq k-1$, we construct a configuration C such that $C(v) = 1, \forall v \in V_i, i = 1, ..., j$, and $C(v) = 0$ otherwise. After each iteration of the algorithm, we check if the resulting C is a fixed point, and return C if so. On the other hand, if all the resulting $k-1$ configurations are not fixed points, we conclude that $\mathcal{S}$ does not have a fixed point. The pseudocode is shown in Algorithm 3.

We establish the correctness of the algorithm. First, we claim that

Claim 4.19.1. *The system $\mathcal{S}$ has a fixed point if and only if there exists a bipartition $\{V', V''\}$ of $V(G_{\mathcal{S}})$, such that $\max\{\tau_1(v) : v \in V'\} \leq |V''| < \min\{\tau_1(v) : v \in V''\}$. Further, such a fixed point C has vertices in V' in state 1, and vertices in V'' in state 0.*

Suppose $\mathcal{S}$ has a fixed point C. Let C' be the successor of C. Let $V' = \{v : C(v) = 1, v \in V(G_{\mathcal{S}})\}$ and $V'' = \{v : C(v) = 0, v \in V(G_{\mathcal{S}})\}$ be a bipartition of the vertex set based on the state of vertices. We argue that $\max\{\tau_1(v) : v \in V'\} < \min\{\tau_1(v) : v \in V''\}$. For contrapositive, assume that there exists a vertex $v \in V'$ such that $\tau_1(v) \geq \tau_1(w)$ for some $w \in V''$. Since C is a fixed point and $C(v) = 1$, we have $|V''| \geq \tau_1(v)$ (i.e., the number of state-0 vertices in C is at least $\tau_1(v)$). However, it follows that

Algorithm 3: `EQF_Complete_SE`($\mathcal{S}$)

Input: A (SE, IT)-SyDS/SDS $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$, where $G_{\mathcal{S}}$ is a complete graph
Output: A fixed point C of $\mathcal{S}$

1: $C \leftarrow$ an initial configuration of all 0's
2: $\mathcal{P} \leftarrow \{V_1, ..., V_k\}$ be a partition of $V(G_{\mathcal{S}})$ such that $\tau_u < \tau_v$ iff $u \in V_i, v \in V_j, 1 \leq i < j \leq k$
3: $\tau_1(V_i) \leftarrow$ the threshold value of vertices in $V_i \in \mathcal{P}, i = 1, ..., k$
4: $a_0 \leftarrow |V(G_{\mathcal{S}})|$ ▷ The number of state-0 vertices in C
5: **for** $j = 1$ **to** $k - 1$ **do**
6:     **for** $v \in V_j$ **do**
7:         $C(v) \leftarrow 1$
8:     **end for**
9:     $a_0 \leftarrow a_0 - |V_j|$
10:     **if** $\tau_1(V_j) \leq a_0 < \tau_1(V_{j+1})$ **then** ▷ Determine if C is a fixed point
        return C
11:     **end if**
12: **end for**
13: **return** NULL ▷ The system has no fixed point

$|V''| \geq \tau_1(v) \geq \tau_1(w)$, thus, $C(w) = 0 \neq C'(w) = 1$ and C is not a fixed point. Overall, we have $\max\{\tau_1(v) : v \in V'\} \leq |V''| < \min\{\tau_1(v) : v \in V''\}$.

As for the necessity of the claim, note that if such a bipartition $\{V', V''\}$ exists, we can construct a fixed point C by assigning $C(v) = 1, \ \forall v \in V'$ and $C(v) = 0 \ \forall v \in V''$. It follows that $|V''| \geq \max\{\tau_1(v) : v \in V'\} \geq \tau_1(v), \ \forall v \in V'$. Thus, $C(v) = C'(v) = 1, \forall v \in V'$. Similarly, we have $|V''| \leq \min\{\tau_1(v) : v \in V''\} \leq \tau_1(v), \ \forall v \in V''$ and $C(v) = C'(v) = 0, \forall v \in V''$. This concludes Claim 4.19.1.

We now argue that the *for loop* of Algorithm 3 from line 5 to 12 essentially discovers if such a bipartition $\{V', V''\}$ exists. In particular, at the j^{th} iteration, $1 \leq j \leq k-1$, denoted by a_0 the number of state-0 vertices in C (after the updates from line 6 to 8). Let $\tau_1(V_j)$ be the threshold value of vertices in $V_j \in \mathcal{P}$. Observe that the algorithm effectively construct a configuration C by setting vertices in $V' = \bigcup_{i=1}^{j} V_i$ to state 1, and vertices in $V'' = \bigcup_{i=j+1}^{k} V_i$ to state 0 .

Claim 4.19.2. *The inequality* $\tau_1(V_j) \leq a_0 < \tau_1(V_{j+1})$ *(i.e., the condition at line 10) is satisfied if and only if* $\max\{\tau_1(v) : v \in V'\} \leq |V''| < \min\{\tau_1(v) : v \in V''\}$.

Observe that $a_0 = |V''|$. Suppose $\tau_1(V_j) \leq a_0 < \tau_1(V_{j+1})$. Since $\tau_1(V_j) \geq \tau_1(v), \ \forall v \in V' = \bigcup_{i=1}^{j} V_i$, it follows that $a_0 = |V''| \geq \max\{\tau_1(v) : v \in V'\}$. Similarly, given that $\tau_1(V_{j+1}) \leq \tau_1(v), \ \forall v \in V'' = \bigcup_{j=i+1}^{k} V_j$, we have $|V''| \leq \min\{\tau_1(v) : v \in V''\}$. For the other direction, by an analogous argument, it is easy to see that if $\max\{\tau_1(v) : v \in V'\} \leq |V''| < \min\{\tau_1(v) : v \in V''\}$, then $\tau_1(V_i) \leq a_0 < \tau_1(V_{i+1})$. The correctness of the algorithm immediately follows.

Overall, the partition $\mathcal{P}$ can be constructed in $O(n)$ time via the bucket sort. Furthermore, the *for loop* at line 7 is called at most $O(n)$ time, and the operations from line 9 to line 12 take constant time. Therefore, the overall running time is $O(n)$. Since a fixed point of $\mathcal{S}$ corresponds to an NE of the underlying game, we conclude that an NE can be found in $O(n)$ time for a SE-SyACG (and therefore for SE-SACG) when the underlying graph is a complete graph. □

With the same argument, we can show that the above result for complete graphs carries over to SN anti-coordination games.

Theorem 4.20. *For a SN anti-coordination game, an NE can be found in $O(n)$ time if the underlying graph is a complete graph.*

4.4 ILP for Finding Nash equilibria in SE mode

In this section, we present an integer linear program formulation that finds a Nash equilibrium (if one exists) for SE anti-coordination games under networks of reasonable sizes. Let $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$ be a (SE, IT)-SyDS that models a SE-SyACG. The proposed ILP constructs a configuration C_{OPT} of $\mathcal{S}$ that maximizes the number of vertices whose states remain unchanged in the successor of C_{OPT}. Subsequently, C_{OPT} is a fixed point if and only if $\mathcal{S}$ has a fixed point (i.e., the states of all vertices remain unchanged). On the other hand, if $\mathcal{S}$ does not have fixed points, our ILP ensures that C_{OPT} is (i) a 2-cycle, and (ii) the number of vertices whose states remain unchanged (under the 2-cycle) is maximized.

The formulation of ILP is presented in (6). In particular, each $u \in V(G_{\mathcal{S}})$ is associated with 4 variables: (i) x_u, (ii) y_u, (iii) a_u, and (iv) b_u. In particular, (i) x_u is the state of u under the constructed configuration C_{OPT}, (ii) y_u is the state of u under the successor of C_{OPT}, denoted by C'_{OPT}, (iii) $a_u = \min\{x_u, y_u\}$, and (iv) $b_u = \max\{x_u, y_u\}$.

$$\max \quad \sum_{u \in V(G_{\mathcal{S}})} a_u - b_u \tag{6a}$$

$$\text{s.t.} \quad \sum_{v \in N^+(u)} 1 - x_v \geq \tau_1(u) \cdot y_u \qquad \forall u \in V(G_{\mathcal{S}}) \tag{6b}$$

$$\sum_{v \in N^+(u)} x_v \geq \tau_0(u) \cdot (1 - y_u) \quad \forall u \in V(G_{\mathcal{S}}) \tag{6c}$$

$$\sum_{v \in N^+(u)} 1 - y_v \geq \tau_1(u) \cdot x_u \qquad \forall u \in V(G_{\mathcal{S}}) \tag{6d}$$

$$\sum_{v \in N^+(u)} y_v \geq \tau_0(u) \cdot (1 - x_u) \qquad \forall u \in V(G_{\mathcal{S}}) \tag{6e}$$

$$a_u \leq x_u \qquad \forall u \in V(G_{\mathcal{S}}) \tag{6f}$$

$$a_u \leq y_u \qquad \forall u \in V(G_{\mathcal{S}}) \tag{6g}$$

$$b_u \geq x_u \qquad \forall u \in V(G_{\mathcal{S}}) \tag{6h}$$

$$b_u \geq y_u \qquad \forall u \in V(G_{\mathcal{S}}) \tag{6i}$$

$$a_u, b_u, x_u, y_u \in \{0, 1\} \qquad \forall u \in V(G_{\mathcal{S}}) \tag{6j}$$

Let x^*_u and y^*_u, $\forall u \in V(G_{\mathcal{S}})$, be optimal state assignments for the variable x_u and y_u, respectively. Let C_{OPT} be the corresponding configuration for which $C_{OPT}(u) = x^*_u$, $\forall u \in V(G_{\mathcal{S}})$. Let C'_{OPT} be the configuration where $C'_{OPT}(u) = y^*_u$, $\forall u \in V(G_{\mathcal{S}})$. We now establish the correctness of the integer linear program (6).

Theorem 4.21. *The resulting configuration C_{OPT} is a fixed point of $\mathcal{S}$ if and only if $\mathcal{S}$ has at least one fixed point. Further, if $\mathcal{S}$ has no fixed points, then $C_{OPT} \longleftrightarrow C'_{OPT}$ is a 2-cycle of $\mathcal{S}$ where the number of vertices v such that $C_{OPT}(v) = C'_{OPT}(v)$ is maximized.*

Proof. Consider a feasible assignment of variables x_u and y_u, $\forall u \in V(G_{\mathcal{S}})$. Let C and C' be the corresponding configurations of $\mathcal{S}$, where $C(u) = x_u$ and $C'(u) = y_u$, $\forall u \in V(G_{\mathcal{S}})$. We first establish the following claim

Claim 4.21.1. *The configuration C is a successor of C', and C' is also a successor of C.*

Consider the constraint 6b. Observe that $\sum_{v \in N^+(u)} 1 - x_v$ is the number of state-0 vertices in u's closed neighborhood under C. If $y_u = 0$, then clearly constraint 6b is always satisfied. On the other hand, if $y_u = 1$ (i.e., $C(u) = 1$), constraint guarantees that the number of state-0 vertices in u's closed neighborhood is at least $\tau_1(u)$, which aligns with the inverted-thershold dynamic.

We now consider constraint 6c. Observe that $\sum_{v \in N^+(u)} x_v$ is the number of state-1 vertices in u's closed neighborhood under C. If $y_u = 1$, then constraint 6c is trivially satisfied. Alternatively, if $y_u = 0$ (i.e., $C'(u) = 0$), then constraint 6c enforces that the number of state-1 vertices in u's closed neighborhood under C is at least $\tau_0(u)$. Overall, constraint 6b and 6c together ensure that C' is a successor of C. By analogous arguments, it follows that constraint 6d and 6e ensure that C is a successor of C'. This concludes the claim.

Next, we argue that maximizing the objective 6a equivalently maximizes the number of vertices u that satisfies $C(u) = C'(u)$.

Claim 4.21.2. *The objective 6a is maximized if and only if the number of vertices u that satisfies $C(u) = C'(u)$ is maximized.*

Let a_u and b_u be feasible assignments of variables a_u and b_u for $u \in V(G_{\mathcal{S}})$. It is easy to see that constraints 6f and 6g ensures that $a_u \leq \min\{x_u, y_u\}$. Similarly, constraints 6h and 6i ensures that $b_u \geq \max\{x_u, y_u\}$. Observe that when $x_u \neq y_u$ (i.e., $C(u) \neq C'(u)$), we must have $a_u = 0$ and $b_u = 1$. Subsequently, the resulting objective $a^*_u - b^*_u = -1$. Alternatively, the objective for vertex u is the maximum when $a_u - b_u = 0$, that is $a_u = b_u$. Suppose the objective 6a is maximized, it follows that the number of vertices u where $x_u = y_u$ (i.e., $C(u) = C'(u)$) is maximized. Conversely, if the number of vertices u where $x_u = y_u$ is maximized, then clearly the objective is maximized. This concludes the Claim.

Overall, let x^*_u and y^*_u, $\forall u \in V(G_{\mathcal{S}})$, be optimal state assignments for the variable x_u and y_u, respectively. Let C_{OPT} and C'_{OPT} be the corresponding configuration for which $C_{OPT}(u) = x^*_u$, $\forall u \in V(G_{\mathcal{S}})$, and $C'_{OPT}(u) = y^*_u$, $\forall u \in V(G_{\mathcal{S}})$. By Claim 4.21.1, C'_{OPT} is a successor of C_{OPT}, and vice versa. If C_{OPT} is a fixed point of $\mathcal{S}$, then clearly $\mathcal{S}$ has a fixed point. Conversely, if $\mathcal{S}$ has a fixed point, that is, the state of all vertices remains unchanged in the successor of the fixed point. By Claim 4.21.2, it follows that $C_{OPT}(u) = C'_{OPT}(u), \forall u \in V(G)$ and C_{OPT} is a fixed point of $\mathcal{S}$.

Lastly, suppose $\mathcal{S}$ does not have a fixed point. By Claim 4.21.1, $C_{OPT} \longleftrightarrow C'_{OPT}$ is a 2-cycle of $\mathcal{S}$. Further, Claim 4.21.2 implies that the number of vertices u such that $C_{OPT}(u) = C'_{OPT}(u)$ is maximized. This concludes the correctness of the ILP (6). □

5 Convergence Time for Synchronous Anti-coordination Dynamics

In this section, we present the detailed proofs of the convergence result for *synchronous* anti-coordination games, given in section 5 of the main manuscript. Based on the connection between the games and dynamical systems, all proofs are given in the context of *synchronous* dynamical systems.

We first present the convergence result for (SN, IT)-SyDS (modeling SN-SyACG). Then with simple modifications of the proof, the result for (SE, IT)-SyDS (modeling SE-SyACG) follows.

Let $\mathcal{S} = (G_{\mathcal{S}}, \mathcal{F})$ be a (SN, IT)-SyDS that models a SN-SyACG. Recall that for a vertex v, $\tau_0(v)$ is the minimum number of state-1 neighbors of v for f_v to equal 0. W.l.o.g., we assume there are no constant vertices. That is, $1 \leq \tau_1(u), \tau_0(u) \leq d_(u)$ for all $u \in V(G_{\mathcal{S}})$.

The potential functions and bounds

Let C be an arbitrary configuration of $\mathcal{S}$ whose successor is C'. We now define the potentials of vertices, edges, and the system under C. For each vertex $u \in V(G_{\mathcal{S}})$, let $\tilde{\tau}_0(u) = \tau_0(u) - 1/2$.

Vertex potential. Given a vertex $u \in V(G_{\mathcal{S}})$, the potential of u under C, is defined as follows:

$$\mathcal{P}(C, u) = C(u) \cdot \tilde{\tau}_0(u) + C'(u) \cdot \tilde{\tau}_0(u)$$

Edge potentials. Given an edge $e = (u, v) \in E(G_{\mathcal{S}})$, the potential of e under C is defined as follows:

$$\mathcal{P}(C, e) = C(u) \cdot C'(v) + C(v) \cdot C'(u)$$

The configuration potential. The potential of the system $\mathcal{S}$ under C is defined as the subtraction of the sum of vertex potentials from the sum of edge potentials:

$$\mathcal{P}(C, \mathcal{S}) = \sum_{e \in E(G_{\mathcal{S}})} \mathcal{P}(C, e) - \sum_{u \in V(G_{\mathcal{S}})} \mathcal{P}(C, u)$$

A lower bound on the configuration potential. We first present a lower bound on the configuration potential under C.

Lemma 5.1. *The configuration potential satisfies*

$$\mathcal{P}(C, \mathcal{S}) \geq -4m + n$$

Proof. First observe that the sum of edge potentials is non-negative. Further, the sum of potentials of vertices is upper bounded by

$$\begin{aligned}\sum_{u\in V(G_{\mathcal{S}})}\mathcal{P}(C,u) &= \sum_{u\in V(G_{\mathcal{S}})} C(u)\cdot\tilde{\tau}_0(u)+C^{'}(u)\cdot\tilde{\tau}_0(u)\\ &\le 2\sum_{u\in V(G_{\mathcal{S}})}\tilde{\tau}_0(u)\\ &\le 2\sum_{u\in V(G_{\mathcal{S}})}(d(u)-\frac{1}{2})\\ &= 4m-n\end{aligned} \tag{7}$$

The lower bound of the configuration potential immediately follows. □

An upper bound on the configuration potential. We now present an upper bound on the configuration potentials under C.

Lemma 5.2. *The configuration potential $\mathcal{P}(C,\mathcal{S})$ is upper bounded by* 0*.*

Proof. Let $E_u = \{(u,v) : (u,v)\in E(G_{\mathcal{S}})\}$ be the set of edges incident to a vertex $u\in V(G_{\mathcal{S}})$. We can restate the sum of edge potentials as follows

$$\begin{aligned}\sum_{e\in E(G_{\mathcal{S}})}\mathcal{P}(C,e) &= \sum_{(u,v)\in E(G_{\mathcal{S}})}\Big(C(u)\cdot C^{'}(v)+C(v)\cdot C^{'}(u)\Big)\\ &= \sum_{u\in V(G_{\mathcal{S}})}\left(C^{'}(u)\cdot\sum_{(u,v)\in E_u}C(v)\right)\end{aligned} \tag{8}$$

We can further expend the configuration potential into the form

$$\mathcal{P}(C,\mathcal{S}) = \sum_{u\in V(G_{\mathcal{S}})}\left(\left(C^{'}(u)\sum_{(u,v)\in E_u}C(v)\right)-C(u)\cdot\tilde{\tau}_0(u)-C^{'}(u)\cdot\tilde{\tau}_0(u)\right) \tag{9}$$

Note that $\sum_{(u,v)\in E_u}C(v)$ is exactly the number of state-1 neighbors of u under C. If $C^{'}(u)=0$, then

$$\left(C^{'}(u)\sum_{e=(u,v)\in E_u}C(v)\right)-C(u)\cdot\tilde{\tau}_0(u)-C^{'}(u)\cdot\tilde{\tau}_0(u) = -C(u)\cdot\tilde{\tau}_0(u)\le 0 \tag{10}$$

Conversely, if $C^{'}(u)=1$, then by the inverted-threshold dynamics, the number of state-1 neighbor of u under C is less than $\tau_0(u)$. Subsequently, we have

$$\sum_{(u,v)\in E_u}C(v)\le\tau_0(u)-1<\tilde{\tau}_0(u) \tag{11}$$

and

$$\left(C^{'}(u) \sum_{(u,v)\in E_u} C(v)\right) - C^{'}(u)\cdot\tilde{\tau}_0(u) - C(u)\cdot\tilde{\tau}_0(u) \tag{12}$$
$$= \left(\sum_{(u,v)\in E_u} C(v)\right) - \tilde{\tau}_0(u) - C(u)\cdot\tilde{\tau}_0(u)$$
$$\leq 0$$

It follows that the overall configuration potential satisfies $\mathcal{P}(C, \mathcal{S}) \leq 0$. □

Overall, we have established that the gap in the configuration potential value between two configurations is at most $4m - n$.

Decrease of the configuration potential before convergence

In this section, we show that from a configuration C, the configuration potential decrease by at least 1 after every two time-steps, until a fixed point or a 2-cycle is reached.

Lemma 5.3. *Let C an arbitrary configuration of $\mathcal{S}$. We have $\mathcal{P}(C^{'}, \mathcal{S}) = \mathcal{P}(C, \mathcal{S})$ if and only if $C = C^{''}$, that is C is a fixed point or is on a 2-cycle $C \longleftrightarrow C^{'}$. Furthermore, if $C \neq C^{''}$ (i.e., the dynamic has not converged), then $\mathcal{P}(C^{'}, \mathcal{S}) - \mathcal{P}(C, \mathcal{S}) \leq -1/2$.*

Proof. We defined the change of potential of an edge $e = (u, v) \in E(G_{\mathcal{S}})$ from C to $C^{'}$ as

$$\Delta(e) = \mathcal{P}(C^{'}, e) - \mathcal{P}(C, e) \tag{13}$$

and the change of potential of a vertex u as

$$\Delta(u) = \mathcal{P}(C^{'}, u) - \mathcal{P}(C, u) \tag{14}$$

Subsequently, the change of the configuration potential from C to $C^{'}$ is as follows

$$\mathcal{P}(C^{'}, \mathcal{S}) - \mathcal{P}(C, \mathcal{S}) \tag{15}$$
$$= \sum_{e\in E(G_{\mathcal{S}})} \mathcal{P}(C^{'}, e) - \sum_{u\in V(G_{\mathcal{S}})} \mathcal{P}(C^{'}, u) - \sum_{e\in E(G_{\mathcal{S}})} \mathcal{P}(C, e) + \sum_{u\in V(G_{\mathcal{S}})} \mathcal{P}(C, u)$$
$$= \sum_{e\in E(G_{\mathcal{S}})} \Delta(e) - \sum_{u\in V(G_{\mathcal{S}})} \Delta(u)$$

We now expand $\Delta(e)$ for $e = (u, v)$ and $\Delta(u)$ as follows

$$\Delta(e) = \Big(C^{'}(u)\cdot C^{''}(v) + C^{'}(v)\cdot C^{''}(u)\Big) - \Big(C(u)\cdot C^{'}(v) + C^{'}(u)\cdot C(v)\Big) \tag{16}$$
$$= C^{'}(u)\cdot C^{''}(v) + C^{'}(v)\cdot C^{''}(u) - C(u)\cdot C^{'}(v) - C^{'}(u)\cdot C(v)$$
$$= C^{'}(u)\cdot\Big(C^{''}(v) - C(v)\Big) + C^{'}(v)\cdot\Big(C^{''}(u) - C(u)\Big)$$

$$\Delta(u) = \Big(C^{'}(u)\tilde{\tau}_0(u) + C^{''}(u)\tilde{\tau}_0(u)\Big) - \Big(C(u)\tilde{\tau}_0(u) + C^{'}(u)\tilde{\tau}_0(u)\Big) \tag{17}$$
$$= \Big(C^{''}(u) - C(u)\Big)\cdot\tilde{\tau}_0(u)$$

We argue that the change in configuration potential is 0 if and only if $C = C^{''}$.

($\Leftarrow$) Suppose $C = C^{''}$, it follows that $C(u) = C^{''}(u), \forall u \in V(G_{\mathcal{S}})$. Subsequently, $\Delta(e) = 0, \forall e \in E(G_{\mathcal{S}})$ and $\Delta(u) = 0, \forall u \in V(G_{\mathcal{S}})$. Overall, we conclude that the change of potential $\mathcal{P}(C^{'}, \mathcal{S}) - \mathcal{P}(C, \mathcal{S}) = 0$.

($\Rightarrow$) For contrapositive, suppose $C \neq C^{''}$. We now show that the configuration potential decreases by at least $1/2$. It is easy to see that given a vertex u such that $C(u) = C^{''}(u)$, we have $\Delta(u) = 0$. Thus, we consider the set of vertices whose states in C are different from the states in $C^{''}$. Let $V_{0-1} = \{u : C(u) = 0,\ C^{''}(u) = 1,\ u \in V(G_{\mathcal{S}})\}$ be the set of vertices whose states are 0's under C and are 1's under $C^{''}$. Analogously, denoted by $V_{1-0} = \{u : C(u) = 1,\ C^{''}(u) = 0,\ u \in V(G_{\mathcal{S}})\}$ the set of vertices whose states are 1's under C and are 0's under $C^{''}$. We remark that since $C \neq C^{''}$, $|V_{0-1}| + |V_{1-0}| \geq 1$.

We now consider the change of potential for a vertex u whose state in C is different from the state in $C^{''}$, under the following cases.

Case 1: The vertex $u \in V_{0-1}$. Subsequently, the change of vertex potential for u is

$$\Delta(u) = \Big(C^{''}(u) - C(u)\Big) \cdot \tilde{\tau}_0(u) = \tilde{\tau}_0(u) \tag{18}$$

Case 2: The vertex $u \in V_{1-0}$. It then follows that

$$\Delta(u) = \Big(C^{''}(u) - C(u)\Big) \cdot \tilde{\tau}_0(u) = -\tilde{\tau}_0(u) \tag{19}$$

Overall, we have

$$\sum_{u \in V(G_{\mathcal{S}})} \Delta(u) = \sum_{u \in V_{0-1}} \tilde{\tau}_0(u) - \sum_{u \in V_{1-0}} \tilde{\tau}_0(u) \tag{20}$$

As for the change of edge potentials, observe that $\Delta(e) = 0$ for $e = (u, v)$ if $C(u) = C^{''}(u)$ and $C(v) = C^{''}(v)$. We now consider $\Delta(e)$, $e = (u, v)$ where either u or v (or both) undergoes state change from C to $C^{''}$.

Case 1: The state of one vertex altered, and the state of the other remained the same. W.o.l.g., suppose the vertex $u \in V_{0-1}$, and $C(v) = C^{''}(v)$. It follows that

$$\begin{aligned}\Delta(e) &= C^{'}(u) \cdot \Big(C^{''}(v) - C(v)\Big) + C^{'}(v) \cdot \Big(C^{''}(u) - C(u)\Big) \\ &= C^{'}(v)\end{aligned} \tag{21}$$

On the other hand, if $u \in V_{1-0}$, we then have $\Delta(e) = -C^{'}(v)$. Analogously, if the state of v changed and the state of u remained the same, then $\Delta(e) = C^{'}(u)$ when $v \in V_{0-1}$, and $\Delta(e) = -C^{'}(u)$ when $v \in V_{1-0}$.

Case 2: The the states of u, v both changed in the same direction from C to $C^{''}$ (i.e., both from 0 to 1 or from 1 to 0). Specifically, if $u, v \in V_{0-1}$, we then have

$$\begin{aligned}\Delta(e) &= C^{'}(u) \cdot \Big(C^{''}(v) - C(v)\Big) + C^{'}(v) \cdot \Big(C^{''}(u) - C(u)\Big) \\ &= C^{'}(v) + C^{'}(u)\end{aligned} \tag{22}$$

Conversely, if $u, v \in V_{1-0}$, then $\Delta(e) = -C^{'}(v) - C^{'}(u)$.

Case 3: The states of u, v changed in different directions from C to $C^{''}$ (i.e., one from 0 to 1 and the other from 1 to 0). W.l.o.g., suppose $u \in V_{0-1}$ and $v \in V_{1-0}$. Subsequently, we have

$$\Delta(e) = C^{'}(v) - C^{'}(u) \tag{23}$$

Similarly, if $u \in V_{1-0}$ and $v \in V_{0-1}$, then $\Delta(e) = C^{'}(u) - C^{'}(v)$

Overall, observe that for any vertex $u \in V_{0-1}$, the change of edge potential $\Delta(e)$ for each incident edge $e = (u, v)$ has a positive term $C^{'}(v)$. Conversely, if $u \in V_{1-0}$, then $\Delta(e)$ has a negative term $-C^{'}(v)$. Let $E_u = \{(u, v) : (u, v) \in E(G_{\mathcal{S}})\}$ be the set of edges incident to $u \in V(G_{\mathcal{S}})$, we can rewire the sum of the change in edge potentials as

$$\sum_{e \in E(G_{\mathcal{S}})} \Delta(e) = \left(\sum_{u \in V_{0-1}} \sum_{(u,v) \in E_u} C^{'}(v) \right) - \left(\sum_{u \in V_{1-0}} \sum_{(u,v) \in E_u} C^{'}(v) \right) \tag{24}$$

In combined with Equation 20, we characterize the change in the configuration potential from C to $C^{'}$ as follows

$$\begin{aligned} &\mathcal{P}(C^{'}, \mathcal{S}) - \mathcal{P}(C, \mathcal{S}) \\ &= \sum_{e \in E(G_{\mathcal{S}})} \Delta(e) - \sum_{u \in V(G_{\mathcal{S}})} \Delta(u) \\ &= \left(\sum_{u \in V_{0-1}} \sum_{(u,v) \in E_u} C^{'}(v) - \sum_{u \in V_{1-0}} \sum_{(u,v) \in E_u} C^{'}(v) \right) - \left(\sum_{u \in V_{0-1}} \tilde{\tau}_0(u) - \sum_{u \in V_{1-0}} \tilde{\tau}_0(u) \right) \\ &= \left(\sum_{u \in V_{0-1}} \left(\sum_{(u,v) \in E_u} C^{'}(v) \right) - \tilde{\tau}_0(u) \right) + \left(\sum_{u \in V_{1-0}} \tilde{\tau}_0(u) - \left(\sum_{(u,v) \in E_u} C^{'}(v) \right) \right) \end{aligned} \tag{25}$$

Remark that the term $\sum_{(u,v) \in E_u} C^{'}(v)$ is the number of state-1 neighbors of a vertex u under $C^{'}$. Suppose $u \in V_{0-1}$, that is, $C^{''}(u) = 1$. By the dynamics of inverted-threshold model, it follows that the number of state-1 neighbors of u under $C^{'}$ is less than $\tau_0(u) = \tilde{\tau}_0(u) + 1/2$, thus,

$$\sum_{(u,v) \in E_u} C^{'}(v) \leq \tau_0(u) - 1 < \tilde{\tau}_0(u)$$

and

$$\left(\sum_{(u,v) \in E_u} C^{'}(v) \right) - \tilde{\tau}_0(u) \leq -\frac{1}{2}$$

Conversely, if $u \in V_{1-0}$ which implies that $C^{''}(u) = 0$. It follows that the number of state-1 neighbors of u is at least $\tau_0(u)$. Thus,

$$\sum_{(u,v) \in E_u} C^{'}(v) \geq \tau_0(u) > \tilde{\tau}_0(u)$$

and

$$\tilde{\tau}_0(u) - \sum_{(u,v) \in E_u} C^{'}(v) \leq -\frac{1}{2}$$

Overall, we have

$$
\begin{aligned}
&\mathcal{P}(C^{'}, \mathcal{S}) - \mathcal{P}(C, \mathcal{S}) \qquad (26) \\
&= \left(\sum_{u \in V_{0-1}} \left(\sum_{(u,v) \in E_u} C^{'}(v) \right) - \tilde{\tau}_0(u) \right) + \left(\sum_{u \in V_{1-0}} \tilde{\tau}_0(u) - \left(\sum_{(u,v) \in E_u} C^{'}(v) \right) \right) \\
&\leq - \sum_{u \in V_{0-1}} \frac{1}{2} - \sum_{u \in V_{1-0}} \frac{1}{2} \\
&= - \frac{|V_{0-1}| + |V_{1-0}|}{2} \\
&\leq -\frac{1}{2}
\end{aligned}
$$

This concludes the proof. □

Overall, we have shown that the potential gap between any two configurations is $4m - n$. Furthermore, the overall system configuration decreases by at least $1/2$ after each time step. It follows that for a (SN, IT)-SyDS, starting from an arbitrary initial configuration, the system dynamic converges in at most $8m - 2n$ time steps, irrespective of the underlying network structure. Subsequently, the convergence time result for SN-SyACG follows.

Theorem 5.4. *For SN-SyACG, starting from any initial action profile, the best-response dynamic converges to a Nash equilibrium or a 2-cycle in $O(m)$ time steps.*

With simple modifications of Lemma 5.3 (i.e., consider u as a neighbor of itself), we can also show that the same convergence time can be extended to SE-SyACG.

Theorem 5.5. *For a SE-SyACG, starting from any initial action profile, the best-response dynamic converges to a Nash equilibrium or a 2-cycle in $O(m)$ time steps.*

Corollary 5.6. *For both SE-SyACG and SN-SyACG, starting from any initial action profile, the best-response dynamic converges to a Nash equilibrium or a 2-cycle in $O(n)$ time steps if the graph is degree bounded.*