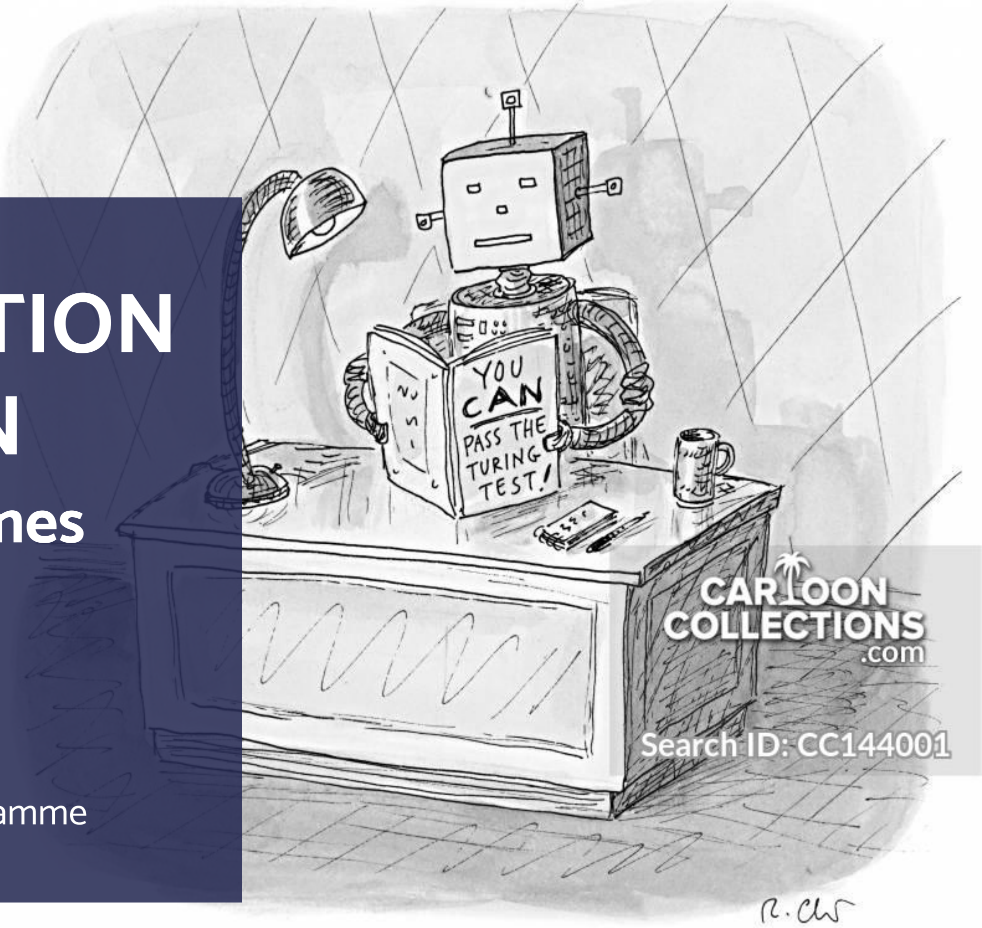
PUBLIC INFORMATION REPRESENTATION
for Adversarial Team Games

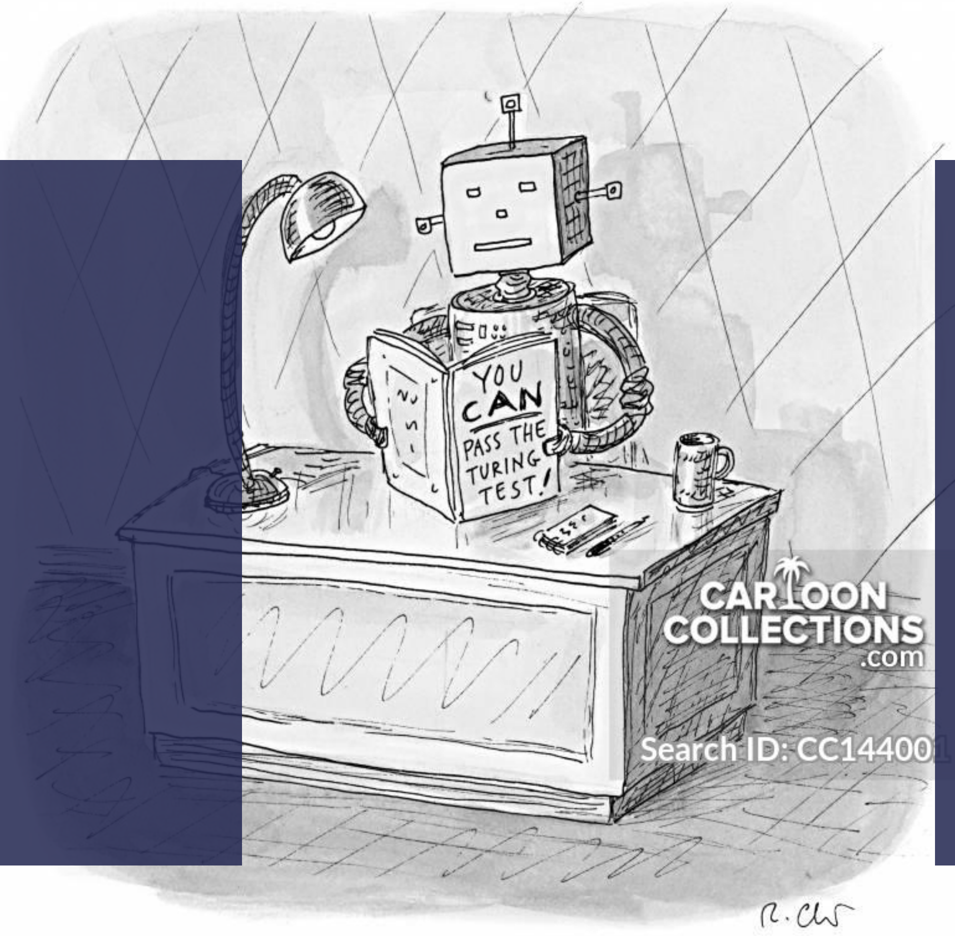
Luca Carminati, MSc student, Honours Programme
luca5.carminati@mail.polimi.it



HONOURS PROGRAMME HP-SR in Information Technology

POLITECNICO MILANO 1863

Introduction to Games



FOCUS OF OUR WORK

Multiagent games with mixed cooperative–competitive structure

Multiple agents organized in teams act sequentially on the environment with the goal of maximizing a payoff.
We focus on the **N vs 1** games, also called **adversarial team games**.

FOCUS OF OUR WORK

Multiagent games with mixed cooperative–competitive structure

Multiple agents organized in teams act sequentially on the environment with the goal of maximizing a payoff.
We focus on the **N vs 1** games, also called **adversarial team games**.

Increasing research interest

10 articles published on AAAI, IJCAI, ICML, NEURIPS since 2017.
Cooperative AI as research focus after the successes of *AlphaGo* and *Libratus* (Dafoe et al., 2020).

FOCUS OF OUR WORK

Multiagent games with mixed cooperative–competitive structure

Multiple agents organized in teams act sequentially on the environment with the goal of maximizing a payoff.
We focus on the **N vs 1** games, also called **adversarial team games**.

Increasing research interest

10 articles published on AAAI, IJCAI, ICML, NEURIPS since 2017.
Cooperative AI as research focus after the successes of *AlphaGo* and *Libratus* (Dafoe et al., 2020).

Applications

GAME THEORY

1.Introduction

GAME THEORY

We adopt the **mathematical approach** defined by ***Algorithmic Game Theory.***

This allows to formally represent:

- **Games,** that are *sequential interactions of agents*
- **Strategies** of the players, that are *functions associating an action to each decision point*
- **Solution concepts,** that are *rational equilibria for player strategies*

GAME THEORY

We adopt the **mathematical approach** defined by ***Algorithmic Game Theory.***

This allows to formally represent:

- **Games,** that are *sequential interactions of agents*
- **Strategies** of the players, that are *functions associating an action to each decision point*
- **Solution concepts,** that are *rational equilibria for player strategies*

Available formalisms for game representation:

GAME THEORY

We adopt the **mathematical approach** defined by ***Algorithmic Game Theory.***

This allows to formally represent:

- **Games,** that are *sequential interactions of agents*
- **Strategies** of the players, that are *functions associating an action to each decision point*
- **Solution concepts,** that are *rational equilibria for player strategies*

Available formalisms for game representation:

Extensive form representation

The game is represented as a **tree** of **decision nodes**. At each node one of the players chooses an action to be played.

GAME THEORY

We adopt the **mathematical approach** defined by ***Algorithmic Game Theory.***

This allows to formally represent:

- **Games,** that are *sequential interactions of agents*
- **Strategies** of the players, that are *functions associating an action to each decision point*
- **Solution concepts,** that are *rational equilibria for player strategies*

Available formalisms for game representation:

Extensive form representation

The game is represented as a **tree** of **decision nodes**. At each node one of the players chooses an action to be played.

Normal form representation

The game is represented as a **matrix**. Each player decides exactly one action at the start of the game, without knowing the actions chosen by the others

GAME THEORY

We adopt the **mathematical approach** defined by ***Algorithmic Game Theory.***

This allows to formally represent:

- **Games,** that are *sequential interactions of agents*
- **Strategies** of the players, that are *functions associating an action to each decision point*
- **Solution concepts,** that are *rational equilibria for player strategies*

Available formalisms for game representation:

Extensive form representation

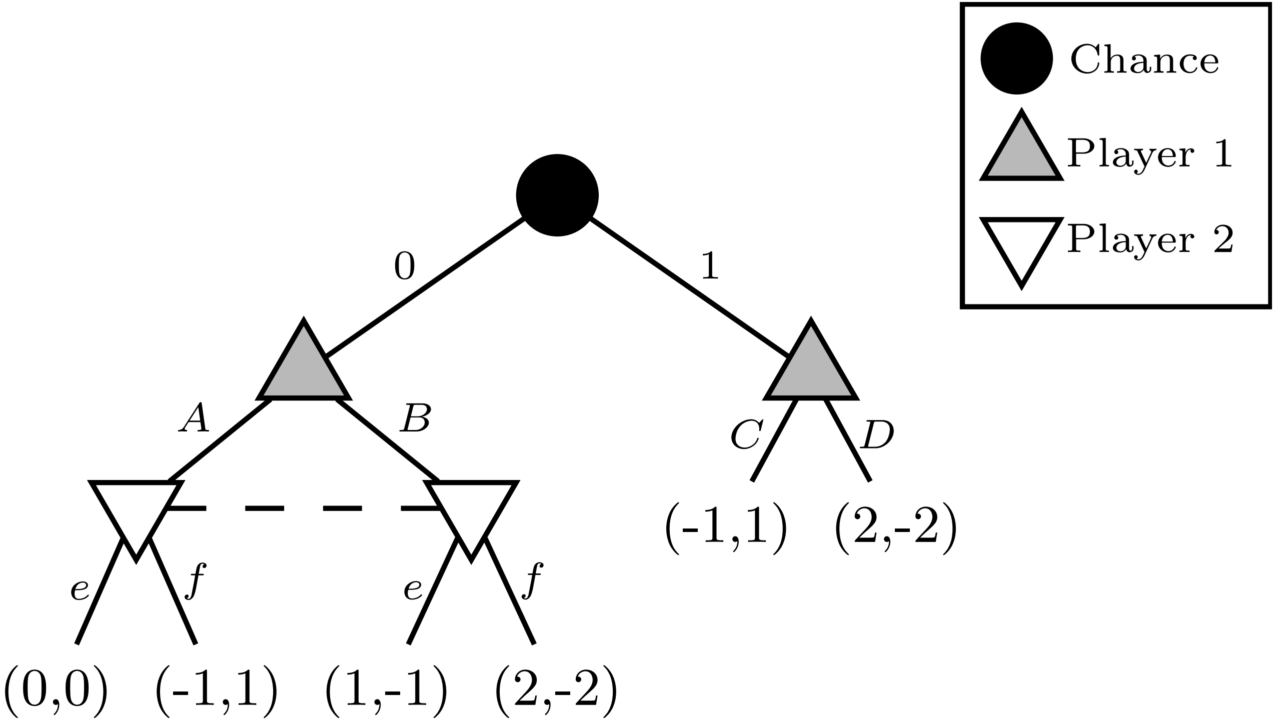
The game is represented as a **tree** of **decision nodes**. At each node one of the players chooses an action to be played.

More efficient and semantic representation for sequential interactions

Normal form representation

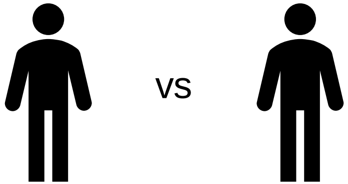
The game is represented as a **matrix**. Each player decides exactly one action at the start of the game, without knowing the actions chosen by the others

EXAMPLE OF A GAME



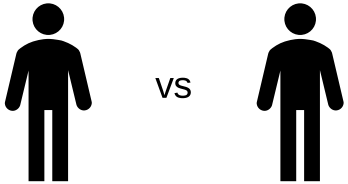
TWO-PLAYERS ZERO-SUM GAMES

Two-players Zero-sum games (2p0s) is a simple class of games, in which **any gain for one player corresponds to a loss of the other one**, expressed as opposite payoffs for each terminal node.



TWO-PLAYERS ZERO-SUM GAMES

Two-players Zero-sum games (2p0s) is a simple class of games, in which **any gain for one player corresponds to a loss of the other one**, expressed as opposite payoffs for each terminal node.

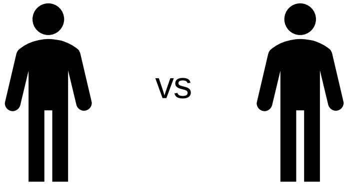


The game presented in the previous slide is 2p0s

TWO-PLAYERS ZERO-SUM GAMES

Two-players Zero-sum games (2p0s) is a simple class of games, in which **any gain for one player corresponds to a loss of the other one**, expressed as opposite payoffs for each terminal node.

Addressed by research in game theory since the seminal work of (Nash, **1951**)



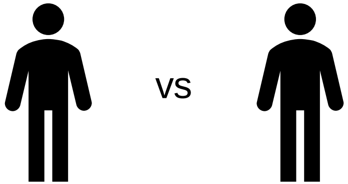
The game presented in the previous slide is 2p0s

TWO-PLAYERS ZERO-SUM GAMES

Two-players Zero-sum games (2p0s) is a simple class of games, in which **any gain for one player corresponds to a loss of the other one**, expressed as opposite payoffs for each terminal node.

Addressed by research in game theory since the seminal work of (Nash, **1951**)

The typical solution concept used in the context of 2p0s games is called **Nash Equilibrium.**

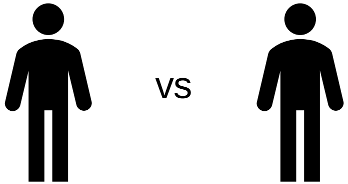


The game presented in the previous slide is 2p0s

TWO-PLAYERS ZERO-SUM GAMES

Two-players Zero-sum games (2p0s) is a simple class of games, in which **any gain for one player corresponds to a loss of the other one**, expressed as opposite payoffs for each terminal node.

Addressed by research in game theory since the seminal work of (Nash, **1951**)



The typical solution concept used in the context of 2p0s games is called **Nash Equilibrium.**

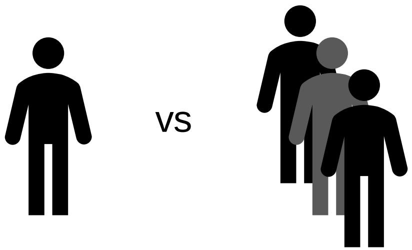
The game presented in the previous slide is 2p0s

Properties:

- Finding a Nash Equilibrium is a **P-complete** problem

ADVERSARIAL TEAM GAMES

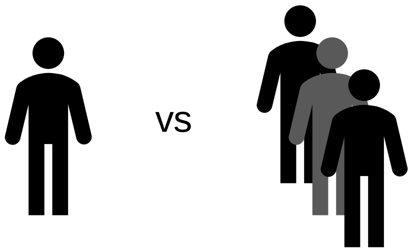
Adversarial Team Games are a generalization of 2p0s games. Two or more players share the same payoff function which is opposite to the one of a single opponent.



ADVERSARIAL TEAM GAMES

Adversarial Team Games are a generalization of 2p0s games. Two or more players share the same payoff function which is opposite to the one of a single opponent.

Addressed by research in game theory since the seminal work of (Von Stengel, 1997)

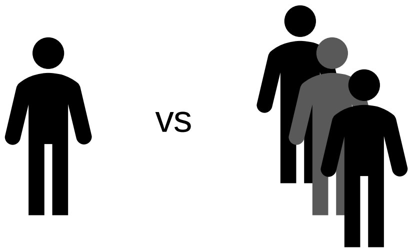


ADVERSARIAL TEAM GAMES

Adversarial Team Games are a generalization of 2p0s games. Two or more players share the same payoff function which is opposite to the one of a single opponent.

Addressed by research in game theory since the seminal work of (Von Stengel, 1997)

In this context, we consider *ex ante correlation,* that allows players to coordinate the choice of a strategy before the start of a game.



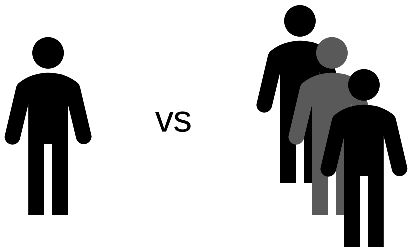
ADVERSARIAL TEAM GAMES

Adversarial Team Games are a generalization of 2p0s games. Two or more players share the same payoff function which is opposite to the one of a single opponent.

Addressed by research in game theory since the seminal work of (Von Stengel, 1997)

In this context, we consider *ex ante correlation,* that allows players to coordinate the choice of a strategy before the start of a game.

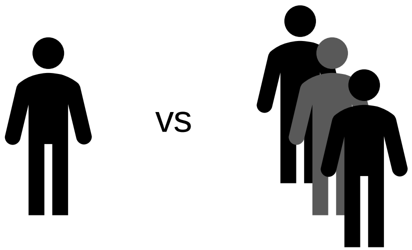
The solution concept used in this context is called **Team Maxmin with correlation (TMECor).**



ADVERSARIAL TEAM GAMES

Adversarial Team Games are a generalization of 2p0s games. Two or more players share the same payoff function which is opposite to the one of a single opponent.

Addressed by research in game theory since the seminal work of (Von Stengel, 1997)



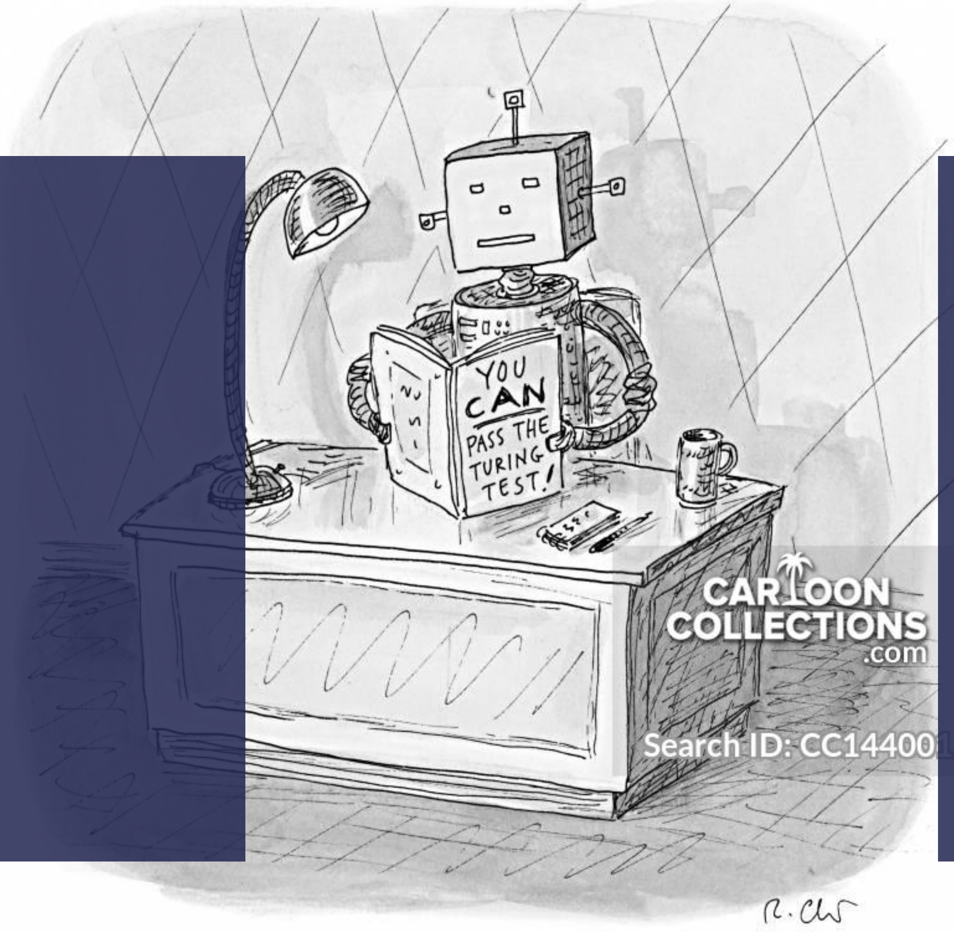
In this context, we consider *ex ante correlation,* that allows players to coordinate the choice of a strategy before the start of a game.

The solution concept used in this context is called **Team Maxmin with correlation (TMECor).**

Properties:

- Finding a TMECor is a **NP-hard** problem (Celli and Gatti, 2018)

Motivations of our work



SOLVING 2P0S GAMES

Approaches to the solution of 2p0s games

SOLVING 2P0S GAMES

Approaches to the solution of 2p0s games

1. **Linear Programming approaches**
   Describe a Nash Equilibrium in the game as a Linear Program

SOLVING 2P0S GAMES

Approaches to the solution of 2p0s games

1. **Linear Programming approaches**
   Describe a Nash Equilibrium in the game as a Linear Program

Not scalable to large games

SOLVING 2P0S GAMES

Approaches to the solution of 2p0s games

1. **Linear Programming approaches**
   Describe a Nash Equilibrium in the game as a Linear Program

Not scalable to large games

2. **Abstraction + Continual Resolving**
   Use a coarse representation of the original game to obtain a approximatively good strategy on whole game; then at runtime solve a depth limited version of the game considering players playing the approximated equilibrium after the limit

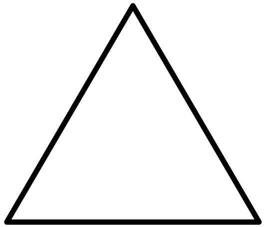
SOLVING 2P0S GAMES

Approaches to the solution of 2p0s games

1. **Linear Programming approaches**
   Describe a Nash Equilibrium in the game as a Linear Program

Not scalable to large games

2. **Abstraction + Continual Resolving**
   Use a coarse representation of the original game to obtain a approximatively good strategy on whole game; then at runtime solve a depth limited version of the game considering players playing the approximated equilibrium after the limit



Original Game

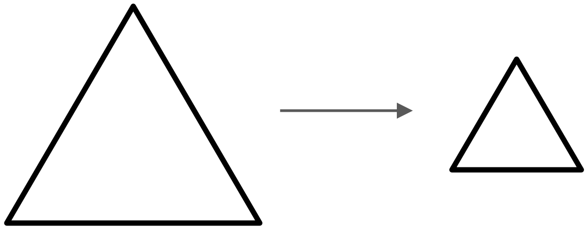
SOLVING 2P0S GAMES

Approaches to the solution of 2p0s games

1. **Linear Programming approaches**
   Describe a Nash Equilibrium in the game as a Linear Program

Not scalable to large games

2. **Abstraction + Continual Resolving**
   Use a coarse representation of the original game to obtain a approximatively good strategy on whole game; then at runtime solve a depth limited version of the game considering players playing the approximated equilibrium after the limit



Original Game Abstract Game

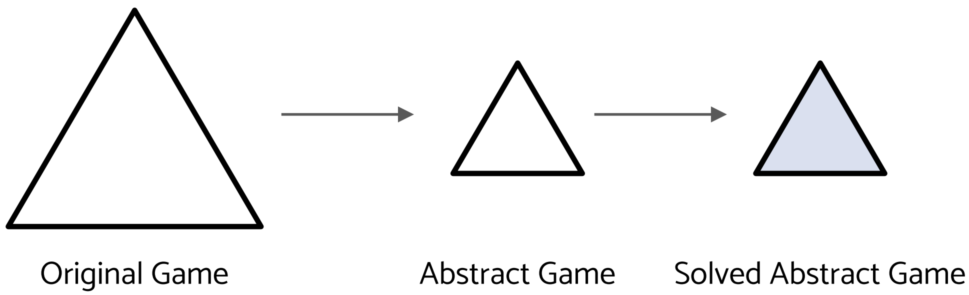
SOLVING 2P0S GAMES

Approaches to the solution of 2p0s games

1 **Linear Programming approaches**
Describe a Nash Equilibrium in the game as a Linear Program

Not scalable to large games

2 **Abstraction + Continual Resolving**
Use a coarse representation of the original game to obtain a approximatively good strategy on whole game; then at runtime solve a depth limited version of the game considering players playing the approximated equilibrium after the limit



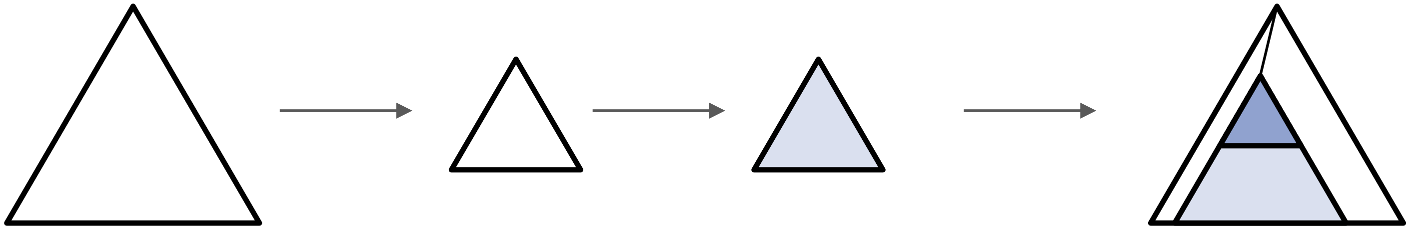
SOLVING 2P0S GAMES

Approaches to the solution of 2p0s games

1. **Linear Programming approaches**
   Describe a Nash Equilibrium in the game as a Linear Program

Not scalable to large games

2. **Abstraction + Continual Resolving**
   Use a coarse representation of the original game to obtain a approximatively good strategy on whole game; then at runtime solve a depth limited version of the game considering players playing the approximated equilibrium after the limit



Original Game | Abstract Game | Solved Abstract Game | Continual Resolving at runtime

SOLVING 2P0S GAMES

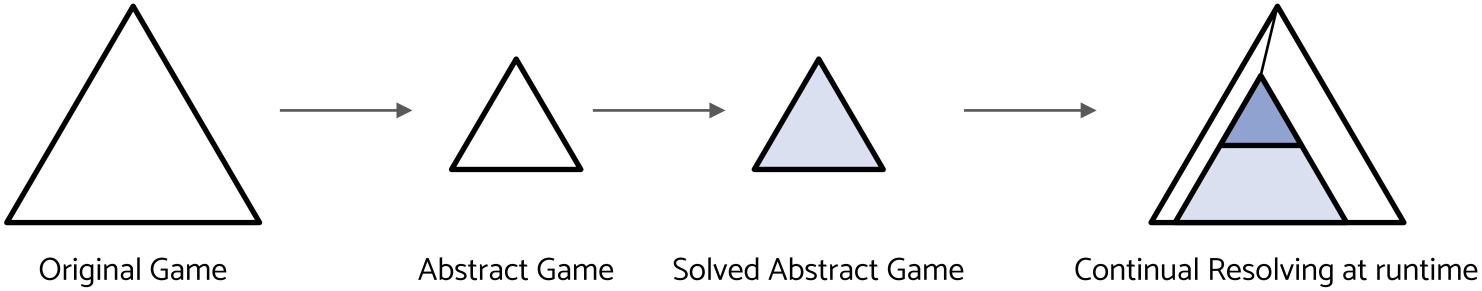
Approaches to the solution of 2p0s games

1. **Linear Programming approaches**
   Describe a Nash Equilibrium in the game as a Linear Program

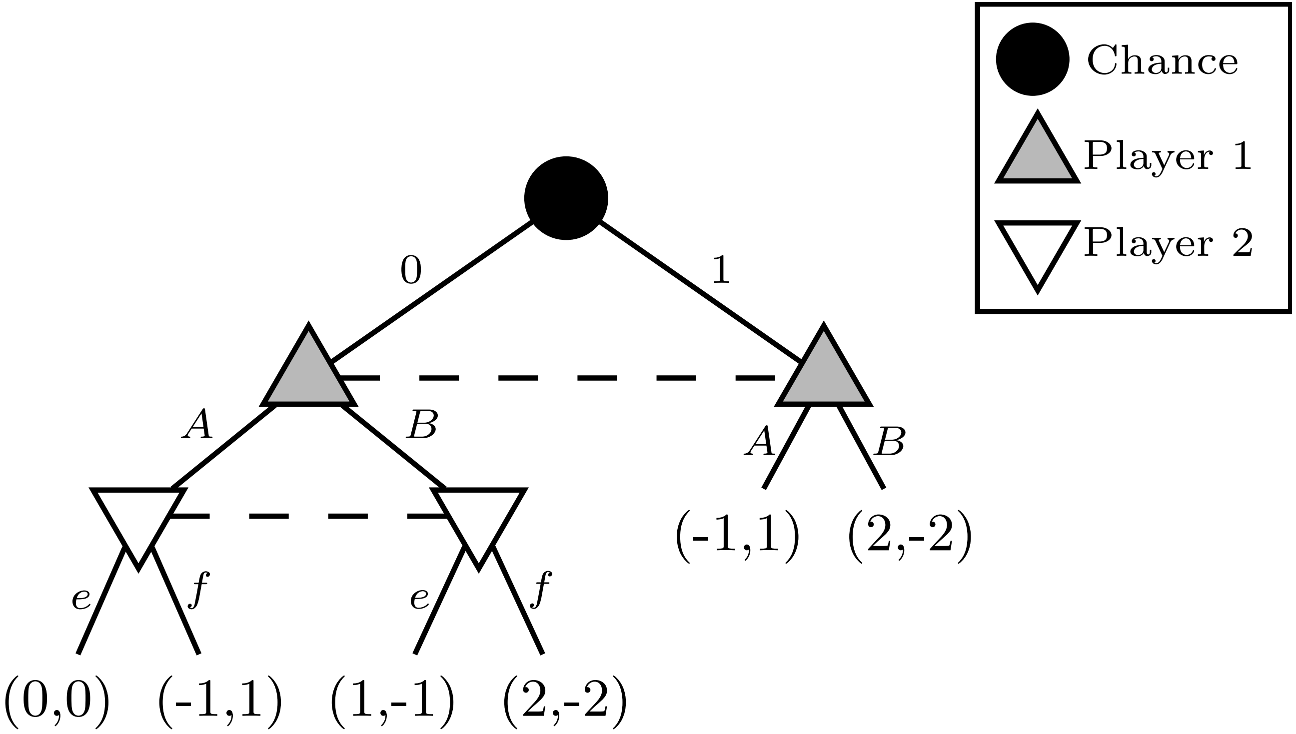
2. **Abstraction + Continual Resolving**
   Use a coarse representation of the original game to obtain a approximatively good strategy on whole game; then at runtime solve a depth limited version of the game considering players playing the approximated equilibrium after the limit

Not scalable to large games

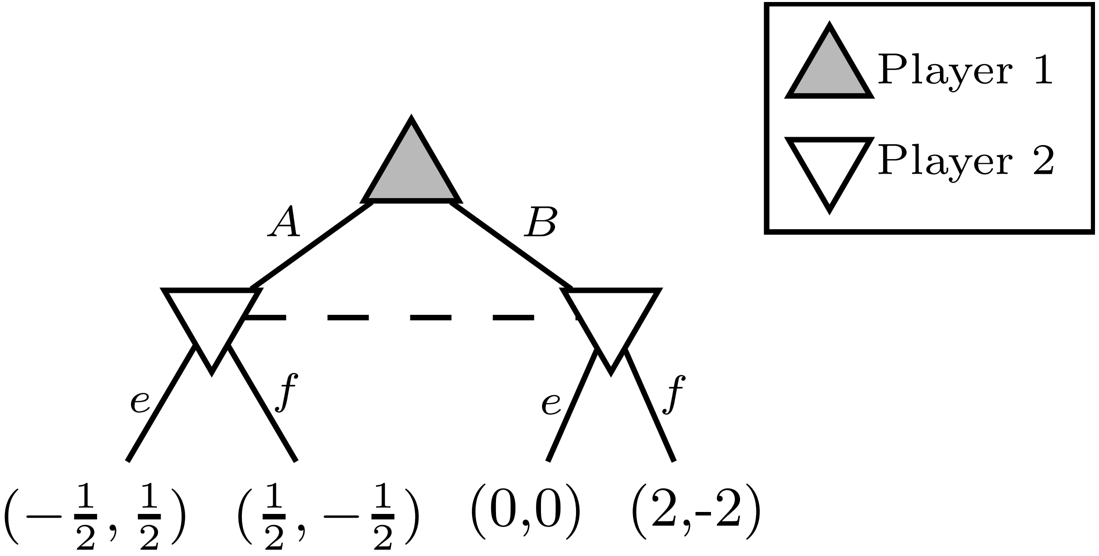
Winning Approach: **Superhuman perfomances in Poker**



EXAMPLE OF ABSTRACTED GAME



EXAMPLE OF ABSTRACTED GAME

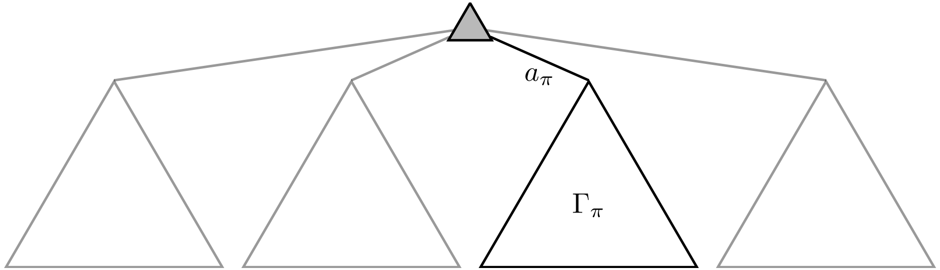


SOLVING ADVERSARIAL TEAM GAMES

The **extensive form representation is not enough** to represent the *ex ante coordination* for the team. A **Auxiliary Game** is created by introducing an initial coodination node, in which the **players select the fixed strategy π** they will use in the game.

SOLVING ADVERSARIAL TEAM GAMES

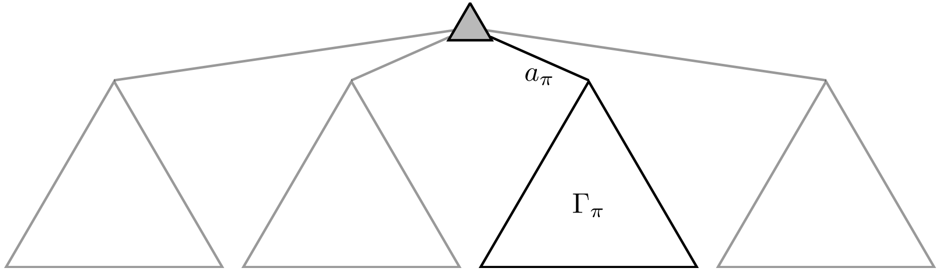
The **extensive form representation is not enough** to represent the *ex ante coordination* for the team. A **Auxiliary Game** is created by introducing an initial coodination node, in which the **players select the fixed strategy π** they will use in the game.



SOLVING ADVERSARIAL TEAM GAMES

The **extensive form representation is not enough** to represent the *ex ante coordination* for the team. A **Auxiliary Game** is created by introducing an initial coodination node, in which the **players select the fixed strategy π** they will use in the game.

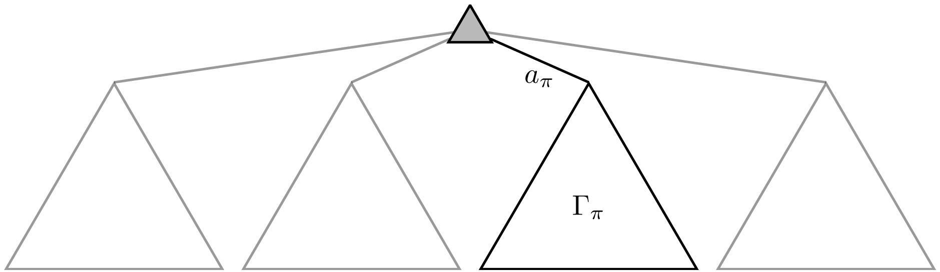
HCG:
π is a plan for both players
(Celli and Gatti 2018)



SOLVING ADVERSARIAL TEAM GAMES

The **extensive form representation is not enough** to represent the *ex ante coordination* for the team. A **Auxiliary Game** is created by introducing an initial coodination node, in which the **players select the fixed strategy π** they will use in the game.

HCG:
π is a plan for both players
(Celli and Gatti 2018)

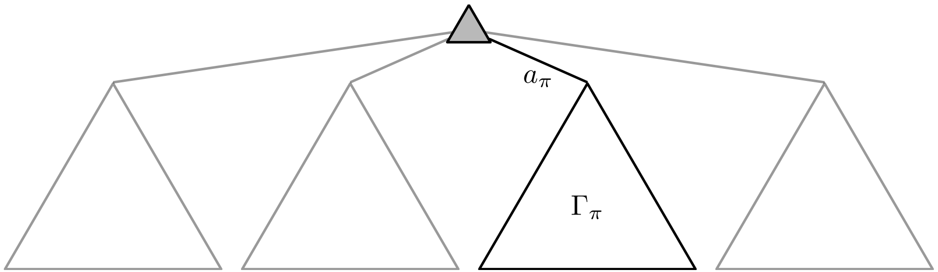


FTP, FCG:
π is a plan for only one of the players
(Farina et al., 2018)
(Farina et al., 2021)

SOLVING ADVERSARIAL TEAM GAMES

The **extensive form representation is not enough** to represent the *ex ante coordination* for the team. A **Auxiliary Game** is created by introducing an initial coodination node, in which the **players select the fixed strategy π** they will use in the game.

HCG:
π is a plan for both players
(Celli and Gatti 2018)



FTP, FCG:
π is a plan for only one of the players
(Farina et al., 2018)
(Farina et al., 2021)

1

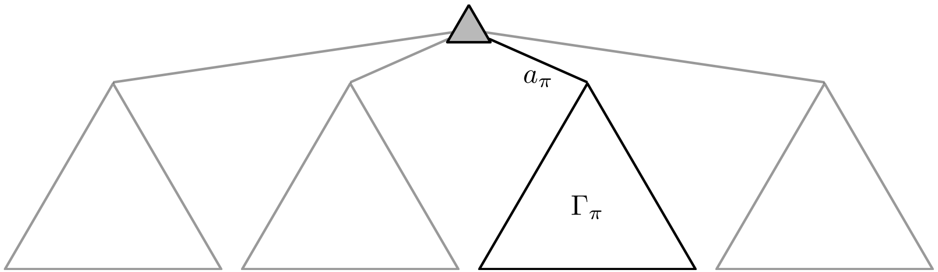
Linear Programming approaches

Describe a TMECor in Auxiliary Game as a Linear Program. Use a **Column Generation** algorithm to iteratively add plans, without explicitly representing the complete coordination node

SOLVING ADVERSARIAL TEAM GAMES

The **extensive form representation is not enough** to represent the *ex ante coordination* for the team. A **Auxiliary Game** is created by introducing an initial coodination node, in which the **players select the fixed strategy π** they will use in the game.

HCG:
π is a plan for both players
(Celli and Gatti 2018)



FTP, FCG:
π is a plan for only one of the players
(Farina et al., 2018)
(Farina et al., 2021)

1

Linear Programming approaches

Describe a TMECor in Auxiliary Game as a Linear Program. Use a **Column Generation** algorithm to iteratively add plans, without explicitly representing the complete coordination node

Not scalable to large games

SOLVING ADVERSARIAL TEAM GAMES

Can we Abstract & Continually Resolve an Auxiliary Game?

SOLVING ADVERSARIAL TEAM GAMES

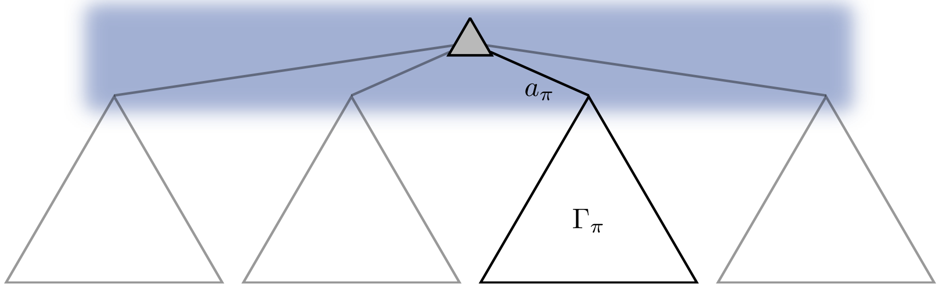
Can we Abstract & Continually Resolve an Auxiliary Game?

No, because the choice of **plans** has no structure
The ***”flatness”*** does not allow any subgame solving

SOLVING ADVERSARIAL TEAM GAMES

Can we Abstract & Continually Resolve an Auxiliary Game?

No, because the choice of **plans** has no structure
The ***"flatness"*** does not allow any subgame solving

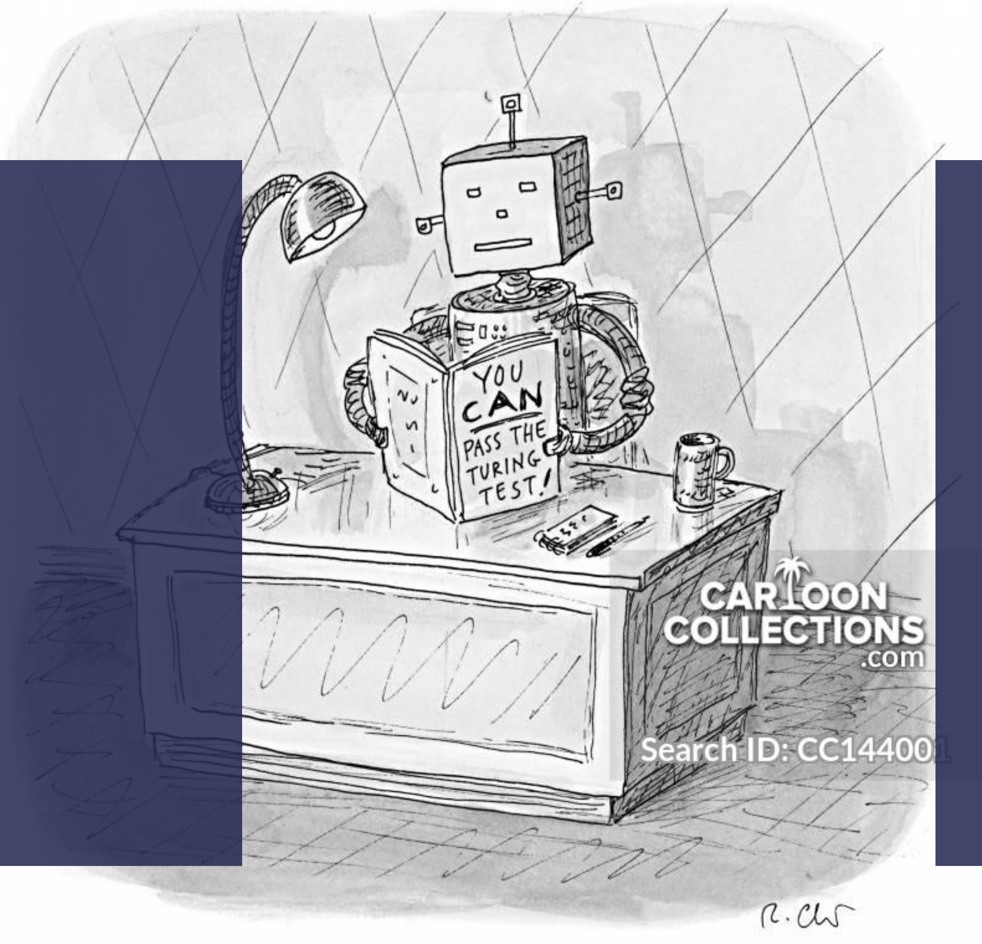


OUR RESEARCH QUESTION

Can we define a better auxiliary game?

We need to keep the game structure

Public Team Conversion procedure



OUR IDEA

Introduce a coordinator which **prescribes an action** to a team player ***during the game.*** *(Nayyar et al., 2013)*

OUR IDEA

Introduce a coordinator which **prescribes an action** to a team player ***during the game.*** *(Nayyar et al., 2013)*

Ex ante coordination:

- ⇒ Team players share a **deterministic strategy of the coordinator** before the game starts
- ⇒ Team players **query** it whenever needed, using the **current state of the game as input**
- ⇒ In order to be shared, the coordinator requires as input **the same information from both players**. Therefore, the **public information state** among the team players is used.

OUR IDEA

Introduce a coordinator which **prescribes an action** to a team player ***during the game.*** *(Nayyar et al., 2013)*

Ex ante coordination:

- ⇒ Team players share a **deterministic strategy of the coordinator** before the game starts
- ⇒ Team players **query** it whenever needed, using the **current state of the game as input**
- ⇒ In order to be shared, the coordinator requires as input **the same information from both players**. Therefore, the **public information state** among the team players is used.

The actions available to the coordinator are called **Prescriptions**. They prescribe a deterministic action for any possible private state of the current player. The players passively apply them according to their private state

OUR IDEA

Introduce a coordinator which **prescribes an action** to a team player ***during the game.*** *(Nayyar et al., 2013)*

Ex ante coordination:

- $\Rightarrow$ Team players share a **deterministic strategy of the coordinator** before the game starts
- $\Rightarrow$ Team players **query** it whenever needed, using the **current state of the game as input**
- $\Rightarrow$ In order to be shared, the coordinator requires as input **the same information from both players**. Therefore, the **public information state** among the team players is used.

The actions available to the coordinator are called **Prescriptions**. They prescribe a deterministic action for any possible private state of the current player. The players passively apply them according to their private state

$\approx$ Localized plan correlation **$\neq$ Ex ante whole plan coordination**

OUR IDEA

Example application in a Poker instance with only two cards, Q and J

OUR IDEA

Example application in a Poker instance with only two cards, Q and J

Information State of a player

3. Public Team Conv

OUR IDEA

Example application in a Poker instance with only two cards, Q and J

Information State of a player

Public Information = Sequence of played actions

$$Player1\colon\ Raise, Player2\colon\ Call$$

OUR IDEA

Example application in a Poker instance with only two cards, Q and J

Information State of a player

Public Information = Sequence of played actions

$$Player1:\ Raise, Player2:\ Call$$

Private Information = Card held by the player

OUR IDEA

Example application in a Poker instance with only two cards, Q and J

Information State of a player

Public Information = Sequence of played actions

$Player1{:}\ Raise, Player2{:}\ Call$

Private Information = Card held by the player

Query Coordinator Strategy

$$\boldsymbol{\sigma}(\ Player1{:}\ Raise, Player2{:}\ Call\ ) = \boldsymbol{\Gamma}$$

3. Public Team Conv

OUR IDEA

Example application in a Poker instance with only two cards, Q and J

Information State of a player

Public Information = Sequence of played actions

$Player1:\ Raise, Player2:\ Call$

Private Information = Card held by the player

Query Coordinator Strategy

$\boldsymbol{\sigma}(\ Player1:\ Raise, Player2:\ Call\ ) = \boldsymbol{\Gamma}$

Apply Prescription

$\boldsymbol{\Gamma}($

$) = Call$

3. Public Team Conv

OUR CONTRIBUTIONS

We developed $\textsc{PublicTeamConversion}$ procedure, able to **convert a generic instance of game to the auxiliary representation** we have described.

OUR CONTRIBUTIONS

We developed $\textsc{PublicTeamConversion}$ procedure, able to **convert a generic instance of game to the auxiliary representation** we have described.

The resulting game is a **2pOs** game (coordinator vs adversary) that shares the **same public information structure as the original game**

OUR CONTRIBUTIONS

We developed PUBLICTEAMCONVERSION procedure, able to **convert a generic instance of game to the auxiliary representation** we have described.

The resulting game is a **2pOs** game (coordinator vs adversary) that shares the **same public information structure as the original game**

Our main result is **Theorem 2**:

- **The converted game is strategically equivalent to the original one**, since there exists a mapping of equivalent strategies between the original and the converted game
- **A Nash Equilibrium in the converted game is a TMECor in the original game**

Its proof is particularly complex, and is out of the scope of this presentation.

OUR CONTRIBUTIONS

Additional results regarding the **NP-hardness of the problem vs the potential exponential increase in size** of the converted game due to the **combinatorial structure of prescriptions**:

OUR CONTRIBUTIONS

Additional results regarding the **NP-hardness of the problem vs the potential exponential increase in size** of the converted game due to the **combinatorial structure of prescriptions**:

- (Farina et al., 2021) proved that finding a TMECor in a **game without private observations** can be done in **polynomial time**. We verified that our conversion procedure does not increase the size of the game in those cases.

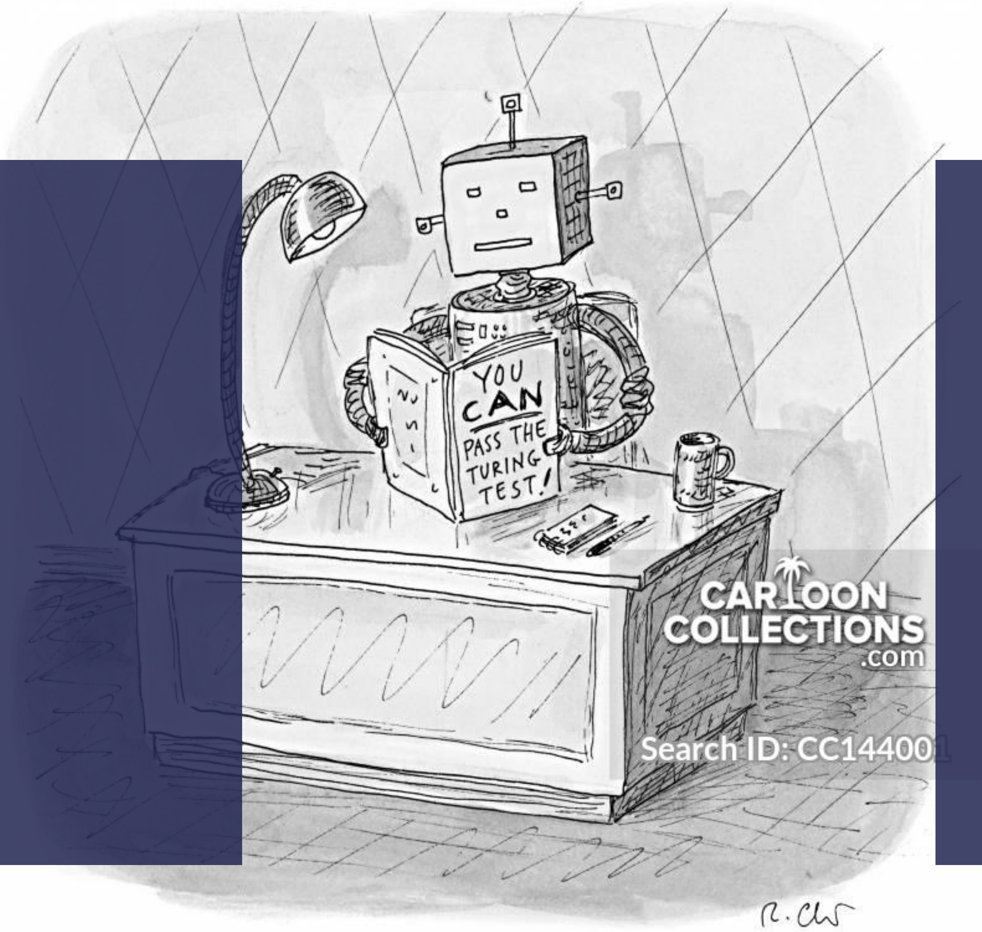
OUR CONTRIBUTIONS

Additional results regarding the **NP-hardness of the problem vs the potential exponential increase in size** of the converted game due to the **combinatorial structure of prescriptions**:

- (Farina et al., 2021) proved that finding a TMECor in a **game without private observations** can be done in **polynomial time**. We verified that our conversion procedure does not increase the size of the game in those cases.

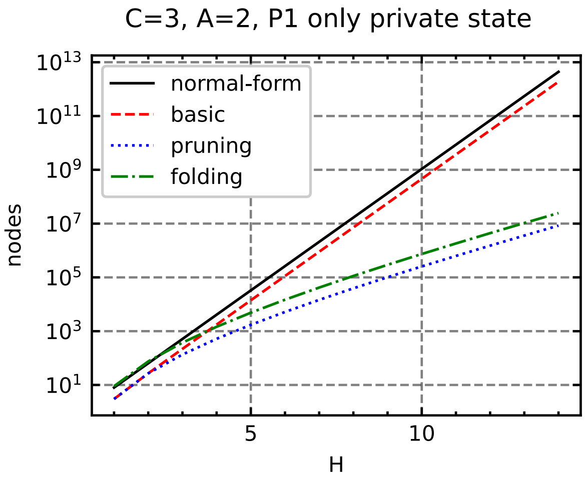
- We developed **three pruning techniques** to mitigate the exponential increase in size of the converted game.
  - **Belief-based pruning**
  - **Folding representation**
  - **Safe Imperfect recall**

Experimental Results

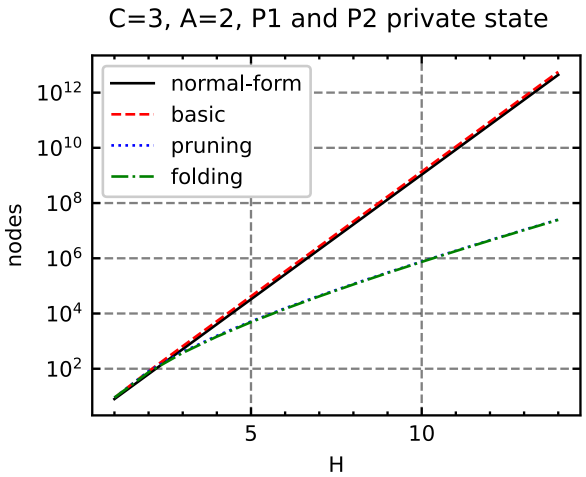


TEST 1 – Impact of Pruning Techniques

We evaluated the impact of the developed pruning techniques on a parametric game with C private states, A actions per node and a variable number of H levels of action



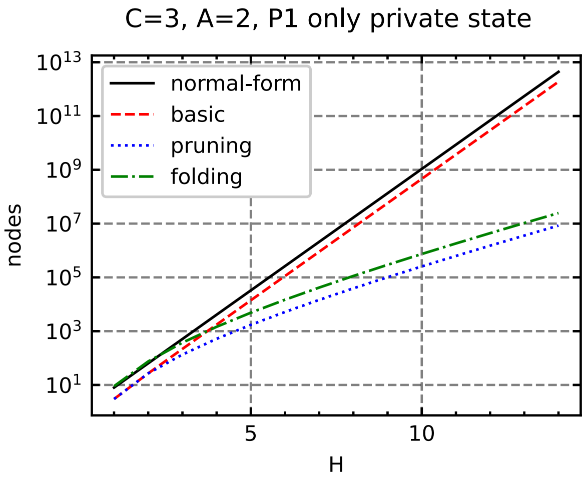
Case when C=3 and A=2, only a single player with private information



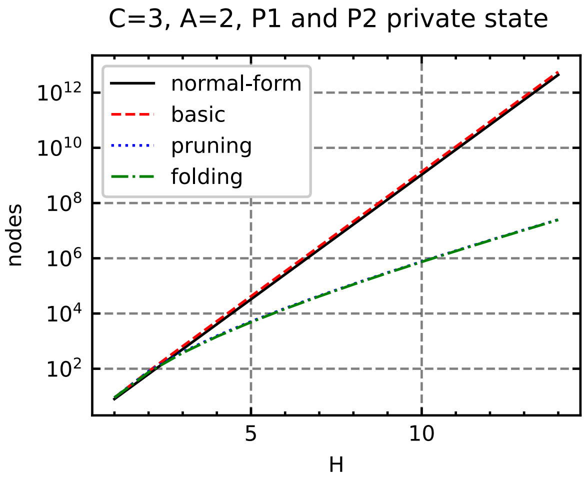
Case when C=3 and A=2, both players with private information

TEST 1 – Impact of Pruning Techniques

We evaluated the impact of the developed pruning techniques on a parametric game with C private states, A actions per node and a variable number of H levels of action



Case when C=3 and A=2, only a single player with private information

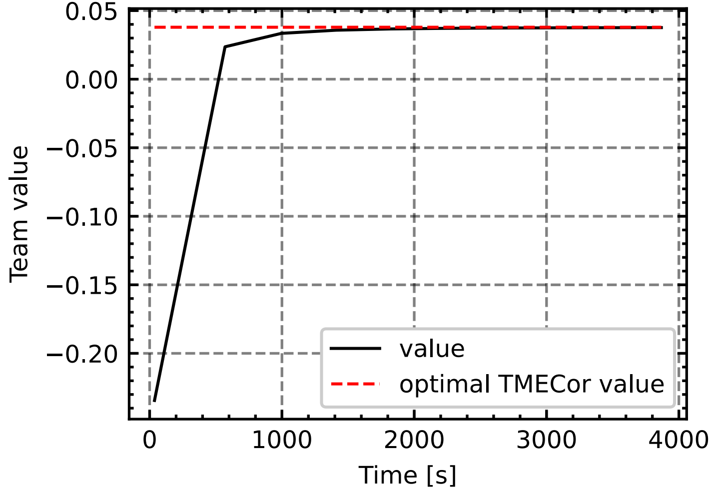


Case when C=3 and A=2, both players with private information

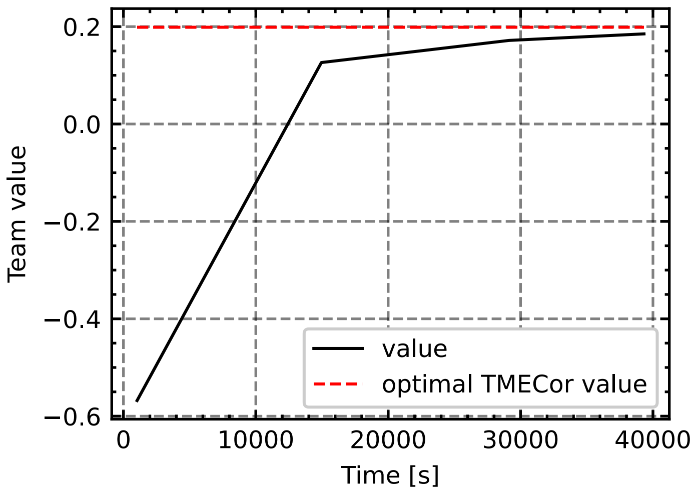
CONCLUSION: we reduce the size of the resulting game up to a square root factor

TEST 2 – Application to Poker instances

We applied standard solving algorithms to the folding representation of Kuhn and Leduc instances



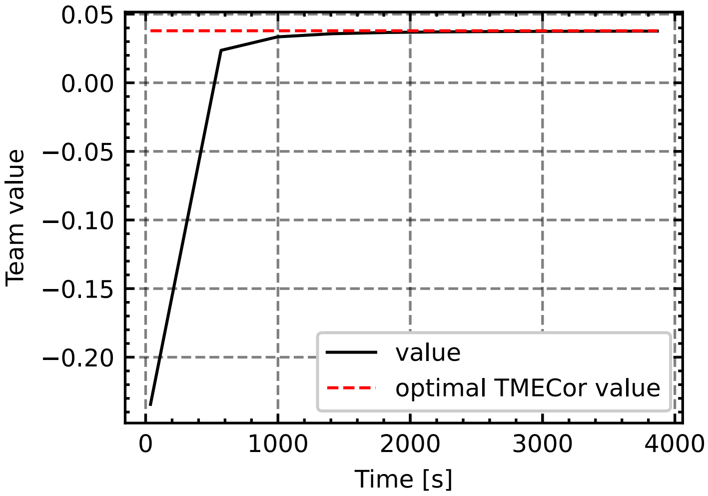
Kuhn Poker, 3 ranks and adversary playing first



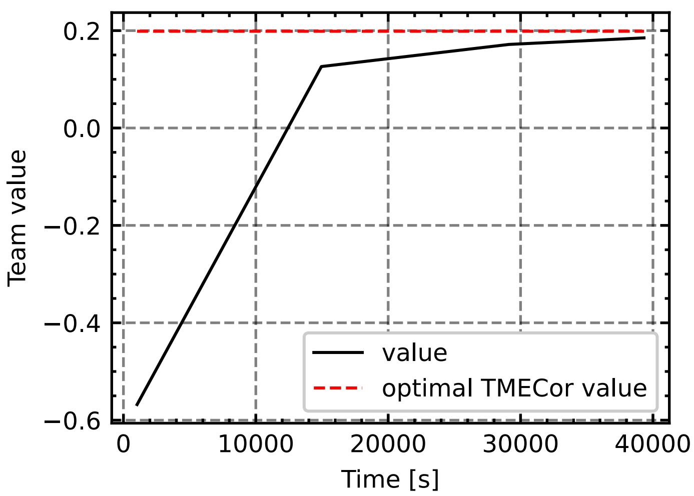
Leduc Poker, 3 ranks, 1 raise, and adversary playing first

TEST 2 – Application to Poker instances

We applied standard solving algorithms to the folding representation of Kuhn and Leduc instances



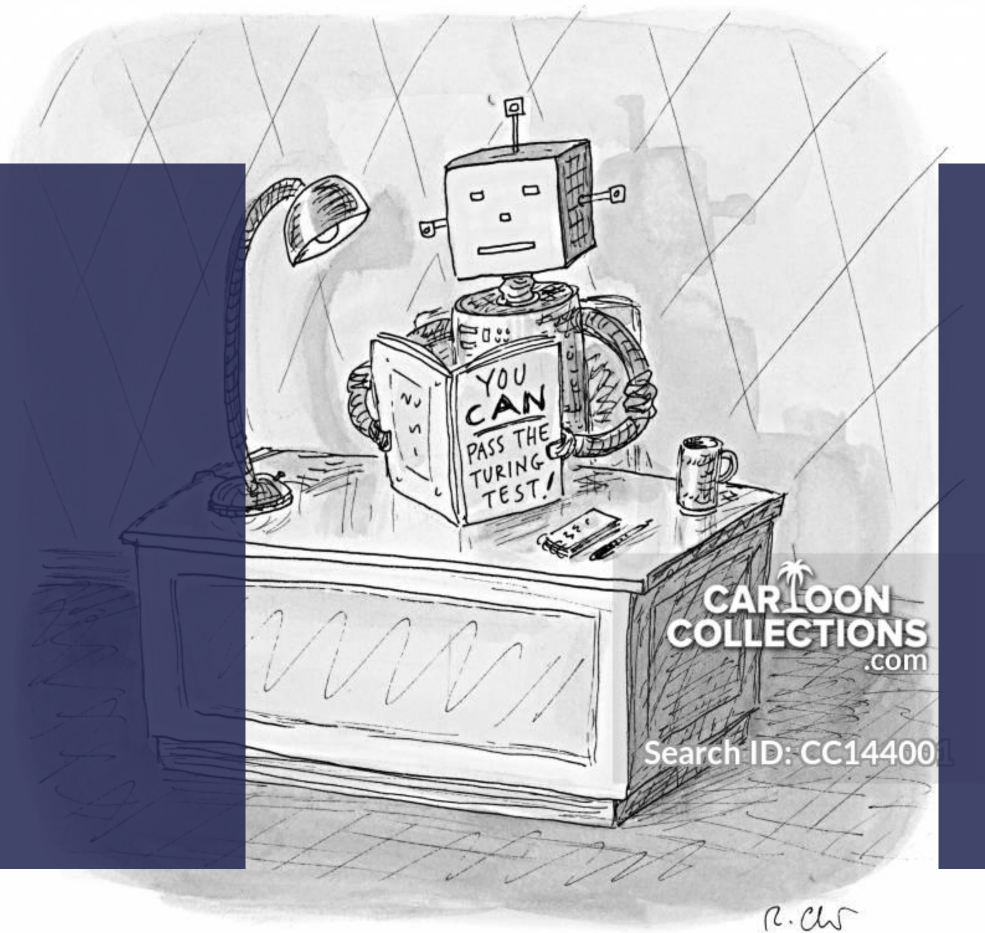
Kuhn Poker, 3 ranks and adversary playing first



Leduc Poker, 3 ranks, 1 raise, and adversary playing first

CONCLUSION: convergence in value to a TMECor is achieved, coherently with our theoretical result

Implications and Future Works



IMPLICATIONS

Our theoretical contribution enables important future developments:

IMPLICATIONS

Our theoretical contribution enables important future developments:

Online Conversion procedure

The $\textsc{PublicTeamConversion}$ procedure is **recursive**; this allows to avoid the complete construction of a new game, and enables the online generation of trajectories in the converted game

IMPLICATIONS

Our theoretical contribution enables important future developments:

Online Conversion procedure

The PUBLICTEAMCONVERSION procedure is **recursive**; this allows to avoid the complete construction of a new game, and enables the online generation of trajectories in the converted game

Abstraction & Continual resolving for Adversarial Team Games

Since our converted game is a 2p0s game presenting a sequential structure, the framework fo Abstraction and Continual resolving can be applied to potentially allow the development of techniques for larger instances of the game

IMPLICATIONS

Our theoretical contribution enables important future developments:

Online Conversion procedure

The PUBLICTEAMCONVERSION procedure is **recursive**; this allows to avoid the complete construction of a new game, and enables the online generation of trajectories in the converted game

Abstraction & Continual resolving for Adversarial Team Games

Since our converted game is a 2p0s game presenting a sequential structure, the framework fo Abstraction and Continual resolving can be applied to potentially allow the development of techniques for larger instances of the game

Extension to N vs N games

From a theoretical point of view, the idea of a shared coordinator sending prescription to team members can be extended also in the case in which two teams of agents are interacting one against the other

FUTURE WORKS

Our next steps:

FUTURE WORKS

Our next steps:

Further Reducing the converted game size

Investigate whether a more efficient pruning technique or a different scheme of auxiliary game can lead to smaller size of the converted games

FUTURE WORKS

Our next steps:

Further Reducing the converted game size

Investigate whether a more efficient pruning technique or a different scheme of auxiliary game can lead to smaller size of the converted games

Find good Abstractions and efficiently implement Continual Resolving

Find if general purpose abstractions techniques can be formulated given the specificities of the converted games, and implement the Abstraction + Continual Resolving pipeline to compare with previous results

FUTURE WORKS

Our next steps:

Further Reducing the converted game size

Investigate whether a more efficient pruning technique or a different scheme of auxiliary game can lead to smaller size of the converted games

Find good Abstractions and efficiently implement Continual Resolving

Find if general purpose abstractions techniques can be formulated given the specificities of the converted games, and implement the Abstraction + Continual Resolving pipeline to compare with previous results

Better algorithms to cope with the asymmetry of coordinator vs adversary

The converted game is highly asymmetrical in size, with most of the tree occupied by actions of the coordinator. Our idea is that of tweaking solving algorithms to take advantege of this situation.

Thank you for your attention!
Any Questions?

Contacts:
luca5.carminati@mail.polimi.it

REFERENCES

https://www.cooperativeai.com

Dafoe, A., Hughes, E., Bachrach, Y., Collins, T., McKee, K.R., Leibo, J.Z., Larson, K., & Graepel, T. (2020). Open Problems in Cooperative AI. ArXiv, abs/2012.08630.

Nash, J.F. (1951). NON-COOPERATIVE GAMES. Annals of Mathematics, 54, 286.

Stengel, B.V., & Koller, D. (1997). Team-Maxmin Equilibria. Games and Economic Behavior, 21, 309-321.

Celli, A., & Gatti, N. (2018). Computational Results for Extensive-Form Adversarial Team Games. AAAI.

Farina, G., Celli, A., Gatti, N., & Sandholm, T. (2018). Ex ante coordination and collusion in zero-sum multi-player extensive-form games. NeurIPS.

Farina, G., Celli, A., Gatti, N., & Sandholm, T. (2021). Connecting Optimal Ex-Ante Collusion in Teams to Extensive-Form Correlation: Faster Algorithms and Positive Complexity Results. ICML.