Implementing the Adaptation Procedure in Misinformation Games

Merkouris Papamichail, Constantinos Varsos, Giorgos Flouris

Information Systems Laboratory, Institute of Computer Science
Foundation for Research and Technology-Hellas

SETN 2022, Corfu 7-9 September

INSTITUTE OF COMPUTER SCIENCE

Table of Contents

FORTH INSTITUTE OF COMPUTER SCIENCE

Table of Contents

Classic Game Theory: Normal Form Games

1. We have n agents (players).
2. Each player has a finite number of strategies $S_i, i \in [n]$.
3. The players can play:
   - either one of their strategies deterministically (*pure strategy*),
   - or chose randomly from a subset of strategies (*mixed strategy*).
4. With $\Sigma_i, i \in [n]$ we denote the set of mixed strategies.
5. The game is played in a *single turn*.
6. Each player chooses a mixed strategy *simultaneously*.
7. The players are getting payed from *mutually known* n-dimensional *payoff matrices* $P^i \in \mathbb{R}^{|S_i| \times |S_2| \times \cdots \times |S_n|}, i \in [n]$.
8. A strategy profile $\sigma \in \Sigma_1 \times \cdots \times \Sigma_n$ characterises a strategic state.
9. The behaviour of the players is predicted by the Nash Equilibria (NE).
10. **Theorem:** A Nash Equilibrium *always exists* [1].

What happens if the players are *misinformed* about their expected payoffs?

Misinformation Games

A (canonical) *Misinformation Game* [2] (MG) is:

1. A tuple of $n+1$ NFGs, i.e. $mG = \langle G^0, G^1, \ldots, G^n \rangle$.
2. G^0 is called *objective* or *actual* game.
3. We call games $G^i, i \in [n]$ *subjective*.
4. Each player, in each game has *the same strategy set*.
5. All G^is differ *only* in the payoff matrices P^i.
6. Player i *only* knows about her *subjective* payoff matrix P^i.

FORTH INSTITUTE OF COMPUTER SCIENCE

Predicting the Behaviour of the Misinformed Players

- Consider a strategy profile $\sigma = (\sigma_1, \ldots, \sigma_n)$, where for each player i:
  - There is a Nash Equilibrium τ of the *subjective game* G^i.
  - Player i plays σ_i in the NE τ, i.e. $\tau = (\sigma_i, \sigma_{-i})$.
- We call σ a *Natural Misinformed Equilibrium* (NME).
- We assume that NMEs *predict* the behaviour of the players under *misinformation*.

Corollary: *Every Misinformation game has a Natural Misinformed Equilibrium.*

Running Example

Consider the Misinformation Game $mG = \langle G^0, G^1, G^2 \rangle$ with payoff matrices

$$P^0 = \begin{pmatrix} (5,1) & (3,1) \\ (4,5) & (1,0) \end{pmatrix}, \quad P^1 = \begin{pmatrix} (3,6) & (0,6) \\ (3,4) & (5,3) \end{pmatrix}, \quad P^2 = \begin{pmatrix} (4,9) & (3,1) \\ (5,3) & (1,7) \end{pmatrix}$$

Running Example

Consider the Misinformation Game $mG = \langle G^0, G^1, G^2 \rangle$ with payoff matrices

$$P^0 = \begin{pmatrix} (5,1) & (3,1) \\ (4,5) & (1,0) \end{pmatrix}, \quad P^1 = \begin{pmatrix} (3,6) & (0,6) \\ (3,4) & (5,3) \end{pmatrix}, \quad P^2 = \begin{pmatrix} (4,9) & (3,1) \\ (5,3) & (1,7) \end{pmatrix}$$

The Nash equilibria from G^1 and G^2 are

$$NE(G^1) = \{((1,0),(1,0)),((0,1),(1,0))\}, \quad NE(G^2) = \{((1/3,2/3),(2/3,1/3))\}$$

Running Example

Consider the Misinformation Game $mG = \langle G^0, G^1, G^2 \rangle$ with payoff matrices

$$P^0 = \begin{pmatrix} (5,1) & (3,1) \\ (4,5) & (1,0) \end{pmatrix}, \quad P^1 = \begin{pmatrix} (3,6) & (0,6) \\ (3,4) & (5,3) \end{pmatrix}, \quad P^2 = \begin{pmatrix} (4,9) & (3,1) \\ (5,3) & (1,7) \end{pmatrix}$$

The Nash equilibria from G^1 and G^2 are

$$NE(G^1) = \{((1,0),(1,0)),((0,1),(1,0))\}, \quad NE(G^2) = \{((1/3,2/3),(2/3,1/3))\}$$

The nmes are

$$\Sigma = NME(mG) = \{((1,0),(2/3,1/3)),((0,1),(2/3,1/3))\}$$

FORTH
INSTITUTE OF COMPUTER SCIENCE

Table of Contents

FORTH
INSTITUTE OF COMPUTER SCIENCE

Adaptation Procedure

- We now let the agents *adapt* and *reconsider* from their payoffs.

Consider a game played in *multiple turns*:

1. We start from a *root* Misinformation Game.
2. In each turn the players choose and *play a natural misinformed equilibrium*.
3. The players get paid from the *actual game* (but base their strategies on their respective *subjective games*).
4. After receiving their payments, the players *update* their subjective views.
5. We call this process *Adaptation Procedure*.

INSTITUTE OF COMPUTER SCIENCE

Position Vectors & Update Operation

Position Vectors:

- Consider a NME σ.
- $\chi(\sigma)$ denotes the positions of the strategies, played with *positive probability*.
- We call $\vec{v} \in \chi(\sigma)$ *position vectors*.

Update Operation:

1. Let a position vector $\vec{v}$.
2. Let $u = P^0(\vec{v})$ be the *objective* payment of the players.
3. We *update* the *subjective* payoff matrices of the players P^i, i.e. $P^i(\vec{v}) \leftarrow u$.
4. We denote the *resulting* Misinformation Game with $mG_{\vec{v}}$.

FORTH INSTITUTE OF COMPUTER SCIENCE

Running Example

Consider the Misinformation Game $mG = \langle G^0, G^1, G^2 \rangle$ with payoff matrices

$$P^0 = \begin{pmatrix} (5,1) & (3,1) \\ (4,5) & (1,0) \end{pmatrix}, \quad P^1 = \begin{pmatrix} (3,6) & (0,6) \\ (3,4) & (5,3) \end{pmatrix}, \quad P^2 = \begin{pmatrix} (4,9) & (3,1) \\ (5,3) & (1,7) \end{pmatrix}$$

The Nash equilibria from G^1 and G^2 are

$$NE(G^1) = \{((1,0),(1,0)),((0,1),(1,0))\}, \quad NE(G^2) = \{((1/3,2/3),(2/3,1/3))\}$$

The nmes are $\Sigma = NME(mG) = \{((1,0),(2/3,1/3)),((0,1),(2/3,1/3))\}$ and the characteristic is $\chi(\Sigma) = \{(1,1),(1,2),(2,1),(2,2)\}$

FORTH
INSTITUTE OF COMPUTER SCIENCE

Running Example

Consider the Misinformation Game $mG = \langle G^0, G^1, G^2 \rangle$ with payoff matrices

$$P^0 = \begin{pmatrix} (5,1) & (3,1) \\ (4,5) & (1,0) \end{pmatrix}, \quad P^1 = \begin{pmatrix} (3,6) & (0,6) \\ (3,4) & (5,3) \end{pmatrix}, \quad P^2 = \begin{pmatrix} (4,9) & (3,1) \\ (5,3) & (1,7) \end{pmatrix}$$

The Nash equilibria from G^1 and G^2 are

$$NE(G^1) = \{((1,0),(1,0)),((0,1),(1,0))\}, \quad NE(G^2) = \{((1/3,2/3),(2/3,1/3))\}$$

The nmes are $\Sigma = NME(mG) = \{((1,0),(2/3,1/3)),((0,1),(2/3,1/3))\}$ and the characteristic is $\chi(\Sigma) = \{(1,1),(1,2),(2,1),(2,2)\}$
Update at $\vec{v} = (1,1)$

$$P^0 = \begin{pmatrix} (5,1) & (3,1) \\ (4,5) & (1,0) \end{pmatrix}, \quad P^1 = \begin{pmatrix} (5,1) & (0,6) \\ (3,4) & (5,3) \end{pmatrix}, \quad P^2 = \begin{pmatrix} (5,1) & (3,1) \\ (5,3) & (1,7) \end{pmatrix}$$

INSTITUTE OF COMPUTER SCIENCE

Adaptation Procedure

We define the Adaptation Procedure recursively:

$$\mathcal{AD}^t(M) = \{mG_{\vec{u}} \mid mG \in \mathcal{AD}^{t-1}(M), \vec{u} \in \chi(\sigma), \sigma \in NME(mG)\}$$

- **End Criterion:** The Adaptation Procedure concludes when, $\mathcal{AD}^{\ell+1}(M) = \mathcal{AD}^{\ell}(M)$, for some $\ell < \infty$.
- We call *length* the *smallest* $\ell \in \mathbb{N}$, s.t. *the end criterion holds*. i.e. $\mathcal{AD}^{\ell+1}(S) = \mathcal{AD}^{\ell}(S)$.
- **Theorem:** *For ever finite mG, the procedure terminates after a finite number of steps.*

Predicting the Points of Stability

- Let *mG* be the *root* of the Adaptation Procedure.
- Let ℓ be the *length* of the Adaptation Procedure.
- Let $\widehat{mG} \in \mathcal{AD}^{\ell}(\{mG\})$.
- Also let σ be a NME of $\widehat{mG}$, such that:
  For every position vector $\vec{v} \in \chi(\sigma)$, $\widehat{mG}_{\vec{v}} = \widehat{mG}$.
- We call σ a *Stable Misinformed Equilibrium* (SME) of the Adaption Procedure on mG.

Theorem: *Every Misinformation Game has a Stable Misinformed Equilibrium.*

SME Example

Consider *another* Misinformation Game $mG^\star$:

$$P^0 = \begin{pmatrix} (5,1) & (3,1) \\ (4,5) & (1,0) \end{pmatrix}, \; P^1 = \begin{pmatrix} (5,1) & (0,6) \\ (4,5) & (5,3) \end{pmatrix}, \; P^2 = \begin{pmatrix} (5,1) & (3,1) \\ (4,5) & (1,7) \end{pmatrix}$$

We have the NMEs:

- $\sigma_1 = ((0.28, 0.72), (0, 1)),\ \chi(\sigma_1) = \{(1, 2), (2, 2)\}$
- $\sigma_2 = ((0.28, 0.72), (1, 0)),\ \chi(\sigma_2) = \{(1, 1), (2, 1)\}$

Hence, σ_2 is a *Stable Misinformed Equilibrium*.

Table of Contents

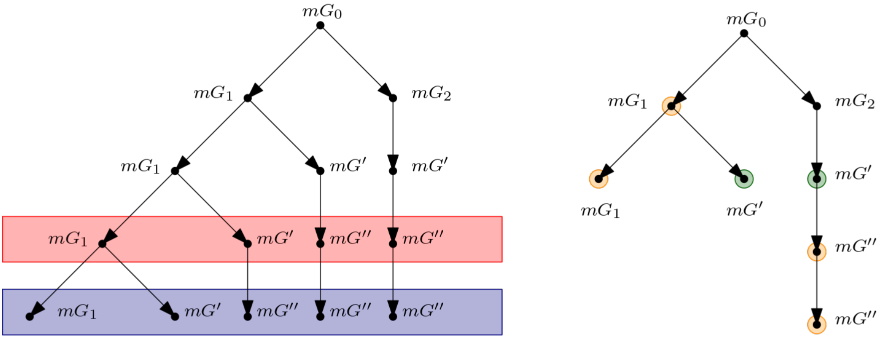
FORTH INSTITUTE OF COMPUTER SCIENCE

The Problem

The Problem

Input: A *root* Misinformation Game mG_0.

Output: *All* the SMEs of mG_0.



(a) Following *naively* the recursive definition.
Number of Nodes: 15.

(b) A more *"economic"* approach.
Number of Nodes: 8

Figure 1: The naive approach is inefficient.

Adaptation Graph

Adaptation Graph

Let a *directed graph* $D = (V, A)$, s.t.

- $V = \cup_{\ell=1}^{\infty} \mathcal{AD}^{\ell}(\{mG_0\})$,
- and $(mG_1, mG_2) \in A$ iff $mG_2 \in \mathcal{AD}^1(mG_1)$.

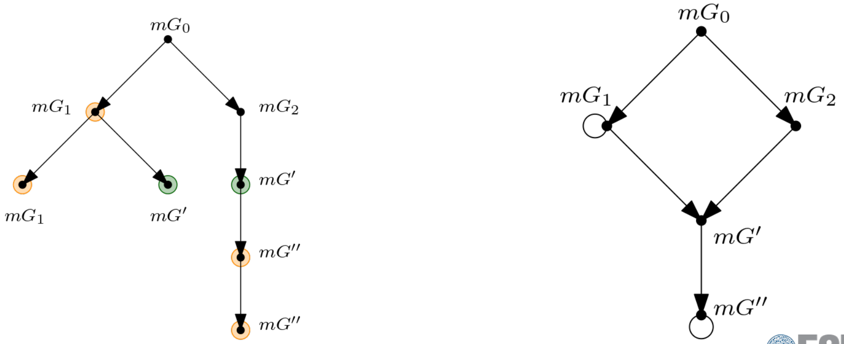


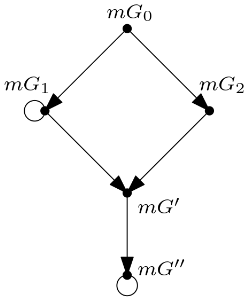
Figure 2: The "minimal" tree and the corresponding graph.

Adaptation Graph: Loops, Diamonds and Terminal Nodes

The Adaptation Graph is *almost a DAG*.

1. We *can't have* directed circles (the agents don't forget).
2. There are *loops*. The agents are stabilising in this MG.
3. There are *diamonds*. The agents may arrive to the same MG following different routes.

Terminal Nodes: We call the nodes that have *at least* one self-loop terminal.



Terminal Node Example

Consider *another* Misinformation Game $mG^{\star\star}$:

$$P^0 = \begin{pmatrix} (5,1) & (3,1) \\ (4,5) & (1,0) \end{pmatrix},\ P^1 = \begin{pmatrix} (5,1) & (0,6) \\ (3,4) & (1,0) \end{pmatrix},\ P^2 = \begin{pmatrix} (5,1) & (3,1) \\ (5,3) & (1,0) \end{pmatrix}$$

We have the NMEs:

- $\sigma_1 = ((0.44, 0.56), (1, 0)),\ \chi(\sigma_1) = \{(1,1), (2,1)\}$
- $\sigma_2 = ((0.44, 0.56), (0, 1)),\ \chi(\sigma_2) = \{(1,2), (2,2)\}$

For $\vec{v} = (1,1)$, we have $(mG^{\star\star})_{\vec{v}} = mG^{\star\star}$. Hence, $mG^{\star\star}$ is *Terminal Node*. *None* of σ_1, σ_2 is an SME. With L we denote the *Terminal Set*

INSTITUTE OF COMPUTER SCIENCE

- The Terminal Set is a *weaker* notion of stability.
- For some SME σ, if $\sigma \in \mathrm{NME}(mG)$, then $mG \in L$.
- The oposite *does not* hold.

From the above we have a strategy for computing the SMEs efficiently:

1. Compute the terminal set L.
2. From the terminal set, compute the SMEs.

The Algorithm

Traverse Adaptation Graph

1. $L \leftarrow \varnothing$ `// Terminal Set`
2. $V \leftarrow \varnothing$ `// Visited`
3. $Q \leftarrow \{mG_0\}$ `// Queue`
4. **while** $Q \neq \varnothing$ **:**
   1. $mG \leftarrow \mathrm{pop}(Q)$
   2. **for each** position vector $\vec{v} \in \chi(\mathrm{NME}(mG))$ **:**
      1. $mG' \leftarrow (mG)_{\vec{v}}$
      2. **if** $mG' = mG$ **:** $L \leftarrow L \cup \{mG'\}$
      3. **if** $mG' \notin V$ **:** $V \leftarrow V \cup \{mG'\}$, $Q \leftarrow Q \cup \{mG'\}$
5. **return** L

Compute SMEs

1. $\mathrm{SME} \leftarrow \varnothing$
2. **for each** $mG \in L$**:**
   1. **for each** $\sigma \in \mathrm{NME}(mG)$
      1. **if** *for all position vectors* $\vec{v} \in$ $\chi(\sigma)mG = (mG)_{\vec{v}}$**:** $\mathrm{SME} \leftarrow \mathrm{SME} \cup \{\sigma\}$
3. **return** SME

FORTH
INSTITUTE OF COMPUTER SCIENCE

Table of Contents

FORTH INSTITUTE OF COMPUTER SCIENCE

Conclusions & Future Work

Conclusions

- We considered a model of non-cooperative games with misinformation.
- We presented an adaptation procedure in the above setting.
- We gave the first non-trivial algorithm for computing the stabilising points of the procedure.

Future Work

- Can the algorithm be *fully* parallelized?
- What is the complexity of computing a single SME?
- Can *approximate* techniques for finding NE improve the complexity?

Thank you for your time!

Thank you for your time!

I'll be happy to answer your questions :)

Table of Contents

FORTH INSTITUTE OF COMPUTER SCIENCE

Efficiency Experiments

	no. strategy profiles	slow mode		fast mode	
		total time	CPU time	total time	CPU time
2-pl.	2×2	0.170s	0.050s	0.151s	0.045s
	3×2	0.439s	0.090s	0.437s	0.090s
	3×3	0.883s	0.134s	1.129s	0.154s
	4×3	68.81s	12.658s	31.373s	3.791s
	4×4	–	–	3634.635s	454.232s
3-pl.	$2 \times 2 \times 2$	–	–	26.963s	0.563s

Properties of the Adaptation Procedure

mode	no. strategy profiles	no. nodes	no. unique mGs	no. leafs	no. (unique mGs $\wedge$ leafs)	no. *smes*
slow	2×2	13.667	4.000	8.667	2.333	4.667
	3×2	20.333	7.333	11.667	6.000	2.000
	3×3	42.333	9.000	29.333	7.667	1.333
	4×3	9008.0	181.667	4694.00	144.00	4.000
fast	2×2	9.667	4.000	5.000	2.333	4.667
	3×2	17.667	7.333	9.333	6.000	2.000
	3×3	28.000	9.000	17.333	7.667	1.333
	4×3	543.00	180.00	191.667	143.00	4.000
	4×4	46285.333	9950.333	26830.667	9392.333	1584.667

Update Operation: Logic Program

Update Operation

1. `v(G, P, SP, V) :- pos(SP), u(0, P, SP, U), V = U.`
   `% the new payoff function if pos(SP).`
2. `v(G, P, SP, V) :- not pos(SP), u(G, P, SP, U), V = U.`
   `% the new payoff function if not pos(SP)`
3. `changed :- v(G, P, SP, V), u(G, P, SP, U), V != U.`
   `unchanged :- not changed.`
   `% check if child-mG != parent-mG`

MG-Pool Data Structure

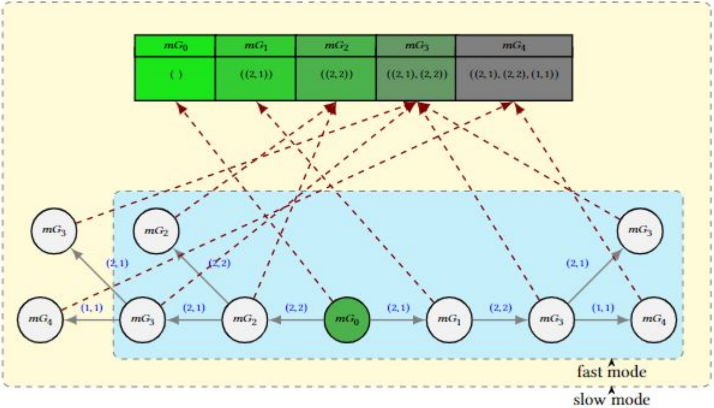


Figure 3: The `MG-Pool` Data Structure.

Bibliography

J. F. Nash, *Non-cooperative games*, The Annals of Mathematics, 54 (1951), pp. 286–295.

C. Varsos, G. Flouris, M. Bitsaki, and M. Fasoulakis, *A study of misinformation games*, in PRICAI 2021, vol. 13031, 2021, pp. 76–87.