Generalized Cops and Robbers: *N*-Player Pursuit Games on Graphs

Ath. Kehagias

(and G. Konstantinidis and M. Koutsmanis)

GCR Games: Informal Description

Generalized Cops and Robbers games (GCR games)

GCR Games: Informal Description

Generalized Cops and Robbers games (GCR games)

They are N-**player “pursuit” games** played on graphs.

GCR Games: Informal Description

Generalized Cops and Robbers games (GCR games)

They are N-**player "pursuit" games** played on graphs.

They generalize

GCR Games: Informal Description

Generalized Cops and Robbers games (GCR games)

They are N**-player “pursuit” games** played on graphs.

They generalize

1. Classic *Cops and Robbers* (CR) and its many variants.
2. Bonato and MacGillivray's 2-player *Generalized Cops and Robbers* games.
3. Previous work by Konstantinidis and Kehagias.

GCR Games: Informal Description

Generalized Cops and Robbers games (GCR games)

They are *N*-**player “pursuit” games** played on graphs.

They generalize

1. Classic *Cops and Robbers* (CR) and its many variants.
2. Bonato and MacGillivray’s 2-player *Generalized Cops and Robbers* games.
3. Previous work by Konstantinidis and Kehagias.

They are special cases of

GCR Games: Informal Description

Generalized Cops and Robbers games (GCR games)

They are *N*-**player "pursuit" games** played on graphs.

They generalize

1. Classic *Cops and Robbers* (CR) and its many variants.
2. Bonato and MacGillivray's 2-player *Generalized Cops and Robbers* games.
3. Previous work by Konstantinidis and Kehagias.

They are special cases of

1. Stochastic games.
2. Recursive games.
3. Graphical games.
4. Reachability games.

Examples

Examples

Example: Cyclic Pursuit

P_1, P_2, P_3 take turns moving along the edges of G.

1. P_1 wants to capture P_2.
2. P_2 wants to capture P_3.
3. P_3 wants to capture P_1.

Examples

Example: Cyclic Pursuit

P_1, P_2, P_3 take turns moving along the edges of G.

1. P_1 wants to capture P_2.
2. P_2 wants to capture P_3.
3. P_3 wants to capture P_1.

Example: Selfish Cops, Adversarial Robber

P_1, P_2, P_3 take turns moving along the edges of G.

1. Pursuers P_1 and P_2 want to capture P_3.
2. The pursuer who captures P_3 receives higher reward than the "other" pursuer.
3. P_3 wants to capture P_1

Rules

Rules

1. Players P_1, P_2, ..., P_N move along the edges of G.

Rules

1. Players P_1, P_2, ..., P_N move along the edges of G.
2. They start from given initial position s_0.

Rules

1. Players P_1, P_2, ..., P_N move along the edges of G.
2. They start from given initial position s_0.
3. At the t-th turn ($t \in \mathbb{N}$) a single player moves.

Rules

1. Players P_1, P_2, ..., P_N move along the edges of G.
2. They start from given initial position s_0.
3. At the t-th turn ($t \in \mathbb{N}$) a single player moves.
4. Game ends when some "capturing" position $s_t \in S_c$ is reached.

Rules

1. Players P_1, P_2, ..., P_N move along the edges of G.
2. They start from given initial position s_0.
3. At the t-th turn ($t \in \mathbb{N}$) a single player moves.
4. Game ends when some "capturing" position $s_t \in S_c$ is reached.
5. On game end each player receives a payoff.

Rules

1. Players P_1, P_2, ..., P_N move along the edges of G.
2. They start from given initial position s_0.
3. At the t-th turn ($t \in \mathbb{N}$) a single player moves.
4. Game ends when some "capturing" position $s_t \in S_c$ is reached.
5. On game end each player receives a payoff.

States

States

1. *Nonterminal states*:

$$S = \left\{ \left(x^1, ..., x^N, i \right) : \left(x^1, ..., x^N \right) \in V^n \text{and } i \in [N] \right\}$$

$x^n \in V$ is the position (vertex) of the n-th player
$i \in [N]$ is the player who has the next move.

States

1. *Nonterminal states*:

   $$S=\left\{\left(x^1,...,x^N,i\right):\left(x^1,...,x^N\right)\in V^n\text{and } i\in[N]\right\}$$

   $x^n\in V$ is the position (vertex) of the n-th player
   $i\in[N]$ is the player who has the next move.
2. $S=S_{nc}\cup S_c$ where
   1. S_{nc} are the *noncapture states*.
   2. S_c are the *capture states*,

States

1. *Nonterminal states*:

   $$S = \left\{ \left(x^1, ..., x^N, i \right) : \left(x^1, ..., x^N \right) \in V^n \text{and } i \in [N] \right\}$$

   $x^n \in V$ is the position (vertex) of the n-th player
   $i \in [N]$ is the player who has the next move.
2. $S = S_{nc} \cup S_c$ where
   1. S_{nc} are the *noncapture states*.
   2. S_c are the *capture states*,
3. *Terminal state* τ.

States

1. *Nonterminal states*:

   $$S=\left\{\left(x^1,...,x^N,i\right):\left(x^1,...,x^N\right)\in V^n\text{and } i\in[N]\right\}$$

   $x^n \in V$ is the position (vertex) of the n-th player
   $i \in [N]$ is the player who has the next move.
2. $S = S_{nc} \cup S_c$ where
   1. S_{nc} are the *noncapture states*.
   2. S_c are the *capture states*,
3. *Terminal state* τ.
4. State set is $\overline{S} = S_{nc} \cup S_c \cup \{\tau\}$.

Histories and Strategies

Histories and Strategies

Histories are state sequences (in a play of the game).

1. Histories of length k : $H_k = \{\mathbf{s} = s_0 s_1 ... s_k\}$;
2. Histories of finite length: $H_* = \cup_{k=1}^{\infty} H_k$;
3. Histories of infinite length: $H_\infty = \{\mathbf{s} = s_0 s_1 ... s_k ...\}$.

Histories and Strategies

Histories are state sequences (in a play of the game).

1. Histories of length k : $H_k = \{\mathbf{s} = s_0 s_1 ... s_k\}$;
2. Histories of finite length: $H_* = \cup_{k=1}^{\infty} H_k$;
3. Histories of infinite length: $H_\infty = \{\mathbf{s} = s_0 s_1 ... s_k ...\}$.

Strategies map histories to next moves.

1. *Deterministic strategy*: $\sigma^n : H_* \to V$ (assigns a move to each finite-length history)
2. *Positional* strategy $\sigma^n (s_0 s_1 ... s_t) = \sigma^n (s_t)$.
3. *Notation*: $\sigma^{-n} = \left(\sigma^j\right)_{j \in [N] \backslash \{n\}}$
   E.g., $\sigma = \left(\sigma^1, \sigma^2, \sigma^3\right)$, $\sigma^{-1} = \left(\sigma^2, \sigma^3\right)$.

Payoffs

Payoffs

1. *Total payoff* of the n-th player $Q^n : H_\infty \to \mathbb{R}$

$$Q^n(s_0, s_1, s_2, ...) = Q^n(s_0, \sigma) = \sum_{t=0}^{\infty} \gamma^t q^n(s_t)$$

Payoffs

1. *Total payoff* of the n-th player $Q^n : H_\infty \to \mathbb{R}$

$$Q^n(s_0, s_1, s_2, \ldots) = Q^n(s_0, \sigma) = \sum_{t=0}^{\infty} \gamma^t q^n(s_t)$$

2. *Turn payoff* of the n-th player $q^n : S \to \mathbb{R}$
   So on game state s, P_n gets $q^n(s)$.
   We will assume $s \in S_{nc} \Rightarrow q^n(s) = 0$.

Payoffs

1. *Total payoff* of the n-th player $Q^n : H_\infty \to \mathbb{R}$

$$Q^n(s_0, s_1, s_2, \ldots) = Q^n(s_0, \sigma) = \sum_{t=0}^{\infty} \gamma^t q^n(s_t)$$

2. *Turn payoff* of the n-th player $q^n : S \to \mathbb{R}$
   So on game state s, P_n gets $q^n(s)$.
   We will assume $s \in S_{nc} \Rightarrow q^n(s) = 0$.
3. *Discount factor* $\gamma \in (0,1)$.

Payoffs

1. *Total payoff* of the n-th player $Q^n : H_\infty \to \mathbb{R}$

$$Q^n(s_0, s_1, s_2, \ldots) = Q^n(s_0, \sigma) = \sum_{t=0}^{\infty} \gamma^t q^n(s_t)$$

2. *Turn payoff* of the n-th player $q^n : S \to \mathbb{R}$
   So on game state s, P_n gets $q^n(s)$.
   We will assume $s \in S_{nc} \Rightarrow q^n(s) = 0$.
3. *Discount factor* $\gamma \in (0, 1)$.

Each player tries to maximize his payoff.

Generalized Cops and Robbers Games

Generalized Cops and Robbers Games

1. A *GCR game*:

$$\left(G, \overline{S}, S_c, \mathbf{q}, \gamma, s_0\right)$$

where s_0 is the *initial state*.

Generalized Cops and Robbers Games

1. A *GCR game*:

$$\left(G,\overline{S},S_c,\mathbf{q},\gamma,s_0\right)$$

where s_0 is the *initial state*.

2. A *GCR family*:

$$\left(G,\overline{S},S_c,\mathbf{q},\gamma\right)=\left\{\left(G,\overline{S},S_c,\mathbf{q},\gamma,s_0\right):s_0\in S\right\}$$

I.e., the same game played from all possible initial positions.

Generalized Cops and Robbers Games

1. A *GCR game*:

$$\left(G, \overline{S}, S_c, \mathbf{q}, \gamma, s_0\right)$$

where s_0 is the *initial state*.

2. A *GCR family*:

$$\left(G, \overline{S}, S_c, \mathbf{q}, \gamma\right) = \left\{\left(G, \overline{S}, S_c, \mathbf{q}, \gamma, s_0\right) : s_0 \in S\right\}$$

I.e., the same game played from all possible initial positions.

When the context is clear, we will also write

$$\Gamma(G, s_0) \text{ and } \Gamma(G).$$

Example: Two-player (and two-token) Cops and Robbers

The capture set is

$$S_c = \{s : (x, x, p), x \in V, p \in [2]\}$$

The payoffs are

$$q^1(s) = -q^2(s) = \begin{cases} 1 & \text{iff} \quad s \in S_c, \\ 0 & \text{else;} \end{cases}$$

This is a zero-sum game.

Example: Two-player (and two-token) Cops and Robbers

Example: Two-player (and two-token) Cops and Robbers

Why is this Cops and Robbers?

Example: Two-player (and two-token) Cops and Robbers

Why is this Cops and Robbers?

1. Let $Q^n(s_0, \sigma^1, \sigma^2)$ be the payoff to P_n ($n \in [2]$).
2. Let $T_c(s_0, \sigma^1, \sigma^2)$ be the capture time.

Example: Two-player (and two-token) Cops and Robbers

Why is this Cops and Robbers?

1. Let $Q^n(s_0, \sigma^1, \sigma^2)$ be the payoff to P_n ($n \in [2]$).
2. Let $T_c(s_0, \sigma^1, \sigma^2)$ be the capture time.

Then we have

$$Q^1(s_0, \sigma^1, \sigma^2) = -Q^2(s_0, \sigma^1, \sigma^2) = \gamma^{T_c(s_0, \sigma^1, \sigma^2)}$$

Example: Two-player (and two-token) Cops and Robbers

Why is this Cops and Robbers?

1. Let $Q^n(s_0, \sigma^1, \sigma^2)$ be the payoff to P_n ($n \in [2]$).
2. Let $T_c(s_0, \sigma^1, \sigma^2)$ be the capture time.

Then we have

$$Q^1(s_0, \sigma^1, \sigma^2) = -Q^2(s_0, \sigma^1, \sigma^2) = \gamma^{T_c(s_0, \sigma^1, \sigma^2)}$$

P_1 wants to max (P_2 wants to min) $T_c(s_0, \sigma^1, \sigma^2)$.

Example: Two-player (and two-token) Cops and Robbers

Why is this Cops and Robbers?

1. Let $Q^n(s_0, \sigma^1, \sigma^2)$ be the payoff to P_n ($n \in [2]$).
2. Let $T_c(s_0, \sigma^1, \sigma^2)$ be the capture time.

Then we have

$$Q^1(s_0, \sigma^1, \sigma^2) = -Q^2(s_0, \sigma^1, \sigma^2) = \gamma^{T_c(s_0, \sigma^1, \sigma^2)}$$

P_1 wants to max (P_2 wants to min) $T_c(s_0, \sigma^1, \sigma^2)$.

Actually the equilibrium (*value*) of the game is

$$\sup_{\sigma^2} \inf_{\sigma^1} Q^1(s_0, \sigma^1, \sigma^2) = \inf_{\sigma^1} \sup_{\sigma^2} Q^1(s_0, \sigma^1, \sigma^2).$$

Example: Three-player Linear Pursuit

The capture set is $S_c = \widetilde{S}^1 \cup \widetilde{S}^2 \cup \widetilde{S}^{12}$

$$\begin{aligned} \widetilde{S}^1 &= \{s : (x,x,y,p), x \neq y \in V, p \in [3]\} \\ \widetilde{S}^2 &= \{s : (y,x,x,p), x \neq y \in V, p \in [3]\} \\ \widetilde{S}^{12} &= \{s : (x,x,x,p), x \in V, p \in [3]\} \end{aligned}$$

The payoffs are

$$q^1(s) = \begin{cases} 1 & \text{iff} \quad s \in \widetilde{S}^1 \cup \widetilde{S}^{12}, \\ 0 & \text{else}; \end{cases}$$

$$q^2(s) = \begin{cases} -1 & \text{iff} \quad s \in \widetilde{S}^1, \\ 1 & \text{iff} \quad s \in \widetilde{S}^2, \\ 0 & \text{else}; \end{cases}$$

$$q^3(s) = \begin{cases} -1 & \text{iff} \quad s \in \widetilde{S}^2 \cup \widetilde{S}^{12}, \\ 0 & \text{else}. \end{cases} \quad .$$

Example: Three-player Cyclic Pursuit

The capture set is $S_c = \widetilde{S}^1 \cup \widetilde{S}^2 \cup \widetilde{S}^3 \cup \widetilde{S}^{123}$ where

$$\begin{aligned}
\widetilde{S}^1 &= \{s : (x, x, y, p), x \neq y \in V, p \in [3]\} \\
\widetilde{S}^2 &= \{s : (y, x, x, p), x \neq y \in V, p \in [3]\} \\
\widetilde{S}^3 &= \{s : (x, y, x, p), x \neq y \in V, p \in [3]\} \\
\widetilde{S}^{123} &= \{s : (x, x, x, p), x \in V, p \in [3]\}
\end{aligned}$$

$$q^1(s) = \begin{cases} -1 & \text{iff} \quad s \in \widetilde{S}^3, \\ 1 & \text{iff} \quad s \in \widetilde{S}^1, \\ 0 & \text{else}; \end{cases}$$

$$q^2(s) = \begin{cases} -1 & \text{iff} \quad s \in \widetilde{S}^1, \\ 1 & \text{iff} \quad s \in \widetilde{S}^2, \\ 0 & \text{else}; \end{cases}$$

$$q^3(s) = \begin{cases} -1 & \text{iff} \quad s \in \widetilde{S}^2, \\ 1 & \text{iff} \quad s \in \widetilde{S}^3, \end{cases} \quad .$$

Example: Two Selfish Cops, One Adversarial Robber

The capture set is $S_c = \widetilde{S}^1 \cup \widetilde{S}^2 \cup \widetilde{S}^{12}$ where

$$\begin{aligned} \widetilde{S}^1 &= \{s : s = (x, y, x, p), x \neq y \in V, p \in [3]\} \\ \widetilde{S}^2 &= \{s : s = (x, y, y, p), x \neq y \in V, p \in [3]\} \\ \widetilde{S}^{12} &= \{s : s = (x, x, x, p), x \in V, p \in [3]\} \end{aligned}$$

The payoffs are, with $\varepsilon \in \left[0, \frac{1}{2}\right]$,

$$\text{for } n \in \{1, 2\} : q^n(s) = \begin{cases} 1 - \varepsilon & \text{if } s \in \widetilde{S}^n, \\ \varepsilon & \text{if } s \in \widetilde{S}^m \text{ with } n \neq m, \\ \frac{1}{2} & \text{if } s \in \widetilde{S}^{12}, \\ 0 & \text{else.} \end{cases}$$

$$\text{for } n \in \{3\} : q^n(s) = \begin{cases} -1 & \text{if } s \in S_c, \\ 0 & \text{else.} \end{cases}$$

Positional Nash Equilibrium

Positional Nash Equilibrium

Theorem

For every GCR family $\Gamma(G)$ *there exists a positional deterministic strategy profile* $\widehat{\sigma} = \left(\widehat{\sigma}^1, ..., \widehat{\sigma}^N\right)$ *such that*

$$\forall s \in S, \forall n \in [N], \forall \sigma^n : Q_s^n(s, \widehat{\sigma}^n, \widehat{\sigma}^{-n}) \geq Q_s^n(s, \sigma^n, \widehat{\sigma}^{-n})$$

Positional Nash Equilibrium

Remarks

Positional Nash Equilibrium

Remarks

1. In other words, $\widehat{\sigma}$ is a Nash equilibrium for every particular game $\Gamma(G,s)$:

Positional Nash Equilibrium

Remarks

1. In other words, $\widehat{\sigma}$ is a Nash equilibrium for every particular game $\Gamma(G,s)$:
   P_n has no incentive to *unilaterally* deviate from his strategy $\widehat{\sigma}^n$.

Positional Nash Equilibrium

Remarks

1. In other words, $\widehat{\sigma}$ is a Nash equilibrium for every particular game $\Gamma(G, s)$:
P_n has no incentive to *unilaterally* deviate from his strategy $\widehat{\sigma}^n$.
2. This does not imply global optimality:

Positional Nash Equilibrium

Remarks

1. In other words, $\widehat{\sigma}$ is a Nash equilibrium for every particular game $\Gamma(G,s)$:
   P_n has no incentive to *unilaterally* deviate from his strategy $\widehat{\sigma}^n$.
2. This does not imply global optimality:
   P_n may be able to achieve a payoff higher than $Q^n(s,\widehat{\sigma})$ if more than one players deviate from the strategy profile $\widehat{\sigma}$.

Positional Nash Equilibrium

Remarks

1. In other words, $\widehat{\sigma}$ is a Nash equilibrium for every particular game $\Gamma(G,s)$: P_n has no incentive to *unilaterally* deviate from his strategy $\widehat{\sigma}^n$.
2. This does not imply global optimality: P_n may be able to achieve a payoff higher than $Q^n(s,\widehat{\sigma})$ if more than one players deviate from the strategy profile $\widehat{\sigma}$.
3. In general, more than one NE will exist, some positional and some not.

Nonpositional Nash Equilibrium

Nonpositional Nash Equilibrium

Theorem

For every GCR family $\Gamma(G)$ there exists a **non***positional deterministic strategy profile $\widehat{\sigma} = \left(\widehat{\pi}^1, ..., \widehat{\pi}^N\right)$ such that*

$$\forall s \in S, \forall n \in [N], \forall \sigma^n : Q^n_s(s, \widehat{\pi}^n, \widehat{\pi}^{-n}) \geq Q^n_s(s, \sigma^n, \widehat{\pi}^{-n})$$

Nonpositional Nash Equilibrium

Theorem

For every GCR family $\Gamma(G)$ *there exists a* **non***positional deterministic strategy profile* $\widehat{\sigma} = \left(\widehat{\pi}^1, ..., \widehat{\pi}^N\right)$ *such that*

$$\forall s \in S, \forall n \in [N], \forall \sigma^n : Q^n_s(s, \widehat{\pi}^n, \widehat{\pi}^{-n}) \geq Q^n_s(s, \sigma^n, \widehat{\pi}^{-n})$$

This is proved using *threat strategies*.

Nonpositional Nash Equilibrium

Nonpositional Nash Equilibrium

Given a GCR game $\Gamma(G, s)$ we form N *auxiliary games* $\Gamma_n(G, s)$, $n \in [N]$.

Nonpositional Nash Equilibrium

Given a GCR game $\Gamma(G, s)$ we form N *auxiliary games* $\Gamma_n(G, s)$, $n \in [N]$.

$\Gamma_n(G, s)$ is the *two-player zero-sum* game played between

1. player n (with payoff Q^n)
2. coalition of players $[N] \setminus \{n\}$ (with payoff $-Q^n$).

Nonpositional Nash Equilibrium

Given a GCR game $\Gamma(G,s)$ we form N *auxiliary games* $\Gamma_n(G,s)$, $n \in [N]$.

$\Gamma_n(G,s)$ is the *two-player zero-sum* game played between

1. player n (with payoff Q^n)
2. coalition of players $[N] \setminus \{n\}$ (with payoff $-Q^n$).

Lemma

For each s and n, the game $\Gamma_n(G,s)$ has a value and the players have deterministic positional optimal strategies.

Nonpositional Nash Equilibrium

Given a GCR game $\Gamma(G,s)$ we form N *auxiliary games* $\Gamma_n(G,s)$, $n \in [N]$.

$\Gamma_n(G,s)$ is the *two-player zero-sum* game played between

1. player n (with payoff Q^n)
2. coalition of players $[N] \setminus \{n\}$ (with payoff $-Q^n$).

Lemma

For each s and n, the game $\Gamma_n(G,s)$ has a value and the players have deterministic positional optimal strategies.

Remark

The value and the optimal strategies can be computed by the *Value Iteration algorithm*.

Nonpositional Nash Equilibrium

Threat Strategies

Nonpositional Nash Equilibrium

Threat Strategies

In the game $\Gamma_n(G,s)$, for every n,m, let

1. ρ_n^n be the optimal strategy of P_n.
2. ρ_n^m be the optimal strategy of P_m as a member of the coalition P_{-n}.

Nonpositional Nash Equilibrium

Threat Strategies

In the game $\Gamma_n(G,s)$, for every n, m, let

1. ρ_n^n be the optimal strategy of P_n.
2. ρ_n^m be the optimal strategy of P_m as a member of the coalition P_{-n}.

In the GCR game $\Gamma(G,s)$, the threat strategy $\widehat{\pi}^n$ of player n is defined as follows:

Nonpositional Nash Equilibrium

Threat Strategies

In the game $\Gamma_n(G,s)$, for every n,m, let

1. ρ_n^n be the optimal strategy of P_n.
2. ρ_n^m be the optimal strategy of P_m as a member of the coalition P_{-n}.

In the GCR game $\Gamma(G,s)$, the threat strategy $\widehat{\pi}^n$ of player n is defined as follows:

1. as long as every player $m \neq n$ follows ρ_m^m, player n follows ρ_n^n;
2. as soon as some player $m \neq n$ deviates from ρ_m^m, player n switches to ρ_m^n and uses it for the rest of the game.

Nonpositional Nash Equilibrium

Remarks

Nonpositional Nash Equilibrium

Remarks

1. If P_n deviates then P_{-n} essentially form a coalition and play the *coalition strategy* optimal in $\Gamma_n(G, s)$.

Nonpositional Nash Equilibrium

Remarks

1. If P_n deviates then P_{-n} essentially form a coalition and play the *coalition strategy* optimal in $\Gamma_n(G, s)$.
2. The deviation will be detected immediately, since the game has perfect information.

Nonpositional Nash Equilibrium

Remarks

1. If P_n deviates then P_{-n} essentially form a coalition and play the *coalition strategy* optimal in $\Gamma_n(G,s)$.
2. The deviation will be detected immediately, since the game has perfect information.
3. The strategies $\widehat{\pi}^1, ..., \widehat{\pi}^N$ are not positional.

Nonpositional Nash Equilibrium

Remarks

1. If P_n deviates then P_{-n} essentially form a coalition and play the *coalition strategy* optimal in $\Gamma_n(G, s)$.
2. The deviation will be detected immediately, since the game has perfect information.
3. The strategies $\widehat{\pi}^1, ..., \widehat{\pi}^N$ are not positional. (Actually in certain cases they are ...)

The Value Iteration Algorithm

The Value Iteration Algorithm

Compute the value (essentially the capture time) of CR:

The Value Iteration Algorithm

Compute the value (essentially the capture time) of CR:

```
for $(x^1, x^2, p) \in S$ do
    if $(x^1, x^2, p) \in S_c$ then $U(x^1, x^2, p) = 1$
    else $U(x^1, x^2, p) = 0$
end
while $1 > 0$ do
    $U_{new} = U$
    for $(x^1, x^2) \in V^2$ do
        if $U(x^1, x^2, 1) \neq 0$ then
            $U_{new}(x^1, x^2, 1) = \min_{z \in N[x^1]} \gamma U(z, x^2, 2)$
        end
    end
    for $(x^1, x^2) \in V^2$ do
        if $U(x^1, x^2, 2) \neq 0$ then
            $U_{new}(x^1, x^2, 2) = \max_{z \in N[x^2]} \gamma U(x^1, z, 1)$
        end
    end
    if $U == U_{new}$ then break
    else $U = U_{new}$
end
```

The Value Iteration Algorithm

Remarks

The Value Iteration Algorithm

Remarks

1. This algorithm computes the *value* (essentially the minimax capture time) of the CR game and can be easily extended to compute optimal strategies.

The Value Iteration Algorithm

Remarks

1. This algorithm computes the *value* (essentially the minimax capture time) of the CR game and can be easily extended to compute optimal strategies.
2. The algorithm has been presented by Hahn and MacGillivray ...

The Value Iteration Algorithm

Remarks

1. This algorithm computes the *value* (essentially the minimax capture time) of the CR game and can be easily extended to compute optimal strategies.
2. The algorithm has been presented by Hahn and MacGillivray ...also by Berarducci and Intrigila, Bonato and MacGillivray, Gavenčiak and others.

The Value Iteration Algorithm

Remarks

1. This algorithm computes the *value* (essentially the minimax capture time) of the CR game and can be easily extended to compute optimal strategies.
2. The algorithm has been presented by Hahn and MacGillivray ...also by Berarducci and Intrigila, Bonato and MacGillivray, Gavenčiak and others.
3. It is a special (deterministic) case of the stochastic games **Value Iteration** algorithm.

The Value Iteration Algorithm

Remarks

1. This algorithm computes the *value* (essentially the minimax capture time) of the CR game and can be easily extended to compute optimal strategies.
2. The algorithm has been presented by Hahn and MacGillivray ...also by Berarducci and Intrigila, Bonato and MacGillivray, Gavenčiak and others.
3. It is a special (deterministic) case of the stochastic games **Value Iteration** algorithm.
4. It is also a form of **Backward Induction**

The Value Iteration Algorithm

Remarks

1. This algorithm computes the *value* (essentially the minimax capture time) of the CR game and can be easily extended to compute optimal strategies.
2. The algorithm has been presented by Hahn and MacGillivray ...also by Berarducci and Intrigila, Bonato and MacGillivray, Gavenčiak and others.
3. It is a special (deterministic) case of the stochastic games **Value Iteration** algorithm.
4. It is also a form of **Backward Induction** in the sense that it first solves games of length 0, then of length 1, etc.

The Value Iteration Algorithm

Remarks

1. This algorithm computes the *value* (essentially the minimax capture time) of the CR game and can be easily extended to compute optimal strategies.
2. The algorithm has been presented by Hahn and MacGillivray ...also by Berarducci and Intrigila, Bonato and MacGillivray, Gavenčiak and others.
3. It is a special (deterministic) case of the stochastic games **Value Iteration** algorithm.
4. It is also a form of **Backward Induction** in the sense that it first solves games of length 0, then of length 1, etc.
5. A detailed analysis of the algorithm appears in a 2021 paper by Kehagias and Konstantinidis

The Multi-Value Iteration Algorithm

Question

Can Value Iteration be applied to N-player games?

The Multi-Value Iteration Algorithm

Question

Can Value Iteration be applied to N-player games?

Remarks

The Multi-Value Iteration Algorithm

Question

Can Value Iteration be applied to N-player games?

Remarks

1. Generally, **NO**.

The Multi-Value Iteration Algorithm

Question

Can Value Iteration be applied to N-player games?

Remarks

1. Generally, **NO**. The problem is that in Value Iteration the update is a **contraction** which is applied to a single target function $U(s)$.

The Multi-Value Iteration Algorithm

Question

Can Value Iteration be applied to N-player games?

Remarks

1. Generally, **NO**. The problem is that in Value Iteration the update is a **contraction** which is applied to a single target function $U(s)$.
2. But the above holds for the general N-player stochastic game.

The Multi-Value Iteration Algorithm

Question

Can Value Iteration be applied to N-player games?

Remarks

1. Generally, **NO**. The problem is that in Value Iteration the update is a **contraction** which is applied to a single target function $U(s)$.
2. But the above holds for the general N-player stochastic game.
   Perhaps for GCR games there is some hope?

The Multi-Value Iteration Algorithm

Question

Can Value Iteration be applied to N-player games?

Remarks

1. Generally, **NO**. The problem is that in Value Iteration the update is a **contraction** which is applied to a single target function $U(s)$.
2. But the above holds for the general N-player stochastic game.
   Perhaps for GCR games there is some hope?

What would *Multi*-value Iteration look like?

The Multi-Value Iteration Algorithm

For the case $N = 3$, let us look at the main part of the algorithm.

The Multi-Value Iteration Algorithm

For the case $N = 3$, let us look at the main part of the algorithm.

```
while $1 > 0$ do
    $U_{new} = U$
    for $\left(x^1, x^2, x^3, p\right) \in S$ do
        if $\left(x^1, x^2, x^3, p\right) \in S_{nc}$ then
            for $i \in [3]$ do
                $\widehat{u} = \arg\max_{u \in N[x^i]} \gamma U\left(u, x^{-i}, \mathbf{suc}(p), i\right)$
                $U_{new}\left(x^1, x^2, x^3, p, 1\right) = \gamma U\left(\widehat{u}, x^{-i}, \mathbf{suc}(p), 1\right)$
                $U_{new}\left(x^1, x^2, x^3, p, 2\right) = \gamma U\left(\widehat{u}, x^{-i}, \mathbf{suc}(p), 2\right)$
                $U_{new}\left(x^1, x^2, x^3, p, 3\right) = \gamma U\left(\widehat{u}, x^{-i}, \mathbf{suc}(p), 3\right)$
            end
        end
    end
    if $U == U_{new}$ then break
    else $U = U_{new}$
end
```

The Multi-Value Iteration Algorithm

Remarks

The Multi-Value Iteration Algorithm

Remarks

1. In numerical experiments (on Linear Pursuit, Cyclic Pursuit, SCAR) the algorithm does not always terminate.

The Multi-Value Iteration Algorithm

Remarks

1. In numerical experiments (on Linear Pursuit, Cyclic Pursuit, SCAR) the algorithm does not always terminate.
2. But when applied to trees and cycles, and with a little help (randomization) it *does* always terminate.

The Multi-Value Iteration Algorithm

Remarks

1. In numerical experiments (on Linear Pursuit, Cyclic Pursuit, SCAR) the algorithm does not always terminate.
2. But when applied to trees and cycles, and with a little help (randomization) it *does* always terminate.
3. Termination is good enough because of the following.

The Multi-Value Iteration Algorithm

Remarks

1. In numerical experiments (on Linear Pursuit, Cyclic Pursuit, SCAR) the algorithm does not always terminate.
2. But when applied to trees and cycles, and with a little help (randomization) it *does* always terminate.
3. Termination is good enough because of the following.

Theorem

For every GCR game, if MVI terminates then the output $\widehat{\sigma}$ *is a deterministic positional NE of the game, i.e.:*

$$\forall n : \forall \sigma^n : \forall s_0 : Q^n\left(s_0, \widehat{\sigma}^n, \widehat{\sigma}^{-n}\right) \geq Q^n\left(s_0, \sigma^n, \widehat{\sigma}^{-n}\right).$$

The Multi-Value Iteration Algorithm

Additional Remarks

The Multi-Value Iteration Algorithm

Additional Remarks

1. Hence MVI is already a useful tool:

The Multi-Value Iteration Algorithm

Additional Remarks

1. Hence MVI is already a useful tool: when it terminates, we know that we have obtained a NE of the game.

The Multi-Value Iteration Algorithm

Additional Remarks

1. Hence MVI is already a useful tool: when it terminates, we know that we have obtained a NE of the game.
2. However, further work is required:

The Multi-Value Iteration Algorithm

Additional Remarks

1. Hence MVI is already a useful tool: when it terminates, we know that we have obtained a NE of the game.
2. However, further work is required:
    1. to establish necessary and/or sufficient conditions for termination;

The Multi-Value Iteration Algorithm

Additional Remarks

1. Hence MVI is already a useful tool: when it terminates, we know that we have obtained a NE of the game.
2. However, further work is required:
   1. to establish necessary and/or sufficient conditions for termination;
   2. to find criteria on the particular NE selected.

The Multi-Value Iteration Algorithm

Additional Remarks

1. Hence MVI is already a useful tool: when it terminates, we know that we have obtained a NE of the game.
2. However, further work is required:
   1. to establish necessary and/or sufficient conditions for termination;
   2. to find criteria on the particular NE selected.

The Multi-Value Iteration Algorithm

Additional Remarks

1. Hence MVI is already a useful tool: when it terminates, we know that we have obtained a NE of the game.
2. However, further work is required:
    1. to establish necessary and/or sufficient conditions for termination;
    2. to find criteria on the particular NE selected.

Conjecture

For certain graph families (e.g., trees) there exists a single function from which the update operator is a contraction.

The Multi-Value Iteration Algorithm

Additional Remarks

1. Hence MVI is already a useful tool: when it terminates, we know that we have obtained a NE of the game.
2. However, further work is required:
    1. to establish necessary and/or sufficient conditions for termination;
    2. to find criteria on the particular NE selected.

Conjecture

For certain graph families (e.g., trees) there exists a single function from which the update operator is a contraction.

It cannot be so simple ...

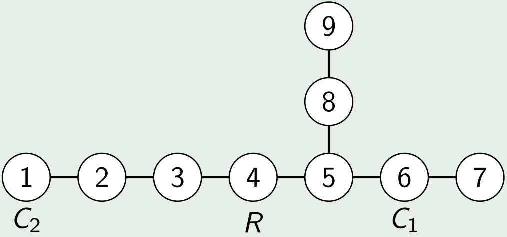
The Multi-Value Iteration Algorithm

Properties of SCAR: an Example

Example

Here minimizing capture time does not yield a NE. Specifically, a Cop may have an incentive to delay capture.

Initial positions as indicated, C_1 has the first move, $\varepsilon < \frac{1}{2}$.



Properties of SCAR: an Example

If the cops use CR-optimal strategies (i.e., go for the robber as fast as possible) the game evolves as follows:

Turn	0	1	2	3	4	5
C_1 **pos.**	6	5	5	5	4	4
C_2 **pos.**	1	1	2	2	2	3
R **pos.**	4	4	4	3	3	3

Properties of SCAR: an Example

If the cops use CR-optimal strategies (i.e., go for the robber as fast as possible) the game evolves as follows:

Turn	0	1	2	3	4	5
C_1 **pos.**	6	5	5	5	4	4
C_2 **pos.**	1	1	2	2	2	3
R **pos.**	4	4	4	3	3	3

The payoffs are

$$Q^1\left(s_0,\widehat{\sigma}^1,\widehat{\sigma}^2,\widehat{\sigma}^3\right)=\varepsilon\gamma^5,$$
$$Q^2\left(s_0,\widehat{\sigma}^1,\widehat{\sigma}^2,\widehat{\sigma}^3\right)=(1-\varepsilon)\,\gamma^5,$$
$$Q^3\left(s_0,\widehat{\sigma}^1,\widehat{\sigma}^2,\widehat{\sigma}^3\right)=-\gamma^5.$$

Now suppose C_2, R use their previous strategies, but C_1 first retreats to 7 and then moves directly towards the robber.

Properties of SCAR: an Example

If the cops use CR-optimal strategies (i.e., go for the robber as fast as possible) the game evolves as follows:

Turn	0	1	2	3	4	5
C_1 **pos.**	6	5	5	5	4	4
C_2 **pos.**	1	1	2	2	2	3
R **pos.**	4	4	4	3	3	3

The payoffs are

$$\begin{aligned} Q^1\left(s_0, \widehat{\sigma}^1, \widehat{\sigma}^2, \widehat{\sigma}^3\right) &= \varepsilon\gamma^5, \\ Q^2\left(s_0, \widehat{\sigma}^1, \widehat{\sigma}^2, \widehat{\sigma}^3\right) &= (1-\varepsilon)\gamma^5, \\ Q^3\left(s_0, \widehat{\sigma}^1, \widehat{\sigma}^2, \widehat{\sigma}^3\right) &= -\gamma^5. \end{aligned}$$

Now suppose C_2, R use their previous strategies, but C_1 first retreats to 7 and then moves directly towards the robber.

Then R can move towards vertex 9 which, increases capture time and results in capture by C_1.

Properties of SCAR: an Example

The game evolves as follows.

Properties of SCAR: an Example

The game evolves as follows.

Turn	0	1	2	3	4	5	6	7	8	9	10	11	12	13
C_1 **pos.**	6	7	7	7	6	6	6	5	5	5	8	8	8	9
C_2 **pos.**	1	1	2	2	2	3	3	3	4	4	4	5	5	5
R **pos.**	4	4	4	5	5	5	8	8	8	9	9	9	9	9

Properties of SCAR: an Example

The game evolves as follows.

Turn	0	1	2	3	4	5	6	7	8	9	10	11	12	13
C_1 **pos.**	6	7	7	7	6	6	6	5	5	5	8	8	8	9
C_2 **pos.**	1	1	2	2	2	3	3	3	4	4	4	5	5	5
R **pos.**	4	4	4	5	5	5	8	8	8	9	9	9	9	9

The payoffs are

Properties of SCAR: an Example

The game evolves as follows.

Turn	0	1	2	3	4	5	6	7	8	9	10	11	12	13
C_1 **pos.**	6	7	7	7	6	6	6	5	5	5	8	8	8	9
C_2 **pos.**	1	1	2	2	2	3	3	3	4	4	4	5	5	5
R **pos.**	4	4	4	5	5	5	8	8	8	9	9	9	9	9

The payoffs are

$$
\begin{aligned}
Q^1\left(s_0,\widetilde{\sigma}^1,\widehat{\sigma}^2,\widehat{\sigma}^3\right) &= (1-\varepsilon)\,\gamma^{13},\\
Q^2\left(s_0,\widetilde{\sigma}^1,\widehat{\sigma}^2,\widehat{\sigma}^3\right) &= \varepsilon\gamma^{13},\\
Q^3\left(s_0,\widetilde{\sigma}^1,\widehat{\sigma}^2,\widehat{\sigma}^3\right) &= -\gamma^{13}.
\end{aligned}
$$

Properties of SCAR: an Example

The game evolves as follows.

Turn	0	1	2	3	4	5	6	7	8	9	10	11	12	13
C_1 **pos.**	6	7	7	7	6	6	6	5	5	5	8	8	8	9
C_2 **pos.**	1	1	2	2	2	3	3	3	4	4	4	5	5	5
R **pos.**	4	4	4	5	5	5	8	8	8	9	9	9	9	9

The payoffs are

$$
\begin{aligned}
Q^1\left(s_0, \widetilde{\sigma}^1, \widehat{\sigma}^2, \widehat{\sigma}^3\right) &= (1-\varepsilon)\,\gamma^{13}, \\
Q^2\left(s_0, \widetilde{\sigma}^1, \widehat{\sigma}^2, \widehat{\sigma}^3\right) &= \varepsilon\gamma^{13}, \\
Q^3\left(s_0, \widetilde{\sigma}^1, \widehat{\sigma}^2, \widehat{\sigma}^3\right) &= -\gamma^{13}.
\end{aligned}
$$

It is easy to see that

$$
\gamma^8 > \frac{\varepsilon}{1-\varepsilon} \Rightarrow Q^1\left(s_0, \widetilde{\sigma}^1, \widehat{\sigma}^2, \widehat{\sigma}^3\right) > Q^1\left(s_0, \widehat{\sigma}^1, \widehat{\sigma}^2, \widehat{\sigma}^3\right).
$$

Properties of SCAR: an Example

The game evolves as follows.

Turn	0	1	2	3	4	5	6	7	8	9	10	11	12	13
C_1 **pos.**	6	7	7	7	6	6	6	5	5	5	8	8	8	9
C_2 **pos.**	1	1	2	2	2	3	3	3	4	4	4	5	5	5
R **pos.**	4	4	4	5	5	5	8	8	8	9	9	9	9	9

The payoffs are

$$Q^1\left(s_0, \widetilde{\sigma}^1, \widehat{\sigma}^2, \widehat{\sigma}^3\right) = (1-\varepsilon)\,\gamma^{13},$$
$$Q^2\left(s_0, \widetilde{\sigma}^1, \widehat{\sigma}^2, \widehat{\sigma}^3\right) = \varepsilon\gamma^{13},$$
$$Q^3\left(s_0, \widetilde{\sigma}^1, \widehat{\sigma}^2, \widehat{\sigma}^3\right) = -\gamma^{13}.$$

It is easy to see that

$$\gamma^8 > \frac{\varepsilon}{1-\varepsilon} \Rightarrow Q^1\left(s_0, \widetilde{\sigma}^1, \widehat{\sigma}^2, \widehat{\sigma}^3\right) > Q^1\left(s_0, \widehat{\sigma}^1, \widehat{\sigma}^2, \widehat{\sigma}^3\right).$$

Hence $\left(\widehat{\sigma}^1, \widehat{\sigma}^2, \widehat{\sigma}^3\right)$ is not a NE.

Properties of SCAR: an Example

The game evolves as follows.

Turn	0	1	2	3	4	5	6	7	8	9	10	11	12	13
C_1 **pos.**	6	7	7	7	6	6	6	5	5	5	8	8	8	9
C_2 **pos.**	1	1	2	2	2	3	3	3	4	4	4	5	5	5
R **pos.**	4	4	4	5	5	5	8	8	8	9	9	9	9	9

The payoffs are

$$\begin{aligned} Q^1\left(s_0,\widetilde{\sigma}^1,\widehat{\sigma}^2,\widehat{\sigma}^3\right) &= (1-\varepsilon)\,\gamma^{13}, \\ Q^2\left(s_0,\widetilde{\sigma}^1,\widehat{\sigma}^2,\widehat{\sigma}^3\right) &= \varepsilon\gamma^{13}, \\ Q^3\left(s_0,\widetilde{\sigma}^1,\widehat{\sigma}^2,\widehat{\sigma}^3\right) &= -\gamma^{13}. \end{aligned}$$

It is easy to see that

$$\gamma^8 > \frac{\varepsilon}{1-\varepsilon} \Rightarrow Q^1\left(s_0,\widetilde{\sigma}^1,\widehat{\sigma}^2,\widehat{\sigma}^3\right) > Q^1\left(s_0,\widehat{\sigma}^1,\widehat{\sigma}^2,\widehat{\sigma}^3\right).$$

Hence $\left(\widehat{\sigma}^1,\widehat{\sigma}^2,\widehat{\sigma}^3\right)$ is not a NE.
In fact, it can be proved that $\left(\widetilde{\sigma}^1,\widehat{\sigma}^2,\widehat{\sigma}^3\right)$ is a NE.

Properties of 3-Player SCAR

Properties of 3-Player SCAR

In what follows

$$\Omega^3 = \left\{ (\gamma, \varepsilon) : \gamma \in (0,1), \varepsilon \in \left[0, \frac{1}{2} \right] \right\}$$

and

$$\widetilde{\Omega}^3 = \left\{ (\gamma, \varepsilon) : \gamma \in (0,1), \varepsilon \in \left[0, \frac{1}{2} \right], \gamma < \frac{\varepsilon}{1-\varepsilon} \right\}$$

Properties of 3-Player SCAR

In what follows

$$\Omega^3 = \left\{ (\gamma, \varepsilon) : \gamma \in (0,1), \varepsilon \in \left[0, \frac{1}{2}\right] \right\}$$

and

$$\widetilde{\Omega}^3 = \left\{ (\gamma, \varepsilon) : \gamma \in (0,1), \varepsilon \in \left[0, \frac{1}{2}\right], \gamma < \frac{\varepsilon}{1-\varepsilon} \right\}$$

Theorem

For any G with $c(G) = 1$ the following holds:

$$\forall (\gamma, \varepsilon) \in \Omega^3, \forall s_0 \in S : \textit{every NE is capturing.}$$

Properties of 3-Player SCAR

Properties of 3-Player SCAR

Theorem

For any G with $c(G) = 2$ the following holds:

$$\forall (\gamma, \varepsilon) \in \Omega^3, \forall s_0 \in S : \textit{there exists a capturing NE.}$$

Properties of 3-Player SCAR

Theorem

For any G with $c(G) = 2$ the following holds:

$$\forall (\gamma, \varepsilon) \in \Omega^3, \forall s_0 \in S : \textit{there exists a capturing NE.}$$

Theorem

For any G with $c(G) = 2$, let $\widehat{\sigma}$ be an optimal strategy profile in the two-cop CR game. Then

$$\forall (\gamma, \varepsilon) \in \widetilde{\Omega}^3, \forall s_0 \in S : \widehat{\sigma} \textit{ is a capturing NE.}$$

Properties of 3-Player SCAR

Theorem

For any G with $c(G) = 2$ the following holds:

$$\forall(\gamma,\varepsilon)\in\Omega^3, \forall s_0\in S : \textit{there exists a capturing NE.}$$

Theorem

For any G with $c(G) = 2$, let $\widehat{\sigma}$ be an optimal strategy profile in the two-cop CR game. Then

$$\forall(\gamma,\varepsilon)\in\widetilde{\Omega}^3, \forall s_0\in S : \widehat{\sigma}\ \textit{is a capturing NE.}$$

Theorem

For any G with $c(G) \geq 2$, the following holds:

$$\forall(\gamma,\varepsilon)\in\Omega^3, \exists s_0\in S : \textit{there exists a non-capturing NE.}$$

Properties of 3-Player SCAR

Properties of 3-Player SCAR

Theorem

For any G with $c(G) \geq 3$ the following holds:

$$\forall (\gamma, \varepsilon) \in \Omega^3, \exists s_0 \in S : \textit{every NE is non-capturing.}$$

Properties of 3-Player SCAR

Theorem

For any G with $c(G) \geq 3$ the following holds:

$$\forall (\gamma, \varepsilon) \in \Omega^3, \exists s_0 \in S : \textit{every NE is non-capturing.}$$

Corollary

Given a graph G:

1. *suppose that for all $(\gamma, \varepsilon) \in \Omega^3$ and $s_0 \in S$, every NE is capturing; then $c(G) = 1$.*
2. *suppose that for all $(\gamma, \varepsilon) \in \Omega^3$ there exists some $s_0 \in S$ such that every NE is non-capturing; then $c(G) \geq 3$.*

Properties of 3-Player SCAR

Properties of 3-Player SCAR

Corollary

G is cop-win iff : for all $(\gamma, \varepsilon) \in \Omega^3$ and $s_0 \in S$, every NE is capturing.

Properties of 3-Player SCAR

Corollary

G is cop-win iff : for all $(\gamma,\varepsilon)\in\Omega^3$ and $s_0\in S$, every NE is capturing.

Properties of N-Player SCAR

Properties of N-Player SCAR

Theorem

For any G with $c(G) = 1$ the following holds:

$$\forall (\gamma, \varepsilon) \in \Omega^N, \forall s_0 \in S : \textit{every NE is capturing.}$$

Properties of N-Player SCAR

Properties of N-Player SCAR

Theorem

For any G with $c(G) \leq N-1$ the following holds:

$$\forall (\gamma, \varepsilon) \in \Omega^N, \forall s_0 \in S : \textit{there exists a capturing NE.}$$

Properties of N-Player SCAR

Theorem

For any G with $c(G) \leq N-1$ the following holds:

$$\forall (\gamma,\varepsilon) \in \Omega^N, \forall s_0 \in S : \text{there exists a capturing NE.}$$

Theorem

For any G with $c(G) \leq N-1$, let $\widehat{\sigma}$ be a strategy profile which is optimal in the $(N-1)$-cop CR game. Then

$$\forall (\gamma,\varepsilon) \in \widetilde{\Omega}^N, \forall s_0 \in S : \widehat{\sigma} \text{ is a capturing NE.}$$

Properties of N-Player SCAR

Theorem

For any G with $c(G) \leq N-1$ the following holds:

$$\forall (\gamma, \varepsilon) \in \Omega^N, \forall s_0 \in S : \text{there exists a capturing NE.}$$

Theorem

For any G with $c(G) \leq N-1$, let $\widehat{\sigma}$ be a strategy profile which is optimal in the $(N-1)$-cop CR game. Then

$$\forall (\gamma, \varepsilon) \in \widetilde{\Omega}^N, \forall s_0 \in S : \widehat{\sigma} \text{ is a capturing NE.}$$

Properties of N-Player SCAR

Properties of N-Player SCAR

Theorem

For any G with $c(G) \geq 2$, the following holds:

$\forall (\gamma, \varepsilon) \in \Omega^N, \exists s_0 \in S$: *there exists a non-capturing NE.*

Properties of N-Player SCAR

Theorem

For any G with $c(G) \geq 2$, the following holds:

$$\forall (\gamma, \varepsilon) \in \Omega^N, \exists s_0 \in S : \textit{there exists a non-capturing NE.}$$

Theorem

For any G with $c(G) \geq N$ the following holds:

$$\forall (\gamma, \varepsilon) \in \Omega^N, \exists s_0 \in S : \textit{every NE is non-capturing.}$$

Properties of N-Player SCAR

Properties of N-Player SCAR

Corollary

Given a graph G*:*

1. *suppose that for all* $(\gamma, \varepsilon) \in \Omega^N$ *and* $s_0 \in S$*, every NE is capturing; then* $c(G) = 1$*.*
2. *suppose that for all* $(\gamma, \varepsilon) \in \Omega^N$ *there exists some* $s_0 \in S$ *such that every NE is non-capturing; then* $c(G) \geq N$*.*

Properties of N-Player SCAR

Corollary

Given a graph G:

1. *suppose that for all $(\gamma, \varepsilon) \in \Omega^N$ and $s_0 \in S$, every NE is capturing; then $c(G) = 1$.*
2. *suppose that for all $(\gamma, \varepsilon) \in \Omega^N$ there exists some $s_0 \in S$ such that every NE is non-capturing; then $c(G) \geq N$.*

Corollary

G is cop-win iff: for all $(\gamma, \varepsilon) \in \Omega^N$ and $s_0 \in S$, every NE is capturing.

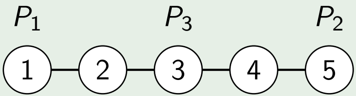
Properties of N-Player SCAR

Example of 3-Player Linear Pursuit

Example

The strategy profile $\overline{\sigma} = \left(\overline{\sigma}^1, \overline{\sigma}^2, \overline{\sigma}^3\right)$ defined as follows is a noncapturing NE.

1. $\overline{\sigma}^1$: P_1 stays in place as long as P_2 does not move; if P_2 moves, P_1 chases him.
2. $\overline{\sigma}^2$: P_2 stays in place as long as P_3 does not move; if P_3 moves, P_2 chases him.
3. $\overline{\sigma}^3$: P_3 stays in place as long as nobody moves; if P_1 moves, P_3 goes towards P_2; if P_2 moves, P_3 goes towards P_1.



Properties of 3-Player Linear Pursuit

Properties of 3-Player Linear Pursuit

Theorem

For every G:

$$\left(\forall s_0 \text{ there exists a capturing NE }\right) \Rightarrow c\left(G\right)=1.$$

Properties of 3-Player Linear Pursuit

Theorem

For every G:

$$(\forall s_0 \textit{ there exists a capturing NE }) \Rightarrow c(G) = 1.$$

Theorem

For every G:

$$c(G) > 1 \Rightarrow (\exists s_0 \textit{ for which we have a noncapturing NE.})$$

Properties of 3-Player Linear Pursuit

Properties of 3-Player Linear Pursuit

Theorem

If $c(G) = 1$ *then every optimal profile* $\widehat{\sigma}$ *of* $\Gamma_2(G, s_0)$ *is a NE of* $\Gamma(G, s_0)$.

Properties of 3-Player Linear Pursuit

Theorem

If $c(G) = 1$ *then every optimal profile* $\widehat{\sigma}$ *of* $\Gamma_2(G, s_0)$ *is a NE of* $\Gamma(G, s_0)$.

Theorem

If G *is a tree then every optimal profile* $\widehat{\sigma}$ *of* $\Gamma_2(G, s_0)$ *is a capturing NE in* $\Gamma(G, s_0)$.

Properties of 3-Player Linear Pursuit

Theorem

If $c(G) = 1$ *then every optimal profile* $\widehat{\sigma}$ *of* $\Gamma_2(G, s_0)$ *is a NE of* $\Gamma(G, s_0)$.

Theorem

If G *is a tree then every optimal profile* $\widehat{\sigma}$ *of* $\Gamma_2(G, s_0)$ *is a capturing NE in* $\Gamma(G, s_0)$.

Corollary

If G *is a tree, then*

$$\forall s_0 \text{ there exists a capturing NE.}$$

Properties of N-player Linear Pursuit

Properties of N-player Linear Pursuit

Theorem

For every G and every $N \geq 3$

$$(\forall s_0 \text{ there exists a capturing NE }) \Rightarrow c(G) = 1.$$

Properties of N-player Linear Pursuit

Theorem

For every G and every $N \geq 3$

$$(\forall s_0 \textit{ there exists a capturing NE }) \Rightarrow c(G) = 1.$$

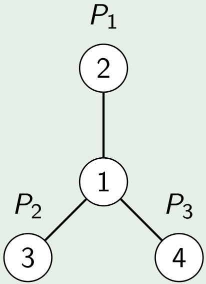
Theorem

For every G and every $N \geq 3$

$$c(G) > 1 \Rightarrow (\exists s_0 \textit{ for which we have a noncapturing NE})$$

Properties of Cyclic Pursuit

Example



Even though the graph is cop-win, the game has only non-capturing NE.

Thank You for your Attention!

Thank You!

Thank You for your Attention!

Thank You!

Questions?

Bibliography

1. A. Berarducci and B. Intrigila. "On the cop number of a graph." *Advances in Applied Mathematics*, 14.4 (1993): 389-403.
2. A. Bonato and G. MacGillivray. "Characterizations and algorithms for generalized Cops and Robbers games." *Contributions to Discrete Mathematics* 12.1 (2017).
3. J. Filar and K. Vrieze. *Competitive Markov decision processes*. 1996.
4. T. Gavenciak, *Hry na grafech (Games on graphs)*, Master's thesis, Department of Applied Mathematics, Charles University, Prague, 2007.
5. G. Hahn and G. MacGillivray. "A note on k-cop, l-robber games on graphs." *Discrete Mathematics*, vol. 306.19-20 (2006): 2492-2497.
6. Ath. Kehagias "Generalized Cops and Robbers: A Multi-player Pursuit Game on Graphs." Dynamic Games and Applications 9.4 (2019): 1076-1099.
7. Ath. Kehagias and G. Konstantinidis. "Selfish cops and passive robber: Qualitative games." Theoretical Computer Science 680 (2017): 25-35.
8. Ath. Kehagias and G, Konstantinidis. "Some Game-Theoretic Remarks on Two-Player Generalized Cops and Robbers Games". *Dynamic Games and Applications*, 2021

Bibliography

9. G. Konstantinidis and Ath. Kehagias. "Simultaneously moving cops and robbers." Theoretical Computer Science 645 (2016): 48-59.
10. G. Konstantinidis and Ath. Kehagias. "Selfish cops and active robber: Multi-player pursuit evasion on graphs." Theoretical Computer Science 780 (2019): 84-102.
11. G. Konstantinidis and Ath Kehagias. "On positionality of trigger strategies: Nash equilibria in SCAR." Theoretical Computer Science 845 (2020): 144-158.