Design of intelligent surveillance systems: a game theoretic case

Nicola Basilico
Department of Computer Science
University of Milan

Outline

- Introduction to Game Theory and solution concepts
  - Game definition
  - Solution concepts: dominance, best response, maxmin, minmax, Nash
  - Leader-Follower games
- Security game with Alarms
  - Signal-Response Game
  - Covering routes

Games

- Formally, a game is defined with a **mechanism** and a **strategy profile**
- Mechanism: the rules of the game (number of players, actions, preferences, outcomes)
- Strategy: describes the behavior of the players in the game
- Solving a game: find a strategy profile that exhibits equilibrium properties (stability)

Normal-form games

A normal-form (strategic) game is defined by:

- Set of players $N = \{1, 2, \ldots, n\}$
- Set of action profiles $A = A_1 \times A_2 \times \cdots \times A_n$
- Set of utility functions $u_i : A \to \mathbb{R}$

Representation: n-dimensional matrix, each element corresponds to an outcome

Examples

Prisoner's dilemma (general-sum game)

	C	NC
C	(2, 2)	(0, 3)
NC	(3, 0)	(1, 1)

Strategy profile, pure

$$\sigma = \{\sigma_1, \sigma_2\}$$

$$\sigma_1 = \begin{cases} C & 0 \\ NC & 1 \end{cases} \quad \sigma_2 = \begin{cases} C & 0 \\ NC & 1 \end{cases}$$

Rock, paper, scissors (zero-sum game)

	Rock	Paper	Scissors
Rock	(0,0)	(-1,1)	(1,-1)
Paper	(1,-1)	(0,0)	(-1,1)
Scissors	(-1,1)	(1,-1)	(0,0)

Strategy profile, mixed

$$\sigma = \{\sigma_1, \sigma_2\}$$

$$\sigma_1 = \begin{cases} R & \frac{1}{3} \\ S & \frac{1}{3} \\ P & \frac{1}{3} \end{cases} \quad \sigma_2 = \begin{cases} R & \frac{1}{3} \\ S & \frac{1}{3} \\ P & \frac{1}{3} \end{cases}$$

Some notation

Expected utility of action **a_i** for player **i**: $\mathrm{E}[u_i(a_i, \sigma_{-i})] = \sum_{a_{-i} \in A_{-i}} u_i(a_i, a_{-i}) \prod_{j \in N_{-i}} \sigma_j(a^j_{-i})$

Expected utility of a mixed strategy: $u_i(\sigma) = \sum_{a \in A} u_i(a) \prod_{j=1}^{n} \sigma_j(a_j)$

Support of a strategy: $S_{\sigma_i} = \{a \mid \sigma_i(a) > 0\}$

Best response for player **i**: σ_i^* such that $u_i(\sigma_i^*, \sigma_{-i}) \geq u_i(\sigma_i, \sigma_{-i}) \; \forall \sigma_i$

Example

	Rock	Paper	Scissors
Rock	(0,0)	(-1,1)	(1,-1)
Paper	(1,-1)	(0,0)	(-1,1)
Scissors	(-1,1)	(1,-1)	(0,0)

$$\sigma_2 = \begin{cases} R & \frac{1}{3} \\ S & \frac{1}{3} \\ P & \frac{1}{3} \end{cases} \qquad \begin{aligned} \mathrm{E}[u_1(R,\sigma_2)] &= 0 \cdot 0.\bar{3} - 1 \cdot 0.\bar{3} + 1 \cdot 0.\bar{3} = 0 \\ \mathrm{E}[u_1(P,\sigma_2)] &= 1 \cdot 0.\bar{3} + 0 \cdot 0.\bar{3} - 1 \cdot 0.\bar{3} = 0 \\ \mathrm{E}[u_1(S,\sigma_2)] &= -1 \cdot 0.\bar{3} + 1 \cdot 0.\bar{3} + 0 \cdot 0.\bar{3} = 0 \end{aligned}$$

$$\sigma_1 = \begin{cases} R & \frac{1}{3} \\ S & \frac{1}{3} \\ P & \frac{1}{3} \end{cases} \qquad \begin{aligned} & u_1(\sigma) = 0.\bar{3} \cdot \mathrm{E}[u_1(R,\sigma_2)] + 0.\bar{3} \cdot \mathrm{E}[u_1(P,\sigma_2)] + 0.\bar{3} \cdot \mathrm{E}[u_1(S,\sigma_2)] = 0 \\ & S_{\sigma_1} = \{R, P, S\} \end{aligned}$$

Solution Concepts

Solving a game: what strategies will be played by **self-interested** agents?

- Non-equilibrium concepts (not stable)
  - Dominant strategies
  - Maxmin / Minmax
- Equilibrium concepts (stable)
  - Nash
  - Leader follower

Dominant Strategies

An agent **i** can safely discard **dominated** actions

An action **a** is dominated if there exists another action **a'** such that

$$u_i(a', \sigma_{-i}) > u_i(a, \sigma_{-i}) \ \forall \sigma_{-i}$$

a' is preferred to a no matter what the opponent does

	C	NC
C	(2, 2)	(0, 3)
NC	(3, 0)	(1, 1)

Dominant Strategies

An agent **i** can safely discard **dominated** actions

An action **a** is dominated if there exists another action **a'** such that

$$u_i(a', \sigma_{-i}) > u_i(a, \sigma_{-i}) \ \forall \sigma_{-i}$$

a' is preferred to a no matter what the opponent does

	C	NC
~~C~~	~~(2, 2)~~	~~(0, 3)~~
NC	(3, 0)	(1, 1)

Dominant Strategies

An agent **i** can safely discard **dominated** actions

An action **a** is dominated if there exists another action **a’** such that

$$u_i(a', \sigma_{-i}) > u_i(a, \sigma_{-i}) \; \forall \sigma_{-i}$$

a’ is preferred to a no matter what the opponent does

	C	NC
C	(2, 2)	(0, 3)
NC	(3, 0)	(1, 1)

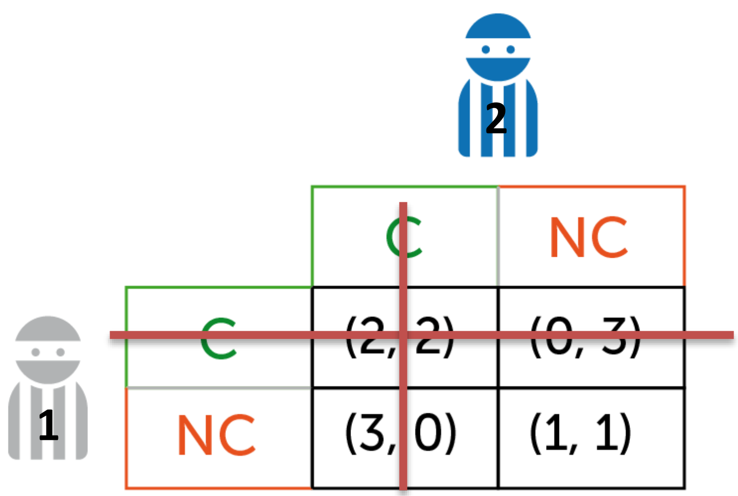
Dominant Strategies

An agent **i** can safely discard **dominated** actions

An action **a** is dominated if there exists another action **a'** such that

$$u_i(a', \sigma_{-i}) > u_i(a, \sigma_{-i}) \ \forall \sigma_{-i}$$

a' is preferred to a no matter what the opponent does



- Very often agents do not have dominant strategies
- Discarding dominated actions can simplify the game

Maxmin and Minmax

Maxmin: seek the best worst case

$$
\begin{aligned}
\max u \quad & \text{s.t.} \\
\sum_{a_1 \in A_1} \sigma_1(a_1) u_1(a_1, a_2) - u \geq 0 \quad & \forall a_2 \in A_2 \\
\sum_{a \in A_1} \sigma_1(a) = 1 \quad & \\
\sigma_1(a_1) \geq 0 \quad & \forall a_1 \in A_1
\end{aligned}
$$

Maxmin and Minmax

Maxmin: seek the best worst case

$$
\begin{aligned}
\max u \quad & \text{s.t.} \\
\sum_{a_1 \in A_1} \sigma_1(a_1) u_1(a_1, a_2) - u \geq 0 \quad & \forall a_2 \in A_2 \\
\sum_{a \in A_1} \sigma_1(a) = 1 \quad & \\
\sigma_1(a_1) \geq 0 \quad & \forall a_1 \in A_1
\end{aligned}
$$

Minmax: seek the worst best case of the opponent

$$
\begin{aligned}
\min u \quad & \text{s.t.} \\
\sum_{a_1 \in A_1} \sigma_1(a_1) u_2(a_1, a_2) - u \leq 0 \quad & \forall a_2 \in A_2 \\
\sum_{a \in A_1} \sigma_1(a) = 1 \quad & \\
\sigma_1(a_1) \geq 0 \quad & \forall a_1 \in A_1
\end{aligned}
$$

Maxmin and Minmax

Maxmin: seek the best worst case

$$
\begin{aligned}
&\max u \quad \text{s.t.}\\
&\sum_{a_1 \in A_1} \sigma_1(a_1) u_1(a_1, a_2) - u \geq 0 \quad \forall a_2 \in A_2\\
&\sum_{a \in A_1} \sigma_1(a) = 1\\
&\sigma_1(a_1) \geq 0 \quad \forall a_1 \in A_1
\end{aligned}
$$

Minmax: seek the worst best case of the opponent

$$
\begin{aligned}
&\min u \quad \text{s.t.}\\
&\sum_{a_1 \in A_1} \sigma_1(a_1) u_2(a_1, a_2) - u \leq 0 \quad \forall a_2 \in A_2\\
&\sum_{a \in A_1} \sigma_1(a) = 1\\
&\sigma_1(a_1) \geq 0 \quad \forall a_1 \in A_1
\end{aligned}
$$

Maxmin is a best response to the opponent's Minmax strategy

Maxmin and Minmax

$$\begin{aligned} \max u \quad & \text{s.t.} \\ \sum_{a_1 \in A_1} \sigma_1(a_1) u_1(a_1, a_2) - u \geq 0 \quad & \forall a_2 \in A_2 \\ \sum_{a \in A_1} \sigma_1(a) = 1 \quad & \\ \sigma_1(a_1) \geq 0 \quad & \forall a_1 \in A_1 \end{aligned}$$

$$\begin{aligned} \min u \quad & \text{s.t.} \\ \sum_{a_1 \in A_1} \sigma_1(a_1) u_2(a_1, a_2) - u \leq 0 \quad & \forall a_2 \in A_2 \\ \sum_{a \in A_1} \sigma_1(a) = 1 \quad & \\ \sigma_1(a_1) \geq 0 \quad & \forall a_1 \in A_1 \end{aligned}$$

- Due to strong duality, in zero-sum games Maxmin and Minmax strategies are the same: they yield the same expected utility **v**
- *In any Nash Equilibrium of a finite, two-player, zero-sum game each player receives a utility of* ***v*** *[von Neumann, 1928]*

Nash Equilibrium

$$\sigma = \{\sigma_1, \sigma_2\} \text{ is a NE if } \sigma_1 \text{ is a best response to } \sigma_2 \text{ and vice versa}$$

Computing NE

- Zero-sum games: can be done efficiently with a linear program [von Neumann, 1920]

- General-sum games: no linear programming formulation is possible

- With two agents:
  - Linear complementarity programming [Lemke and Howson, 1964]
  - Mixed integer linear program (MILP) [Sandholm, Giplin, and Conitzer, 2005]
  - **Multiple linear programs (an exponential number in the worst case) [Porter, Nudelman, and Shoham, 2004]**

- With more than two agents?
  - Non-linear complementarity programming
  - Other methods

- Complexity:
  - The problem is in NP
  - It is not NP-Complete unless P=NP, but complete w.r.t. PPAD ("Polynomial Parity Arguments on Directed graphs" which is contained in NP and contains P) [Papadimitrou, 1991]
  - Commonly believed that no efficient algorithm exists

Searching for a NE

- Suppose that an oracle tells us that at the NE $\sigma^* = \{\sigma_1^*, \sigma_2^*\}$
  $S_{\sigma_1} = \{a_x, a_y\}$, $S_{\sigma_2} = \{b_u, b_w, b_z\}$

- We know which actions will be played with non-null probability at the equilibrium, can we find the equilibrium?

Searching for a NE

- Suppose that an oracle tells us that at the NE $\sigma^* = \{\sigma_1^*, \sigma_2^*\}$
  $S_{\sigma_1} = \{a_x, a_y\}$, $S_{\sigma_2} = \{b_u, b_w, b_z\}$

- We know which actions will be played with non-null probability at the equilibrium, can we find the equilibrium?

- At the equilibrium, each action played by **i** with non-null probability should provide the same expected utility, say $\mathbf{v_i}$. In other words, the player should be indifferent among all of them.

- On the other side, the actions played with null probability should provide an expected utility lower than $\mathbf{v_i}$

Searching for a NE

- We can write the following feasibility linear program:

$$v_i = \sum_{a_j \in A_j} \sigma_j(a_j) u_i(a_i, a_j) \quad a_i \in S_{\sigma_i}, i, j \in \{1,2\}, i \neq j$$

Expected utility at the equilibrium

$$v_i \geq \sum_{a_j \in A_j} \sigma_j(a_j) u_i(a_i, a_j) \quad \forall a_i \notin S_{\sigma_i}, i, j \in \{1,2\}, i \neq j$$

Expected utility outside S

$$\sigma_i(a) > 0 \quad a \in S_{\sigma_i}, i \in \{1,2\}$$

$$\sigma_i(a) = 0 \quad a \notin S_{\sigma_i}, i \in \{1,2\}$$

Positive probability in the support

$$\sum_{a \in A_i} \sigma_i(a) = 1 \quad \forall i \in \{1,2\}$$

Null probability outside the support

- If we knew the supports, we could easily find the equilibrium ☺

- But we don’t know the supports ☹

Searching for a NE

- Simple search procedure:

Choose two supports $S_{\sigma_1}, S_{\sigma_2}$

Is the following LP feasible?

$$
\begin{aligned}
v_i &= \sum_{a_j \in A_j} \sigma_j(a_j) u_i(a_i, a_j) && a_i \in S_{\sigma_i}, i, j \in \{1,2\}, i \neq j \\
v_i &\geq \sum_{a_j \in A_j} \sigma_j(a_j) u_i(a_i, a_j) && \forall a_i \notin S_{\sigma_i}, i, j \in \{1,2\}, i \neq j \\
\sigma_i(a) &> 0 && a \in S_{\sigma_i}, i \in \{1,2\} \\
\sigma_i(a) &= 0 && a \notin S_{\sigma_i}, i \in \{1,2\} \\
\sum_{a \in A_i} \sigma_i(a) &= 1 && \forall i \in \{1,2\}
\end{aligned}
$$

no

yes

NE ☺

Searching for a NE

- Simple search procedure: in the worst case $o(2^n)$

- In practice it achieves good performance, search can be driven with heuristics:
  - Do not include dominated actions
  - Prefer balanced profiles
  - Prefer small supports

- We can easily embed the support in decision variables (n binary variables, single MILP formulation)

Leader-Follower Games

- Leader follower games (a.k.a. Stackelberg games) have a different mechanism
  - A player, denoted as *Leader*, can **commit** to a strategy before playing
  - The other player, denoted as *Follower*, acts as a best responder

- The mechanism entails some kind of communications between players beforehand, where the Leader announces its strategy

- *Notice that, declaring a strategy is different from declaring an action!*
- *Notice that, the follower is a mere best responder!*

Example

	C	D
A	(5,1)	(1,0)
B	(6,2)	(-1,5)

- Let's suppose that, before the game begins, **L** makes the following announcement:

$$\sigma_L = \begin{cases} A & \frac{3}{4} \\ B & \frac{1}{4} \end{cases}$$

Example

	C	D
A	(5,1)	(1,0)
B	(6,2)	(-1,5)

- Let's suppose that, before the game begins, **L** makes the following announcement:

$$\sigma_L = \begin{cases} A & \frac{3}{4} \\ B & \frac{1}{4} \end{cases}$$

If I play C, then $\mathrm{E}[u_F(\sigma_L, C)] = \frac{3}{4} \cdot 1 + \frac{1}{4} \cdot 2 = \frac{5}{4}$

If I play D, then $\mathrm{E}[u_F(\sigma_L, D)] = \frac{3}{4} \cdot 0 + \frac{1}{4} \cdot 5 = \frac{1}{4}$

Example

	C	D
A	(5,1)	(1,0)
B	(6,2)	(-1,5)

- Let's suppose that, before the game begins, **L** makes the following announcement:

$$\sigma_L = \begin{cases} A & \frac{3}{4} \\ B & \frac{1}{4} \end{cases}$$

If I play C, then $E[u_F(\sigma_L, C)] = \frac{3}{4} \cdot 1 + \frac{1}{4} \cdot 2 = \frac{5}{4}$

If I play D, then $E[u_F(\sigma_L, D)] = \frac{3}{4} \cdot 0 + \frac{1}{4} \cdot 5 = \frac{1}{4}$

I will play C

Example

	C	D
A	(5,1)	(1,0)
B	(6,2)	(-1,5)

If I announce $\sigma_L = \begin{cases} A & \frac{3}{4} \\ B & \frac{1}{4} \end{cases}$ the follower will play C

and I will get $\mathrm{E}[u_L(\sigma_L, C)] = \frac{3}{4} \cdot 5 + \frac{1}{4} \cdot 6 = \frac{21}{4}$

Example

F

L	C	D
A	(5,1)	(1,0)
B	(6,2)	(-1,5)

If I announce $\sigma_L = \begin{cases} A & \frac{3}{4} \\ B & \frac{1}{4} \end{cases}$ the follower will play C
and I will get $\mathrm{E}[u_L(\sigma_L, C)] = \frac{3}{4}\cdot 5 + \frac{1}{4}\cdot 6 = \frac{21}{4}$

L

The best σ_L^* to announce is ?

Leader follower equilibrium (LFE)

L

Example

	C	D
A	(5,1)	(1,0)
B	(6,2)	(-1,5)

If I announce $\sigma_L = \begin{cases} A & \frac{3}{4} \\ B & \frac{1}{4} \end{cases}$ the follower will play C

and I will get $\mathrm{E}[u_L(\sigma_L, C)] = \frac{3}{4} \cdot 5 + \frac{1}{4} \cdot 6 = \frac{21}{4}$

The best σ_L^* to announce is ?

Leader follower equilibrium (LFE)

Two important properties:

1. The follower does **not** randomize: it chooses the action that maximizes its expected utility. *If indifferent between one or more actions, it will break ties in favor of the leader (compliant follower).*
2. LFE is not worse than any NE (the leader can always announce a NE)

Computing a LFE

Idea:

1. For each action **b** of the Follower:
   - Find the best commitment C(**b**) to announce, given that **b** will be the action played by **F**

2. Select the best C(**b**)

Computing a LFE

Idea:

1. For each action **b** of the Follower:
   – Find the best commitment C(**b**) to announce, given that **b** will be the action played by **F**

2. Select the best C(**b**)

Step 1

$$\max \sum_{a \in A_L} \sigma_L(a) u_L(a,b) \quad \text{s.t.}$$

$$\sum_{a \in A_L} \sigma_L(a) u_F(a,b) \geq \sum_{a \in A_L} \sigma_L(a) u_F(a,b') \quad \forall b' \in A_F$$

$$\sum_{a \in A_L} \sigma_L(a) = 1$$

$$\sigma_L(a) \geq 0 \quad \forall a \in A_L$$

$$\sigma^*_{L,b}$$

Computing a LFE

Step 2:

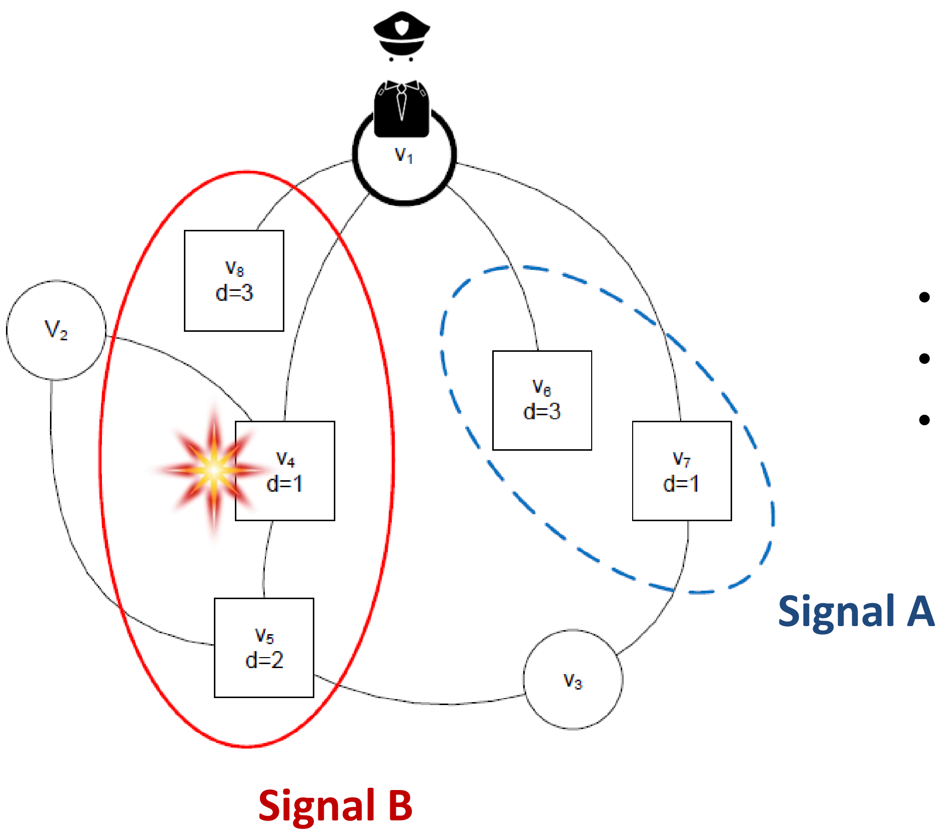
$$\sigma_L^* = \sigma_{L,b^*}$$

$$b^* = \arg\max_{b \in A_F} \sum_{a \in A_L} \sigma_{L,b}(a) u_L(a,b)$$

- We need to solve a LP n times, where n is the number of actions for the Follower

>>> Security Games in the presence of an alarm system

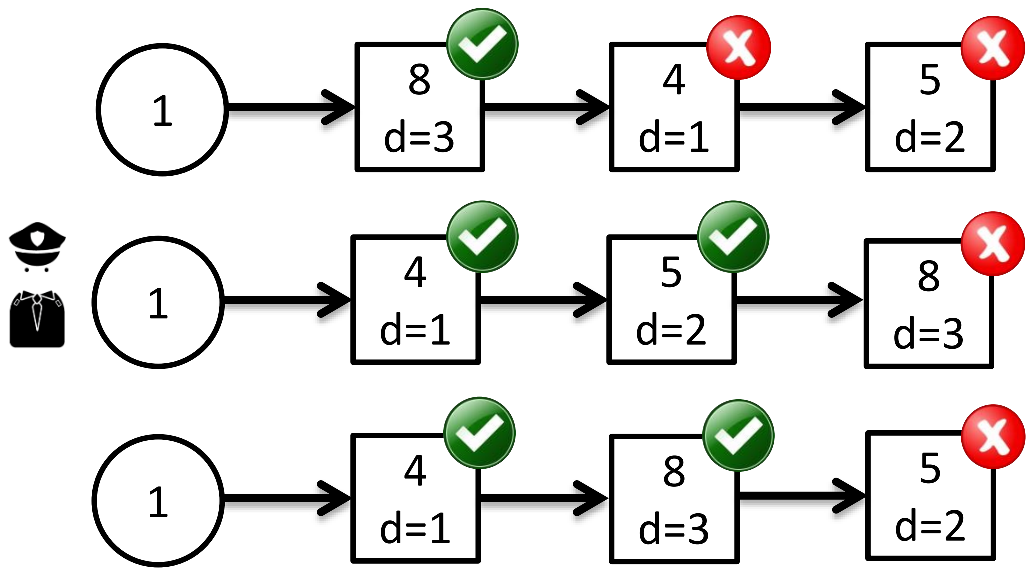
The Alarm System



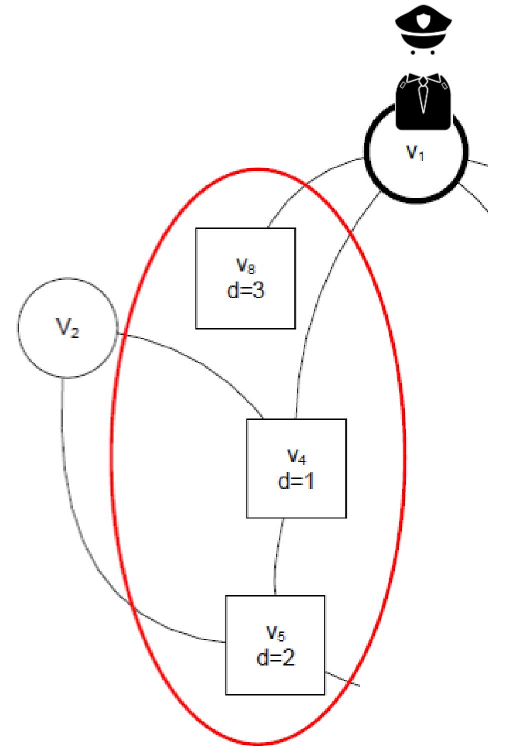
- The Defender is in 1
- The Attacker attacks 4
- The Alarm system generates with prob. 1 **signal B**

The Alarm System

- Upon receiving the signal, the Defender knows that the Attacker is in 8, 4, or 5
- In principle, it should check each target no later than d(t)



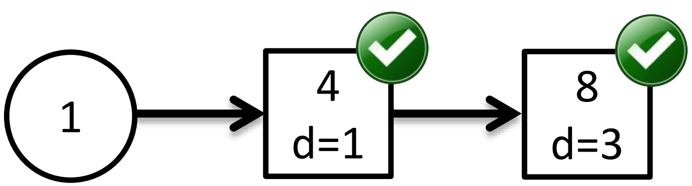
Covering routes



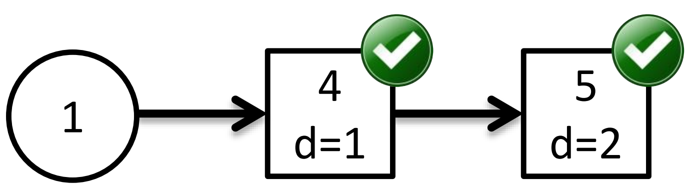
The Alarm System

- Covering routes: a permutation of targets which specifies the order of first visits (covering shortest paths) such that each target is first-visited before its deadline

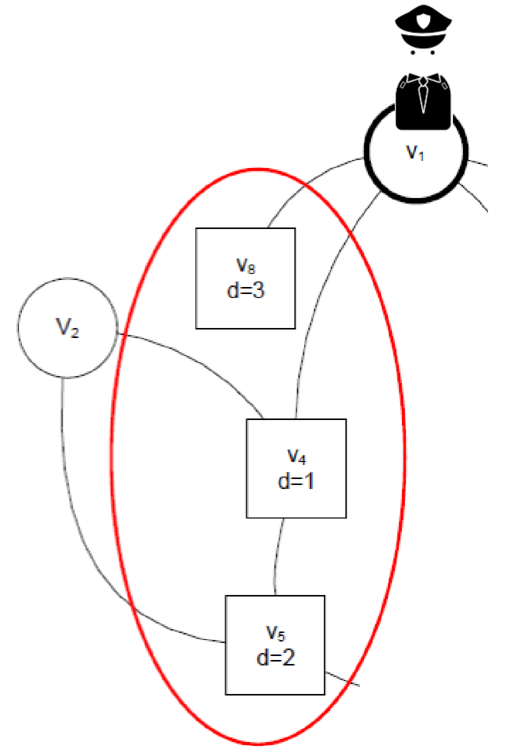
- Example



Covering route: **<4,8>**

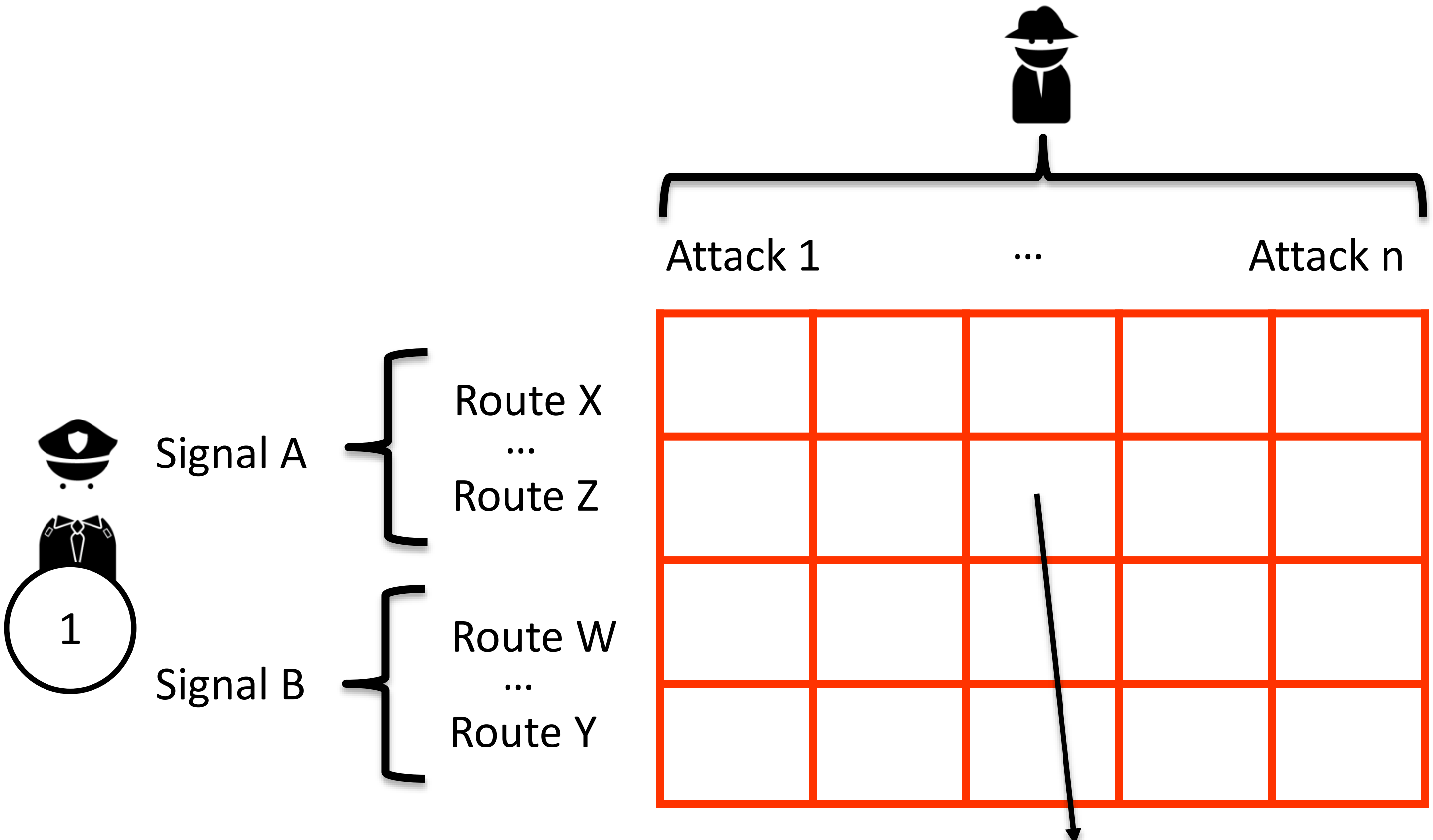


Covering route: **<4,5>**



The Signal Response Game

- We can formulate the game in strategic (normal form), for vertex 1

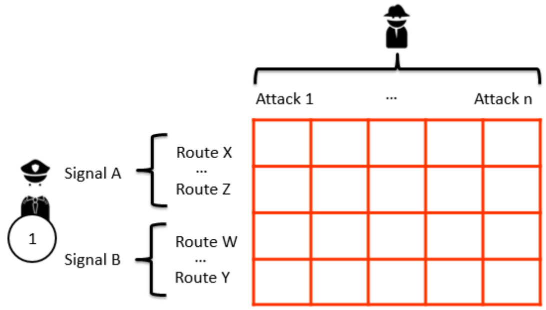


$$U(\{r,t\}) = \begin{cases} (1,0) & \text{if } t \in T(r) \\ (1-\pi(t), \pi(t)) & \text{otherwise.} \end{cases}$$

The Signal Response Game

Solving the SRG, Minmax (NE):

- T is the set of targets, S is the set of signals, R is the set of routes, p(s|t) is the probability that signal **s** is issued when target **t** is attacked



$$\min \quad u \qquad \text{s.t.}$$

$$\sum_{s \in S(t)} p(s \mid t) \sum_{r \in R_{v,s}} \sigma_{\mathcal{D},s}(r) U_{\mathcal{A}}(\{r,t\}) \leq u \qquad \forall t \in T$$

$$\sum_{r \in R_{v,s}} \sigma_{\mathcal{D},s}(r) = 1 \qquad \forall s \in S$$

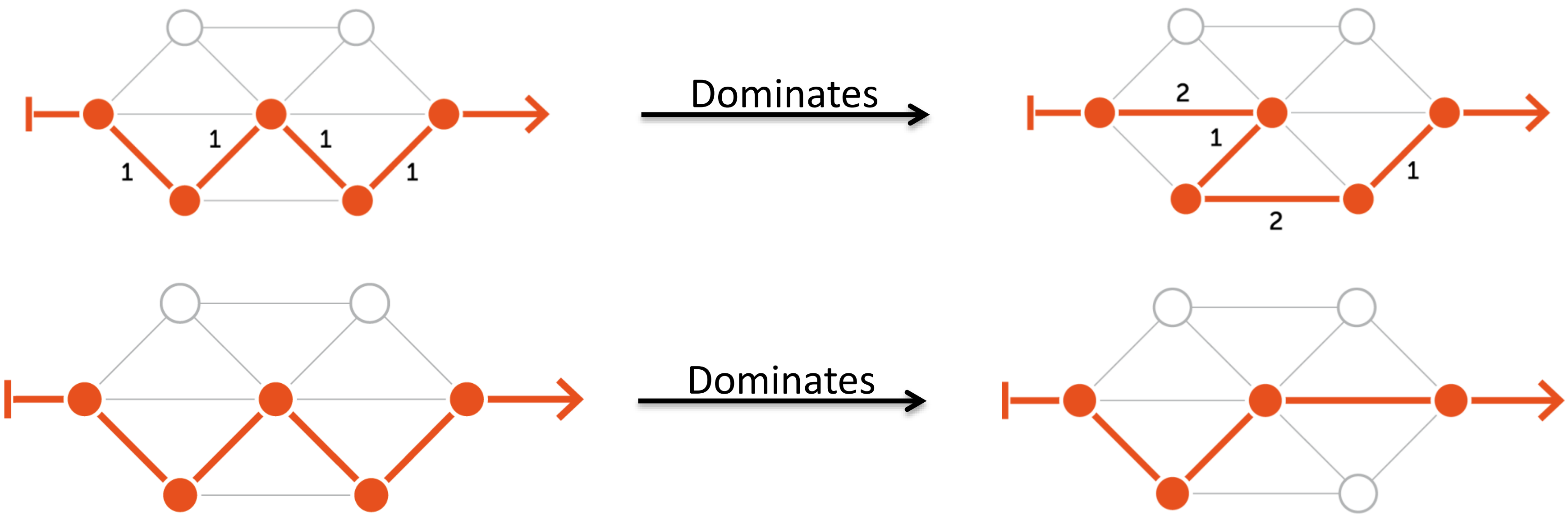
$$\sigma_{\mathcal{D},s}(r) \geq 0 \quad \forall r \in R_{v,s}, s \in S$$

- Repeat this for each starting vertex **v**

Building the Game

- The number of covering routes is, in the worst case, prohibitive: $O(n^n)$ (all the permutations for all the subsets of targets)

- Should we compute all of them? No, some covering routes will never be played



- Even if we remove dominated covering routes, their number is still very large

Building the Game

- Idea: can we consider **covering sets** instead?

From $\langle t_1, t_2, t_3 \rangle$ to $\{t_1, t_2, t_3\}$

- Covering sets are in the worst case: $O(2^n)$ (still exponential but much better than before)

- Problem: we still need routes operatively!

- Solution: we find covering sets and then we try to reconstruct routes

Building the Game

INSTANCE: a covering set that admits at least a covering route

QUESTION: find one covering route

This problem is not only NP-Hard, but also *locally* NP-Hard: a solution for a *very similar* instance is of no use. ☹☹

Building the Game

- Idea: simultaneously build covering sets and the shortest associated covering route
- Dynamic programming inspired algorithm: we can compute all the covering routes in $O(2^n)$!

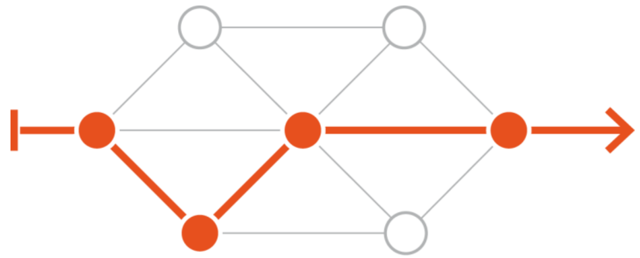
Algorithm 1 ComputeCovSets (Basic)

$\forall t \in T, k \in \{2, \ldots, |T|\}, C_t^1 = \{t\}, C_t^k = \emptyset$
$\forall t \in T, c(\{t\}) = \omega^*_{v,t}, c(\emptyset) = \infty$
for all $k \in \{2 \ldots |T|\}$ **do**
  for all $t \in T$ **do**
    for all $Q_t^{k-1} \in C_t^{k-1}$ **do**
      $Q^+ = \{f \in T \setminus Q_t^{k-1} \mid c(Q_t^{k-1}) + \omega^*_{t,f} \leq d(f)\}$
      for all $f \in Q^+$ **do**
        $Q_f^k = Q_t^{k-1} \cup \{f\}$
        $U = Search(Q_f^k, C_f^k)$
        if $c(U) > c(Q_t^{k-1}) + \omega^*_{t,f}$ **then**
          $C_f^k = C_f^k \cup \{Q_f^k\}$
          $c(Q_f^k) = c(Q_t^{k-1}) + \omega^*_{t,f}$
        end if
      end for
    end for
  end for
end for

Is this the best we can do?
If we find a better algorithm we could build an algorithm for Hamiltionan Path which would outperform the best algorithm known in literature (for general graphs).

Algorithm

- Idea: simultaneously build covering sets and the shortest associated covering route



Covering set: C

Covering route: r

Terminal vertex: t

Q_t^k Covering set with k target whose shortest covering route ends in t

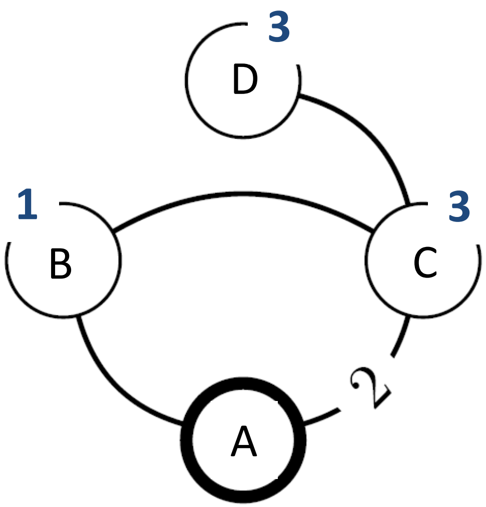
$c(Q_t^k)$ Cost of the associated shortest covering route

$$\text{if } c(Q_t^{k-1}) + \omega^*_{t,f} \leq d(f), \text{ then we have } Q_f^k = Q_t^{k-1} \cup \{f\}$$

Shortest path between t and f

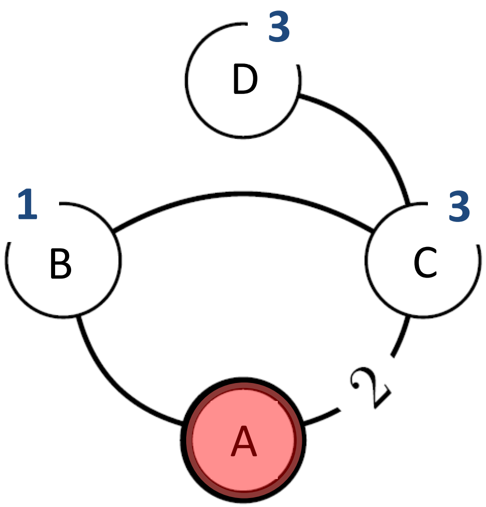
Algorithm

- Example



Algorithm

- Example

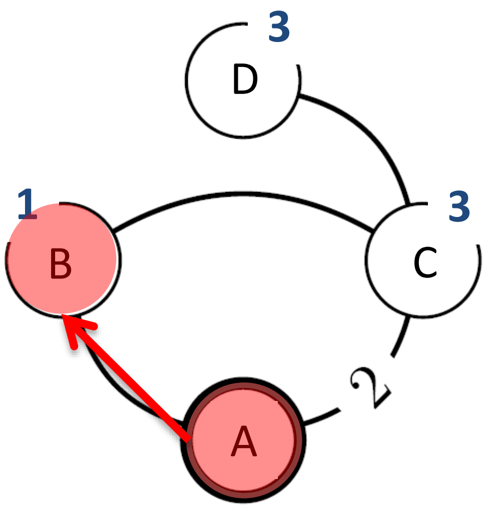


k=1

<{A}->A, 0>

Algorithm

- Example



k=1

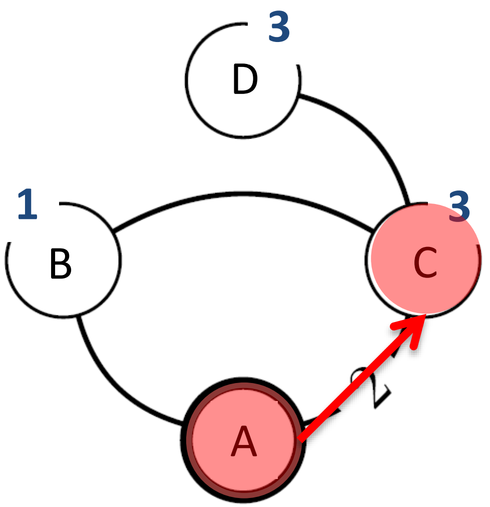
<{A}->A, 0>

k=2

<{A,B}->B, 1>

Algorithm

- Example



k=1

<{A}->A, 0>

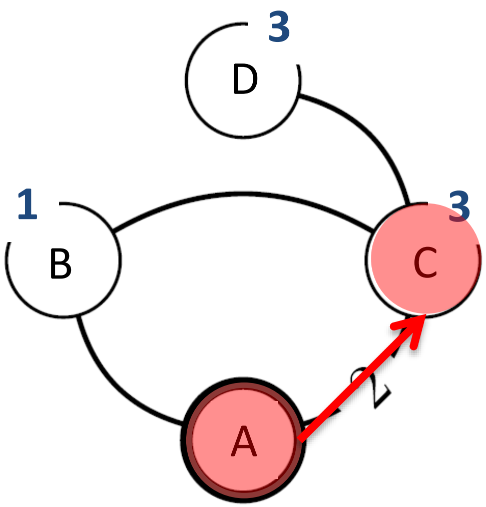
k=2

<{A,B}->B, 1>

<{A,C}->C, 2>

Algorithm

- Example



k=1

<{A}->A, 0>

dominated

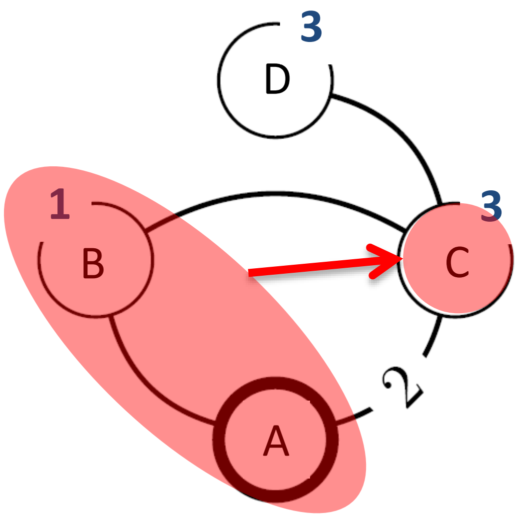
k=2

<{A,B}->B, 1>

<{A,C}->C, 2>

Algorithm

- Example



k=1

<{A}->A, 0>

dominated

k=2

<{A,B}->B, 1>

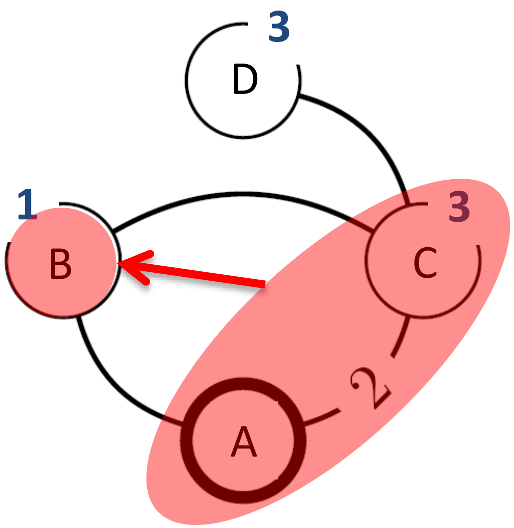
<{A,C}->C, 2>

k=3

<{A,B,C}->C, 2>

Algorithm

- Example



k=1

<{A}->A, 0>

dominated

k=2

<{A,B}->B, 1>

<{A,C}->C, 2>

k=3

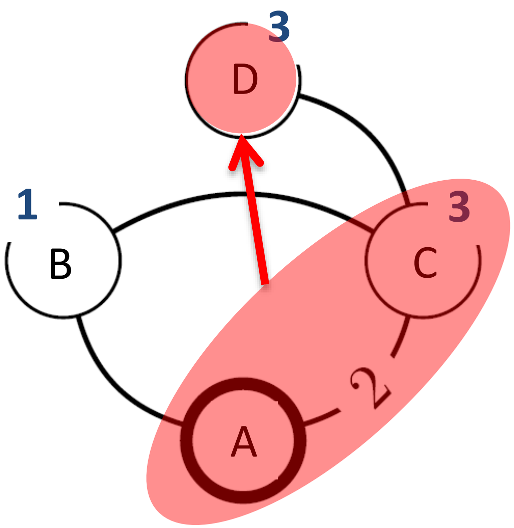
unfeasible

<{A,B,C}->B, 3>

<{A,B,C}->C, 2>

Algorithm

- Example



k=1

<{A}->A, 0>

dominated

k=2

<{A,B}->B, 1>

<{A,C}->C, 2>

k=3

unfeasible

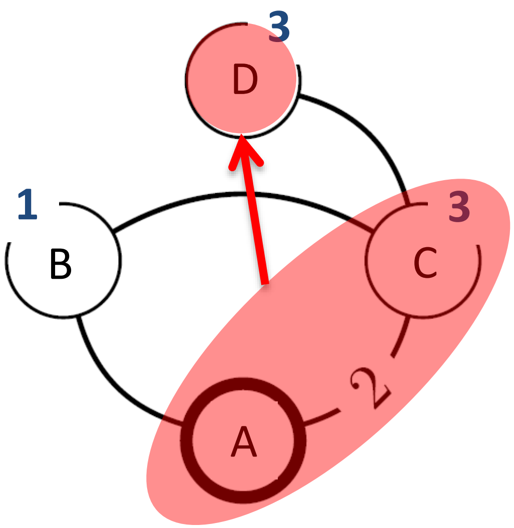
<{A,B,C}->B, 3>

<{A,B,C}->C, 2>

<{A,C,D}->D, 3>

Algorithm

- Example



k=1

<{A}->A, 0>

dominated

k=2

<{A,B}->B, 1>

<{A,C}->C, 2>

dominated

k=3

unfeasible

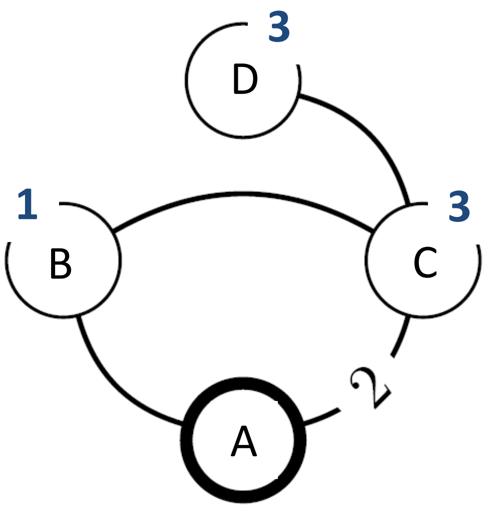
<{A,B,C}->B, 3>

<{A,B,C}->C, 2>

<{A,C,D}->D, 3>

Algorithm

- Example



k=1

<{A}->A, 0>

dominated

k=2

<{A,B}->B, 1>

<{A,C}->C, 2>

dominated

k=3

unfeasible

<{A,B,C}->B, 3>

<{A,B,C}->C, 2>

<{A,C,D}->D, 3>

k=4? All unfeasible

Building the Game (some numbers)

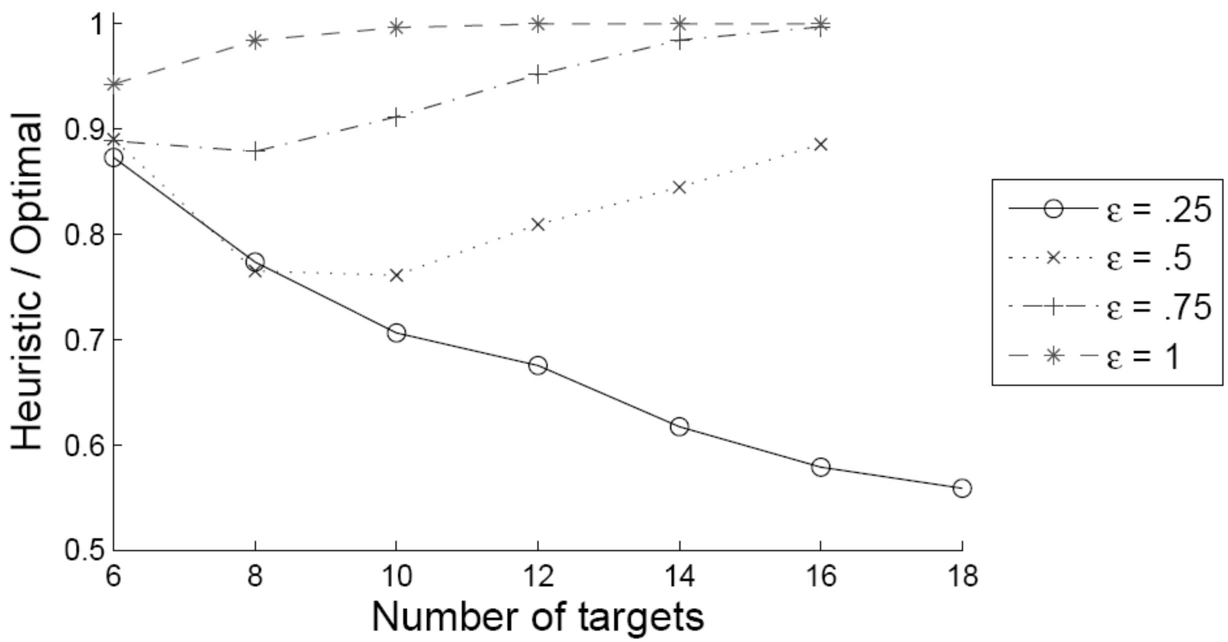
		$\|T\|$						
		6	8	10	12	14	16	18
ε	.25	0,07	0,34	1,91	11,54	82,26	439,92	**4068,8**
	.5	0,07	0,38	4,04	53,14	536,7	**4545,4**	≥ 5000
	.75	0,09	0,96	11,99	114,3	**935,74**	≥ 5000	≥ 5000
	1	0,14	1,86	17,46	143,05	**1073,**	≥ 5000	≥ 5000

- The edge density is a critical parameter. The more dense the graph, the more difficult to build the game.

		$\|T(s)\|$		
		5	10	15
m	2	-	17,83	**510,61**
	3	-	33	**769,3**
	4	0,55	35,35	**1066,76**
	5	0,72	52,43	**1373,32**

Building the Game (some numbers)

- Comparison with an heuristic sub-optimal algorithm.



- Good news: the heuristic method seems to perform better where we the exact algorithm requires the highest computational effort

Open Problems

- Detection errors (false positive, false negatives) , can they be exploited by an attacker?
- Approximability: very unlikely, trying to prove non-approximability (APX-Hardness)
- Study Complexity of particular classes of graphs (trees, grids, etc…)
- Attackers with limited rationality
- Attackers with limited observation capabilities
- …

Available Thesis

- Develop an interactive game where the model can be tested under real conditions (e.g., limited rationality, errors, etc …)
- Try to derive opponent models from human-players behavior (how a real human would deal with the problem of attacking an infrastructure?)
- Model extensions to include more realistic aspects, e.g., allowing false positives and false negatives in the alarm system
- Model scalings: multi-defender, multi-attacker

Available Thesis

- Develop an interactive game where the model can be tested under real conditions (e.g., limited rationality, errors, etc …)

- Try to derive opponent models from human-players behavior (how a real human would deal with the problem of attacking an infrastructure?)

- Model extensions to include more realistic aspects, e.g., allowing false positives and false negatives in the alarm system

- Model scalings: multi-defender, multi-attacker

References

- Nicola Basilico and Nicola Gatti. "Strategic guard placement for optimal response toalarms in security games." *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

- Chao Zhang, et al. "Defending Against Opportunistic Criminals: New Game-Theoretic Frameworks and Algorithms." *Decision and Game Theory for Security*. Springer International Publishing, 2014. 3-22.

- Bo An et al. "Guards and protect: Next generation applications of security games." *ACM SIGecom Exchanges* 10.1 (2011): 31-34.

- Nicola Basilico, Nicola Gatti, and Francesco Amigoni. "Leader-follower strategies for robotic patrolling in environments with arbitrary topologies." *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

References (extra)

- Steve Alpern, Alec Morton, and Katerina Papadaki. "Patrolling games." *Operations research* 59.5 (2011): 1246-1257.

- Dmytro Korzhyk et al. "Stackelberg vs. Nash in Security Games: An Extended Investigation of Interchangeability, Equivalence, and Uniqueness." *J. Artif. Intell. Res.(JAIR)* 41 (2011): 297-327.

- Nicola Basilico, Nicola Gatti, and Francesco Amigoni. "Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder." *Artificial intelligence* 184 (2012): 78-123.

- Yoav Shoham, and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.