TEL AVIV UNIVERSITY
Pursuing the Unknown

Multi-payoff Cyber-Security Games

Amiram Moshaiov

School of Mech. Eng.
&
Sagol School of Neuroscience

PhD Student: Erella Matalon-Eisenstadt
MSc Student: Roi Chananel

TEL AVIV UNIVERSITY
Pursuing the Unknown

Intro. to our
Computational Intelligence
Research Group

Currently: 8 PhD & 3 MSc Students

Main Research Topics

- Multi-objective Optimization and Exploration
  - Multi-Concept Optimization
- Multi-objective Games
- Multi-criteria Decision Analysis
- Multi-objective Neuro-Evolution
- Multi-objective Neuro-Fuzzy Systems
- Multi-objective Genetic Transfer Learning

Outline

1. Motivation & Background
2. Problem description
3. Introduction to rationalizability
4. Methodology and solution approach
5. Cyber-security example
6. Algorithms and Results
7. Conclusions & future work

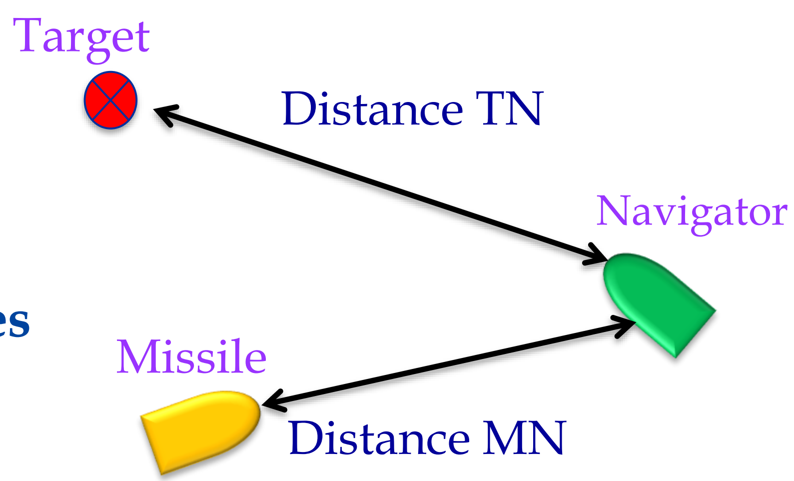
Motivation

- Multi-Objective Games (MOGs)
  - Games with self-conflicting objectives
  - Introduced by Blackwell and by Shapley (1956-9)
- Examples of application areas of MOGs:
  - Defense: (Aerial, Marine, Ground, Cyber)
    - Minimize time-to-capture & Minimize risk of casualties
  - Business, Economics, OR
    - Minimize working hours & Maximize profits

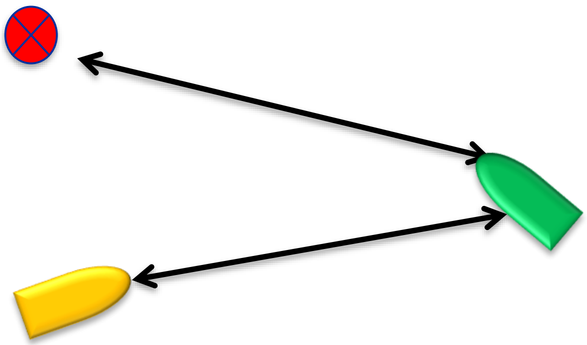
- Motivation in a nutshell:
  - Usefulness of MOG models
  - Deficiencies of existing solution approaches
  - Scientific curiosity (inspired by Pareto-optimality)

MOGs vs. SOGs
Reach & Avoid Bi-objective Game

- Combination of 2 pursuit-evasion games
- Navigator's objectives:
  - Maximize the distance MN
  - Minimize the distance TN
  - These are self-conflicting objectives
- T-M Coalition's objectives:
  - Opposite to those of the Navigator
- Question: Is it a zero-sum game?
- Answer: Yes and No ☺
  - Yes, per each component of the payoff vector
  - No, when the opponent's preference of objectives is not the same



Deficiencies of A-priori Scalarization

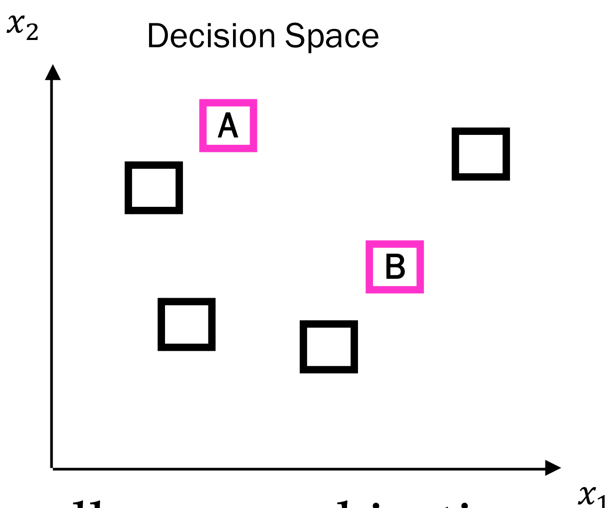


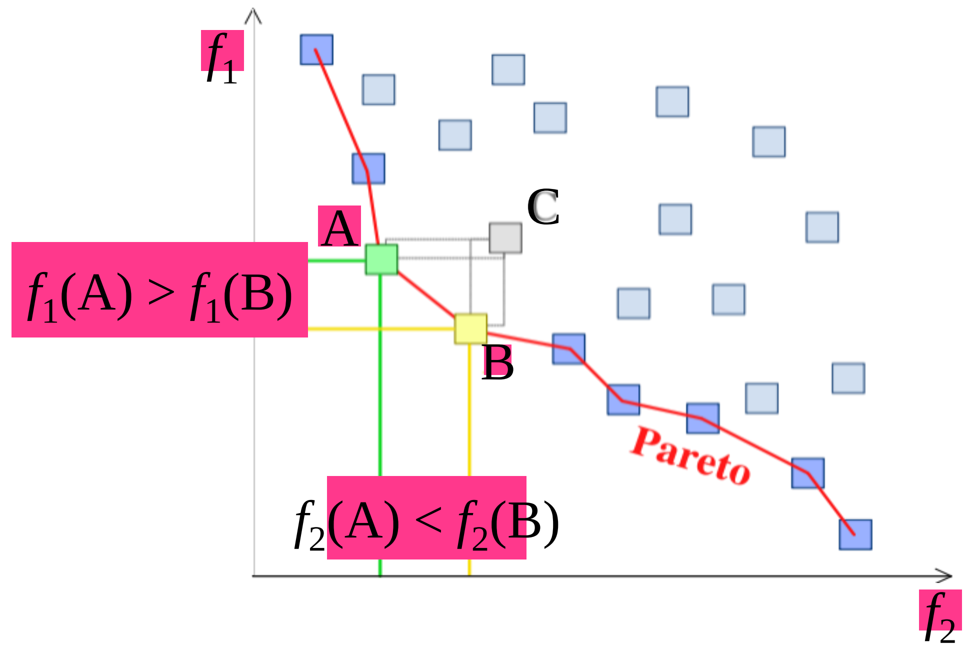
- Most studies on MOGs use a utility function
  - A-priori articulation of objective preferences
  - Transforms a MOG into a surrogate SOG
- Disadvantages of the traditional utility fn. approaches:
  - Subjective and hard to rationalize
  - Do not reveal the involved trade-offs
  - May ignore potential solutions in concave sets of payoff vectors

Can we explore alternative strategies without a-priori declaration of objective preferences?

Pareto-based Multi-Objective Optimization

- A performance-vector based approach
- A solution is evaluated based on more than one objective
- Domination relation is used





- Usually some objectives are contradicting
- Namely, Pareto-optimal set and front exist
  - It reveals the performance tradeoffs
- Posteriori selection of preferred solution
  - Multi-criteria decision-making

From Pareto-optimality to Solving MOGs

- Inspired by Pareto-based Optimization
  - Yet, much more complicated due to the multiplicity of sides
- A novel type of solution approach to MOGs
  - MOGs with undecided objective preferences
- As in Pareto-based one-sided optimization:
  - Two stage solution approach
  - Trade-offs to be revealed before strategy selection
- From inspiration to formulation – a non-trivial task!

Outline

1. Motivation & Background
2. Problem description
3. Introduction to rationalizability
4. Methodology and solution approach
5. Cyber-security example
6. Algorithms and Results
7. Conclusions & future work

The Considered Game:
MOG with undecided objective preferences

THE GAME FEATURES:

Zero-sum game (component-wise):

One player's gain is the other player's loss

Non cooperative:

No agreement is made between the players

Single act:

Both players choose one strategy only once

Imperfect information:

The player does not know what is the chosen action of the other players

Undecided obj. preferences → Incomplete information

Outline

1. Motivation & application areas
2. Background
3. Problem description
4. Introduction to rationalizability
5. Methodology and solution approach
6. Cyber-security example
7. Algorithms and Results
8. Conclusions & future work

Rationalizability Solution Concept for SOGs

- Introduced by Bernheim & by Pearce (1984)
- There is no single optimal strategy
- Common knowledge of rationality
- The set of rationalizable strategies in SOGs is:
  - The remaining set after iterative elimination of strictly dominated strategies

Demonstration of Rationalizability in a zero-sum SOG

The order of elimination is not important

minimizer

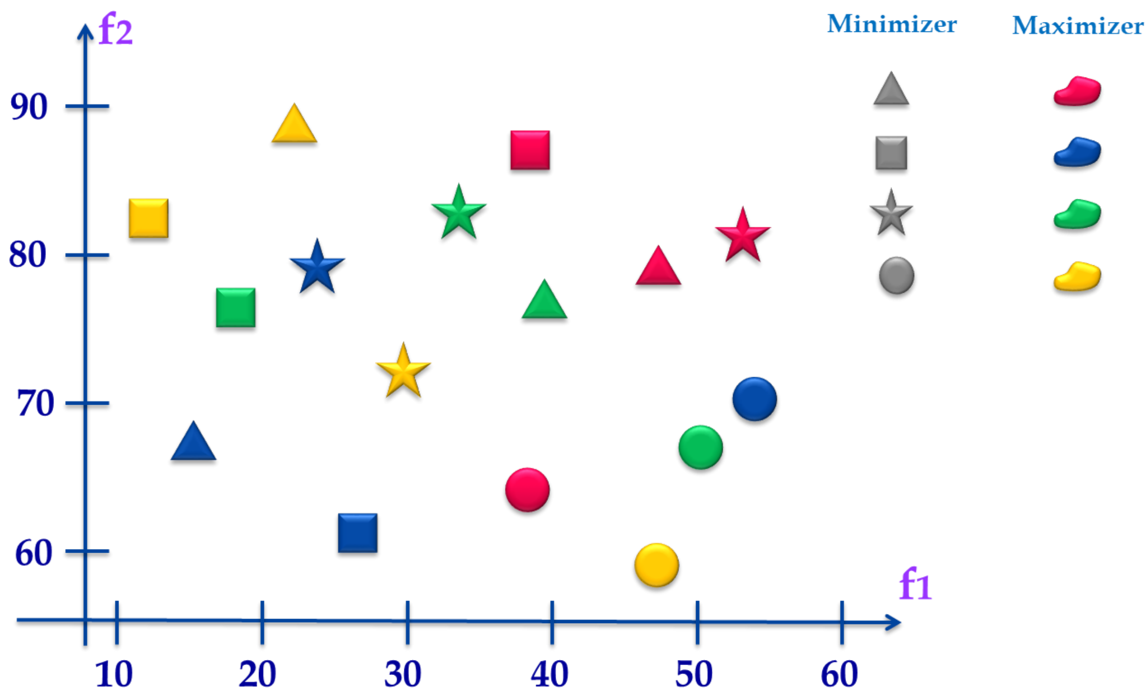
Maximizer

	2	3	-8	11
	9	12	6	16
	-3	10	-4	13
	15	0	8	14

The minimizer chosen strategies 

The maximizer chosen strategies 

Extending the rationalizability approach to MOGs



- **Two main questions:**

  How to evaluate a strategy in MOGs?

  How to employ rationalizability in MOGs?

Outline

1. Motivation & application areas
2. Background
3. Problem description
4. Introduction to rationalizability
5. Methodology and solution approach
6. Cyber-security example
7. Algorithms and Results
8. Conclusions & future work

Our unique two-stage approach to solving MOGs

- **First stage:**
  - Find all rationalizable strategies and their performances
- **Second Stage:**
  - Strategy selection by multi-criteria decision analysis techniques

How to Evaluate a Strategy ?

- For Each strategy:
  - Interact with each of the opponent strategies
  - Obtain the performance for each interaction
- Note:
  - The strategy's performances is a **set** of payoffs
    - In **SOGs** it is a set of **scalars**
    - In **MOGs** it is a set of **vectors**
- **What is the equivalent of "strategy's performances" in Pareto-optimality?**

Introduction to our Approach

Recall:

1. **How to evaluate a strategy in MOGs?**
2. **How to employ rationalizability in MOGs?**

Also recall: The set of rationalizable strategies is:

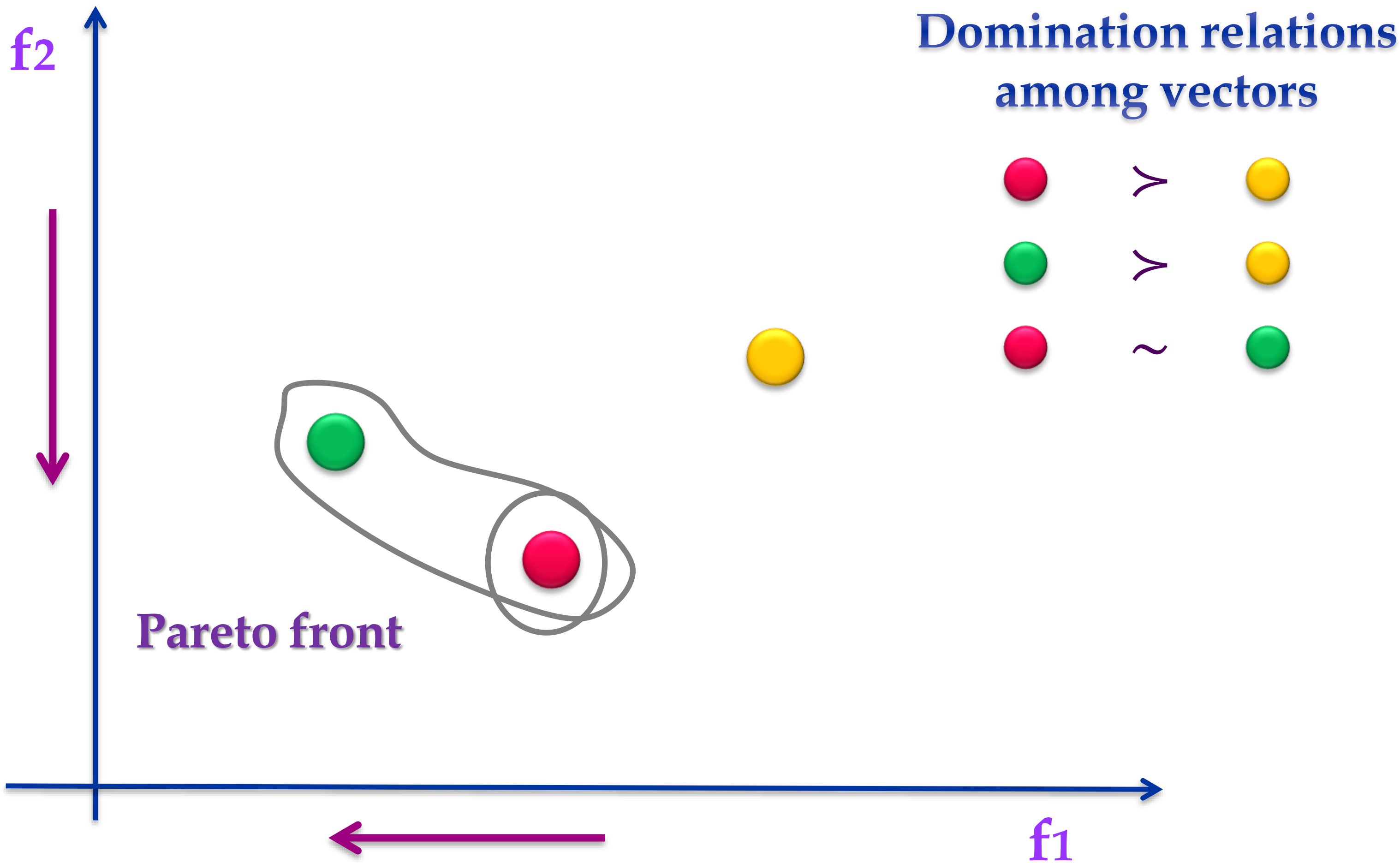
- **The remaining set after iterative elimination of strictly dominated strategies.**

Proposed mutual-rationalizability approach:

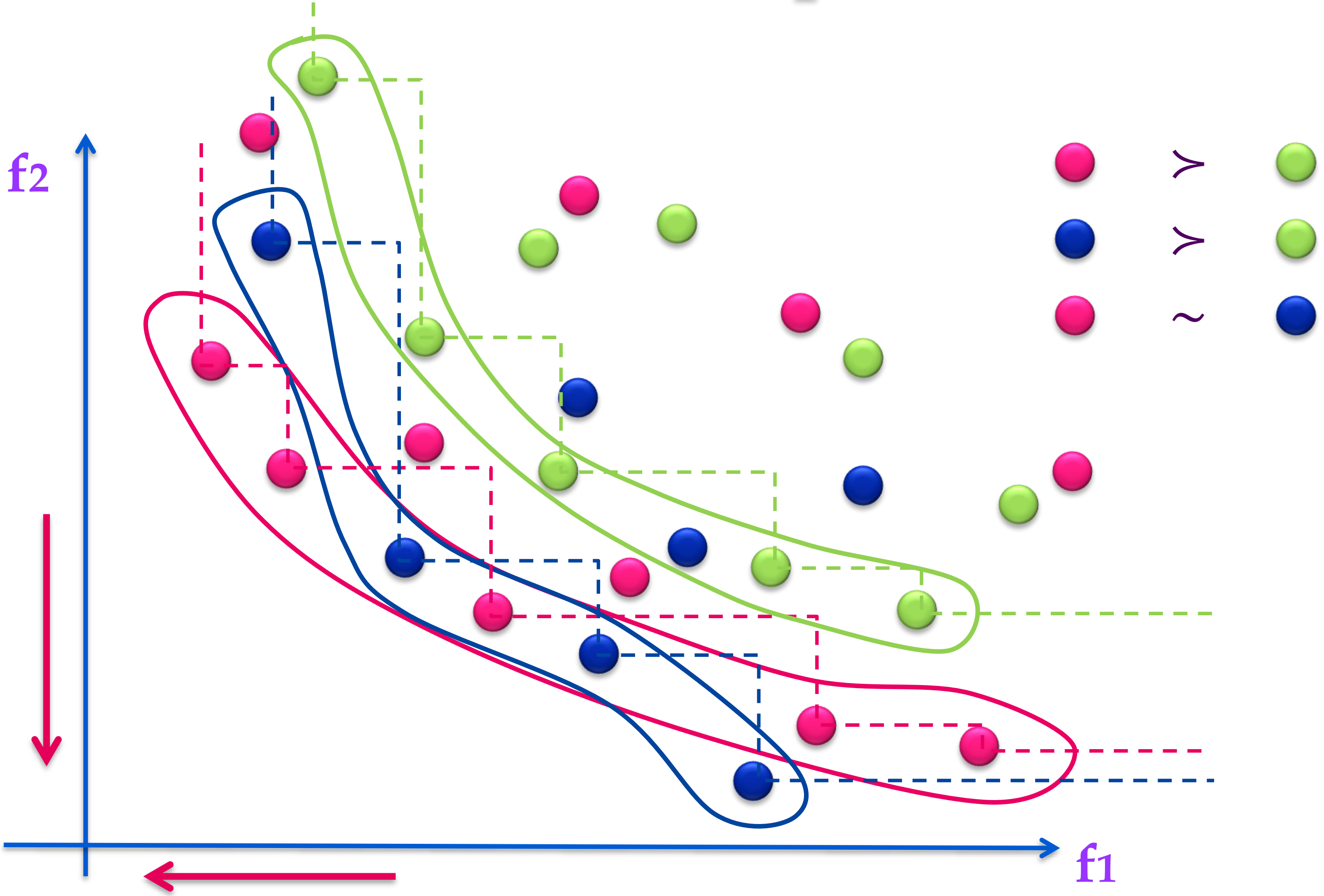
1. Worst-case-base evaluation (Anti-optimal front)
2. Iteratively remove any strategy that will never be chosen under any objective preferences

We also proposed one-sided rationalizability

Recall: The elimination of solutions in multi-objective optimization



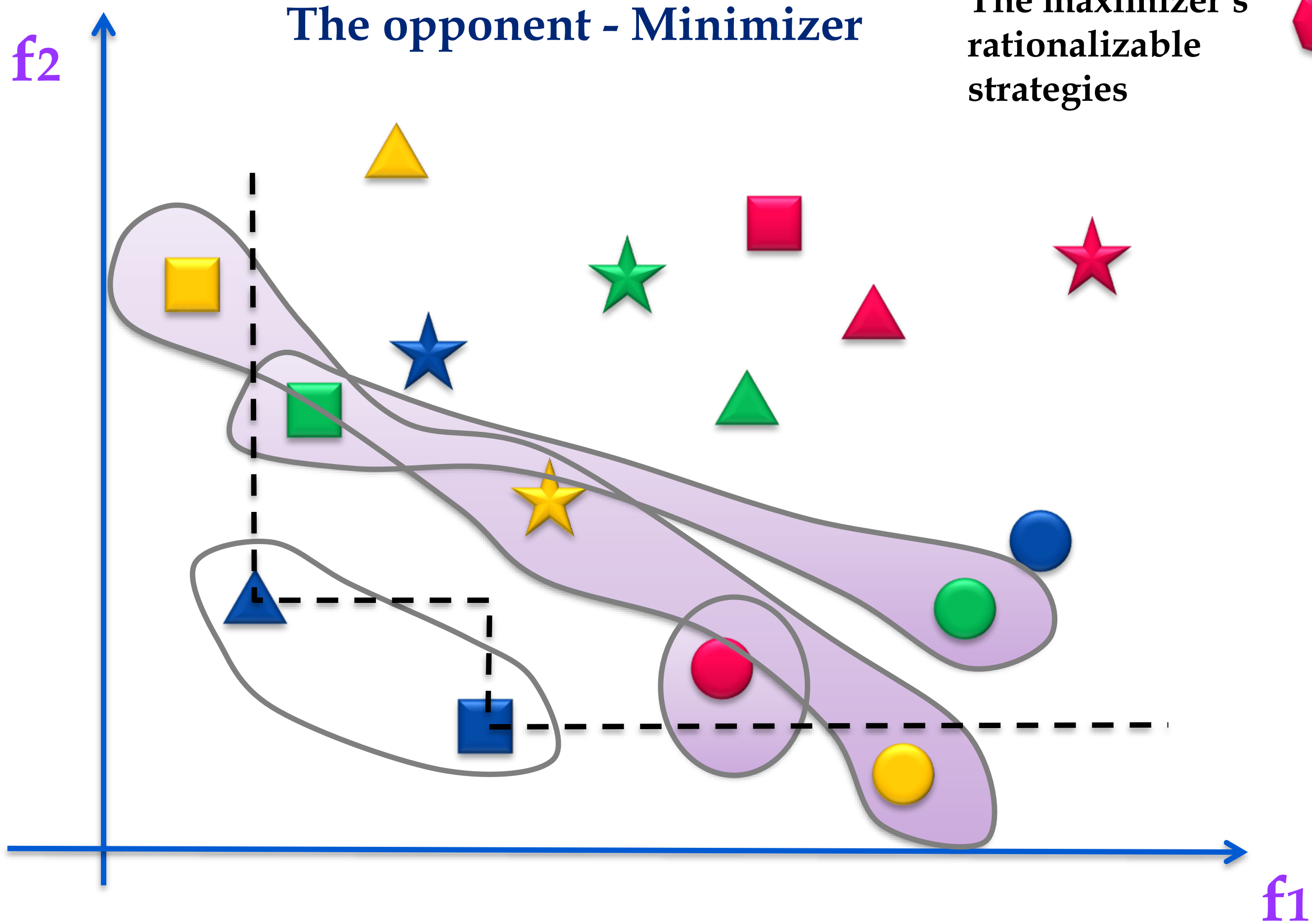
Domination relations among sets
in a minimization problem



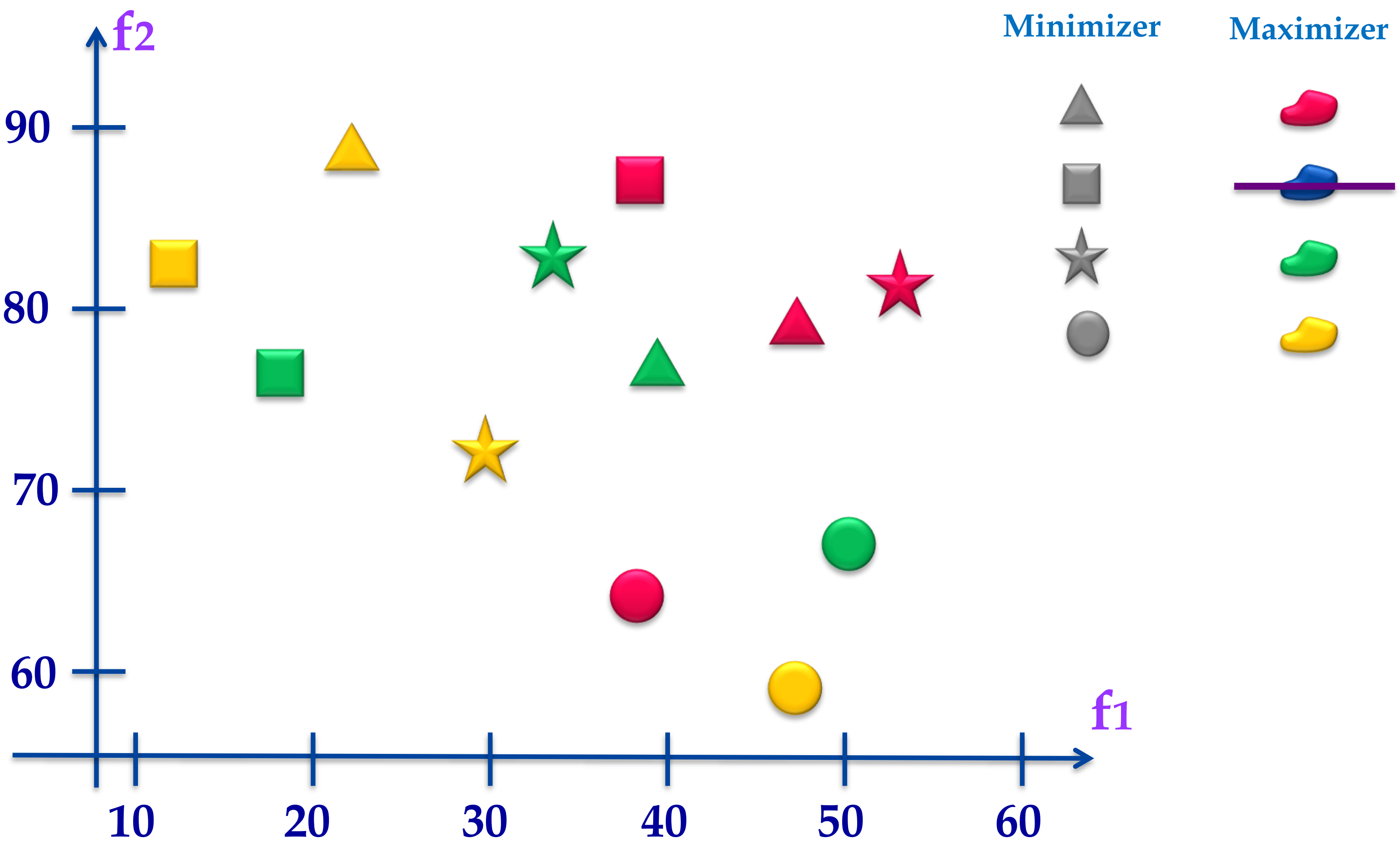
Solving the MOG without a utility function
The maximizer viewpoint

The DM – Maximizer
The opponent - Minimizer

The maximizer's rationalizable strategies

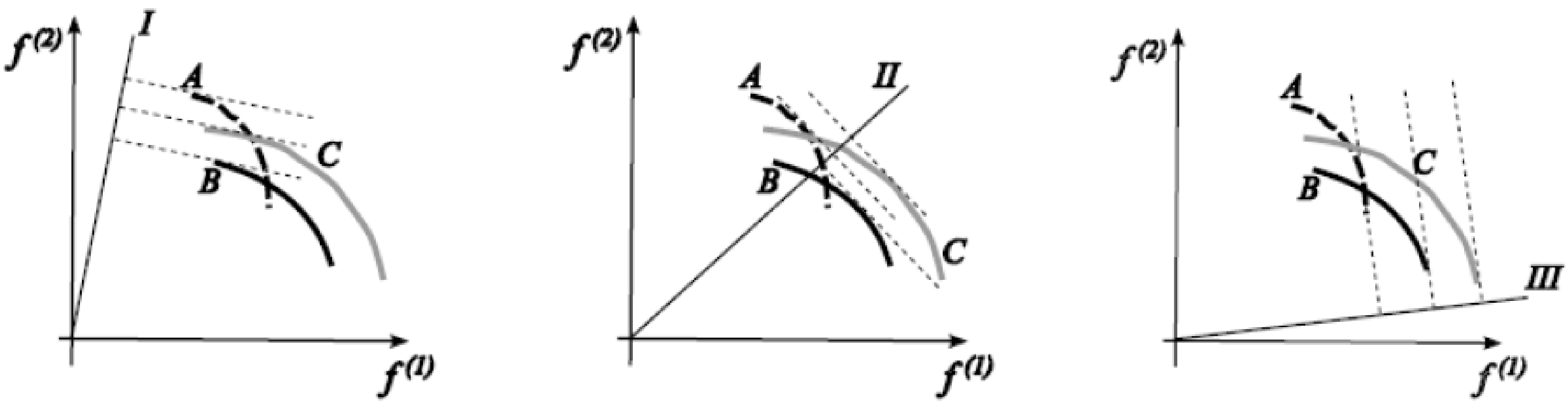


The MOG after the first iteration



Demonstration of an Irrational Strategy

A strategy is irrational if it will never be chosen under any objective preferences



(a) Case 1 (b) Case 2 (c) Case 3

Figure 1: Illustration of irrational strategy

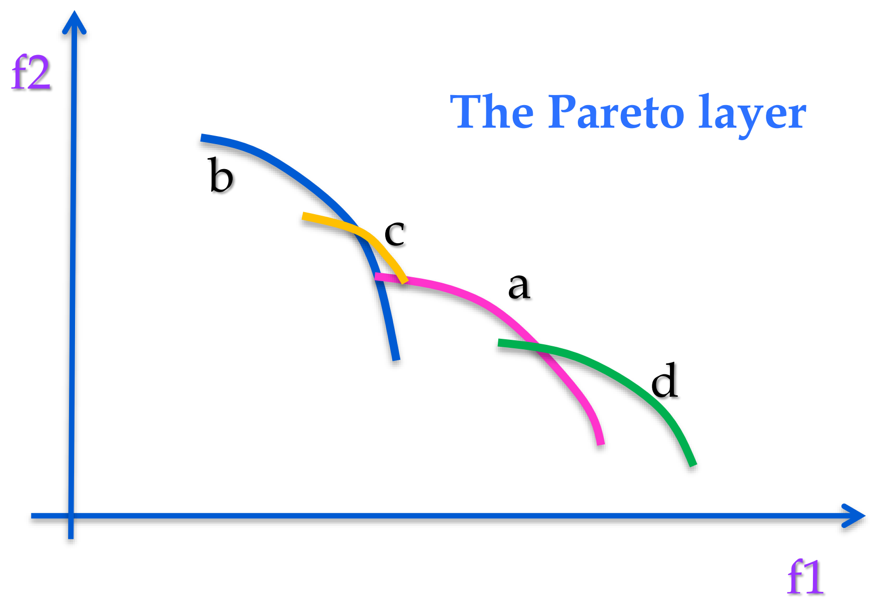
Second-Stage:
Considerations when selecting a strategy

The question is:

How to make a justifiable decision on a strategy?

Which strategy will you prefer?

Which criteria did you use to make the decision?



Set-based MCDA

Motivation:

- Reducing the set of rationalizable strategies
- Selecting a strategy

Suggested methods:

- Sensitivity-Distance (SD)
- Weighted-sum and Aspired-Constraint (WAC)

E. Eisenstadt and A. Moshaiov, "Decision-making in non-cooperative games with conflicting self-objectives," J. Multi-Criteria Decision Analysis, pp. 1–12, 2018.

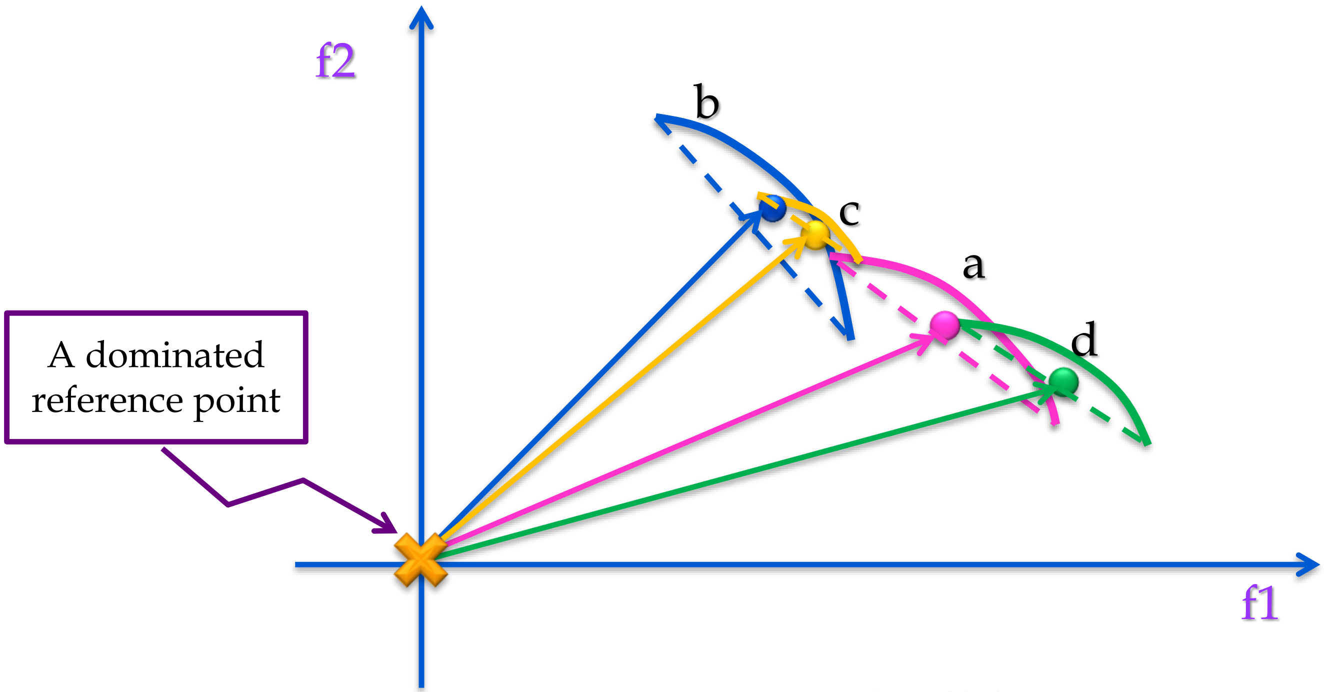
The SD method

- **"Distance"-**

  Distance of the front's center of gravity from a reference dominated point. **The smaller the better**

- **"Sensitivity"-**

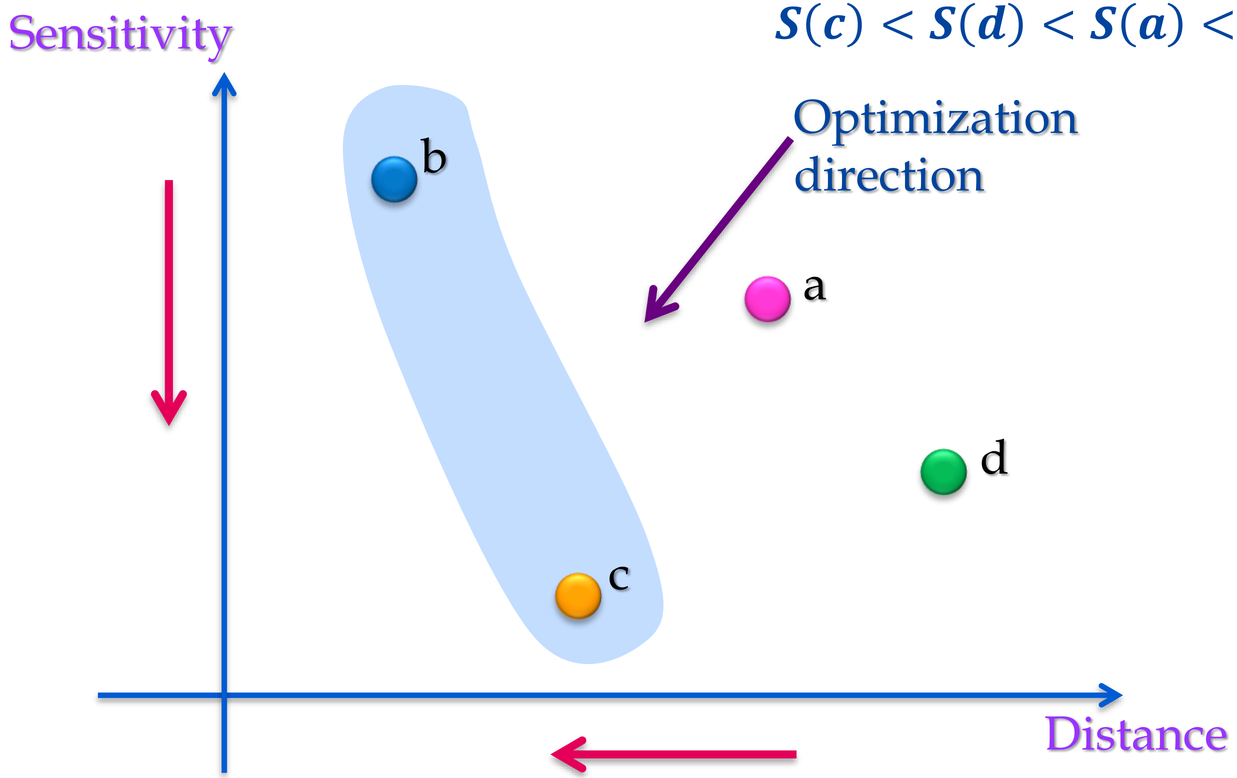
  The front's chord length. **The smaller the better**



Decision Support Auxiliary Space (for the minimizer) SD

$$D(b) < D(c) < D(a) < D(d)$$

$$S(c) < S(d) < S(a) < S(b)$$



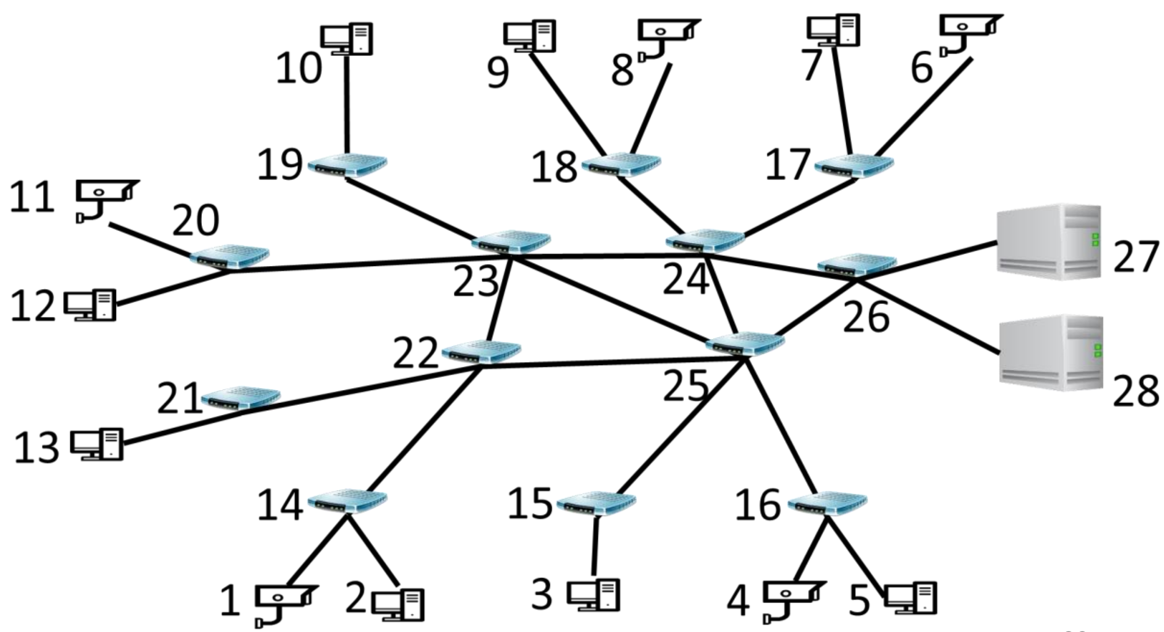
Outline

1. Motivation & application areas
2. Background
3. Problem description
4. Introduction to rationalizability
5. Methodology and solution approach
6. Cyber-security MOG example
7. Algorithms and Results
8. Conclusions & future work

Game Highlights

- **The players:**
  - Hacker (Attacker)
  - IT system's manager (De
- **Objectives:**
  - network functionality
  - involved costs

Value	Link (#,#)
1	(1,14), (2,14), (3,15), (4,16), (5,16), (6,17), (7,17), (8,18), (9,18), (10,19), (11,20), (12,20), (13,21)
2	(14,22), (15,25), (16,25), (17,24), (18,24), (19,23), (20,23), (21,22), (22,23), (22,25), (23,24), (23,25), (24,25), (24,26), (25,26)
5	(26,27)
200	(26,28)



Defender Strategies

- Choses links to change their BW from the initial value
- Decide on the actual BW change for each of the chosen links
- But the defender has a limited amount of BW to add
- Discrete BW values are used to avoid a mixed-integer problem
- There is a cost associated with the BW changes
- Total # of defender strategies = 32,815

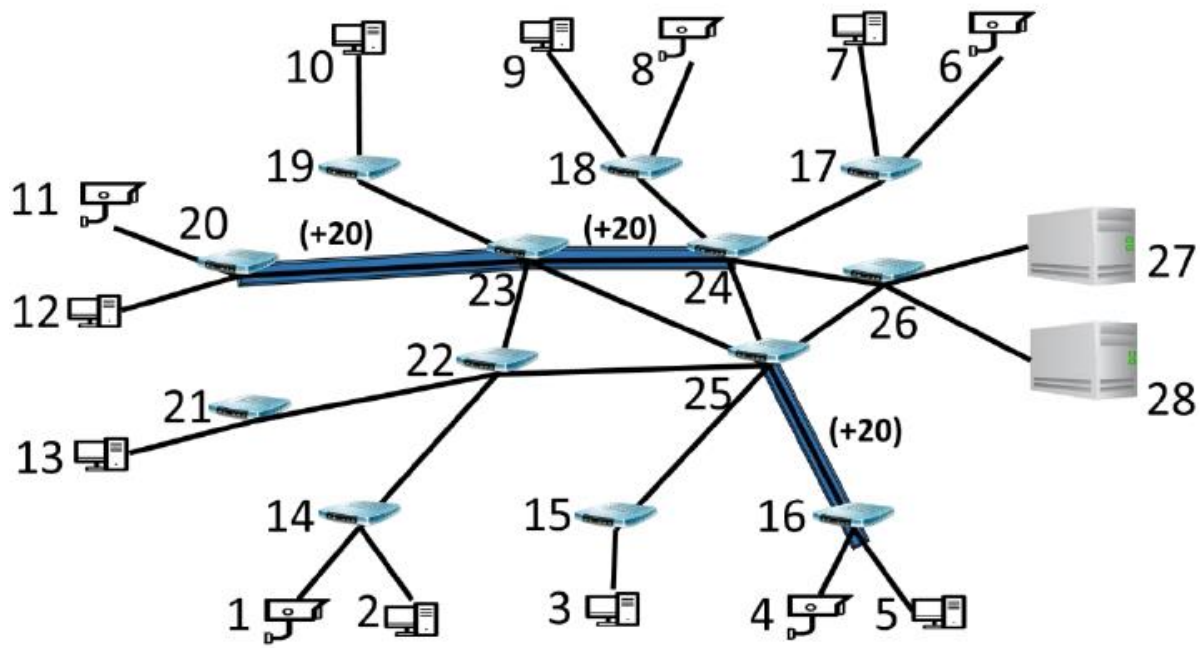
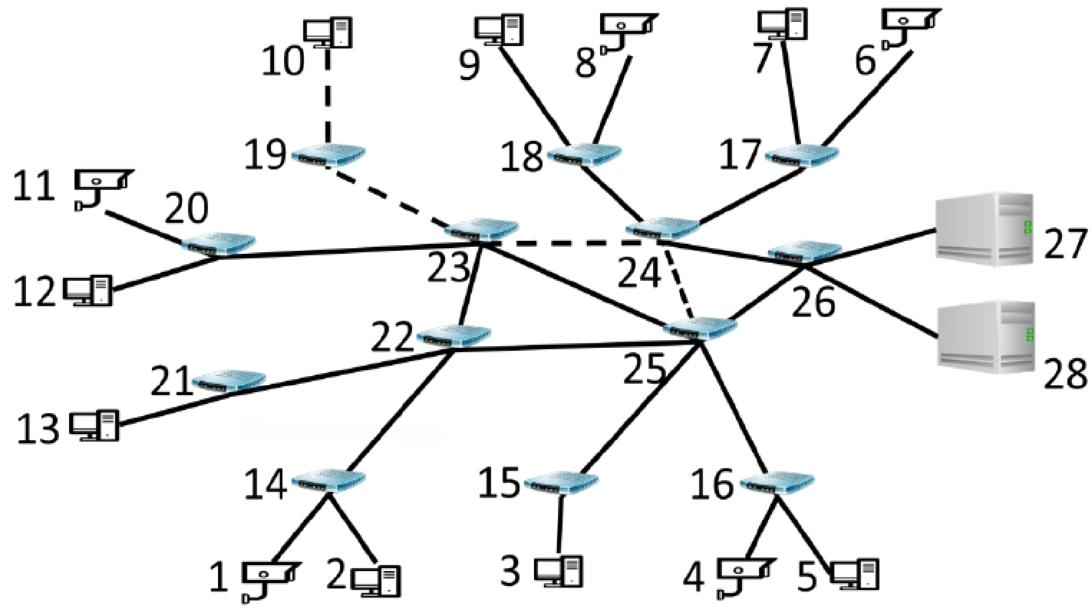


Figure 10: Case study B: Defender strategies

Attacker Strategies

- Chooses a path from an accessible node
- There is a cost for capturing a node (Risk of getting caught)
- Chooses BW of his interference signal
  - Discrete BWs are used (as for the defender)
- Actual BW of attacker's signal is bounded by path bottleneck
- Actual signal may differ from the attempted one!
- There is a cost proportional to the BW of the attempted signal
- # of attacker's strategies = 28,026



Cost	Accessible Leaf Node #	Non-accessible Leaf Node #	Other Node #
1	8	-	-
2	-	-	14-26
5	-	27,28	-
1500	1-7,9-13	-	-

Interaction Example

Initial BW=20 in all links
Defender added 20 to each of the marked three links
Attacker sends BW=20 thru four links

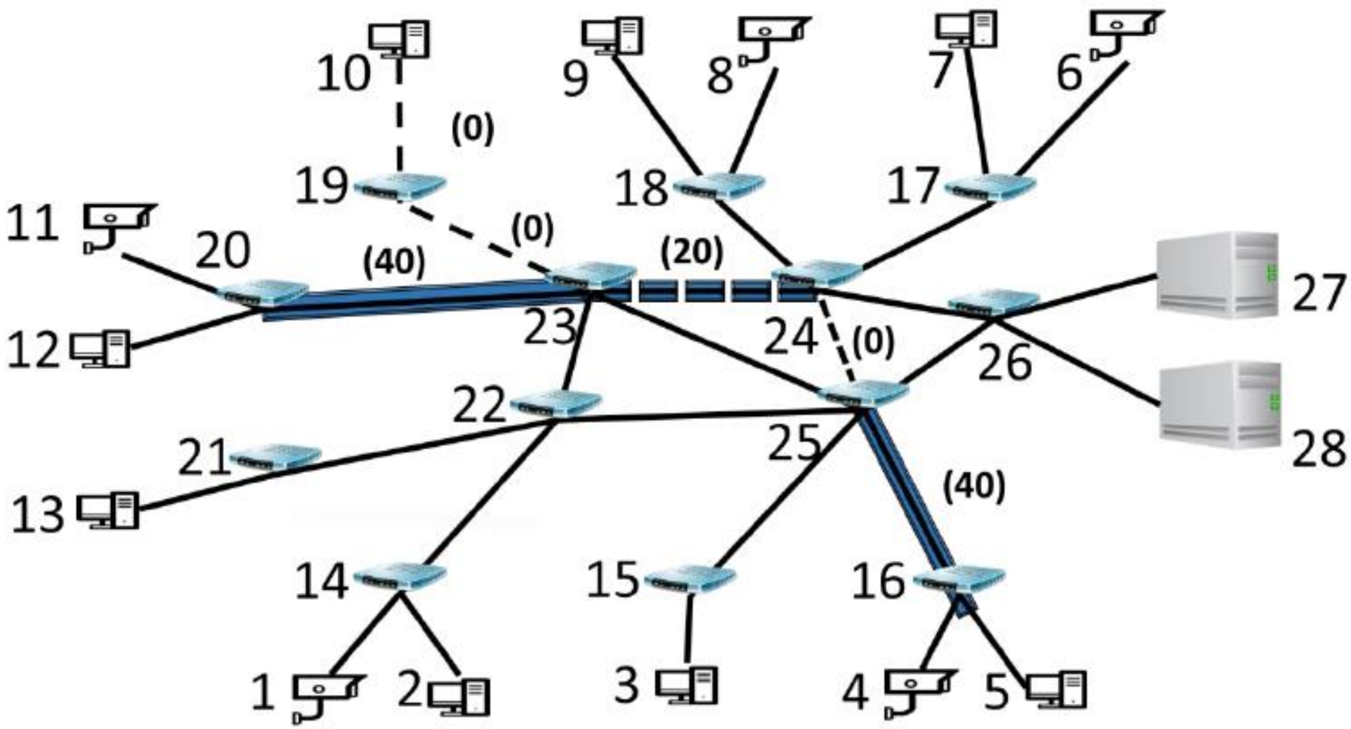


Figure 12: Case study B: Example of strategies interaction

Payoffs and Objectives

Network functionality

- This property describes the efficiency of the network by summing all the available *bw* of the links weighted by their importance.

$$f^{(1)} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} bw(i,j) v_{link}(i,j)$$

- The defender aims to maximize $f^{(1)}$
- The attacker aims to minimize it.

More on Payoffs and Objectives

Cost differential

- This property describes the difference between the attacker cost and the defender cost.

$C_A = \sum_{i=1}^{n} C_{node}(i) + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} C_{trans}(i)$, **when** $C_{trans}\big(chain(i)\big) = \beta \times bw_a\big(chain(i)\big)$

$C_D = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} C_{chang}(i,j)$, **when** $C_{chang}(i,j) = \alpha \times |bw_d(i,j)|$

$$\boldsymbol{f^{(2)} = C_A - C_D}$$

- The defender aims to maximize $f^{(2)}$
- The attacker aims to minimize it

How many interactions ?

Total # of interactions 32,815X28,026= ~9.2 10^8

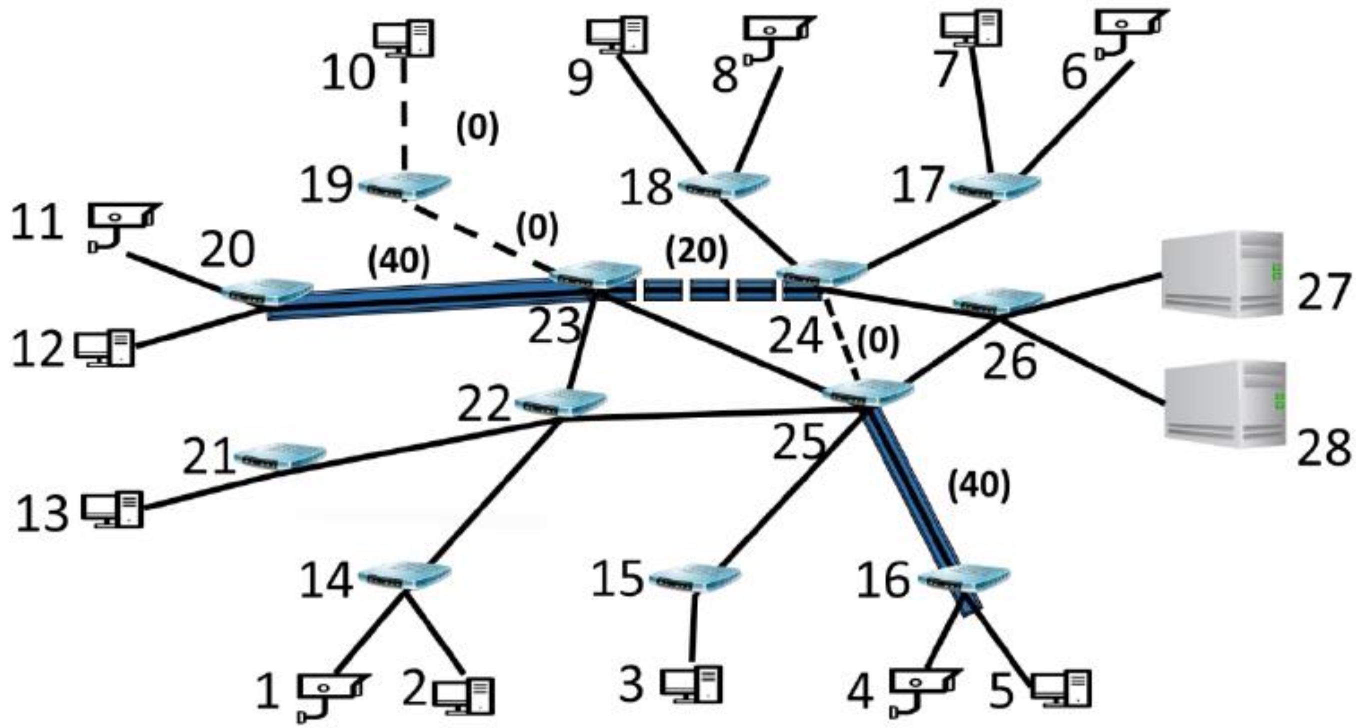


Figure 12: Case study B: Example of strategies interaction

Outline

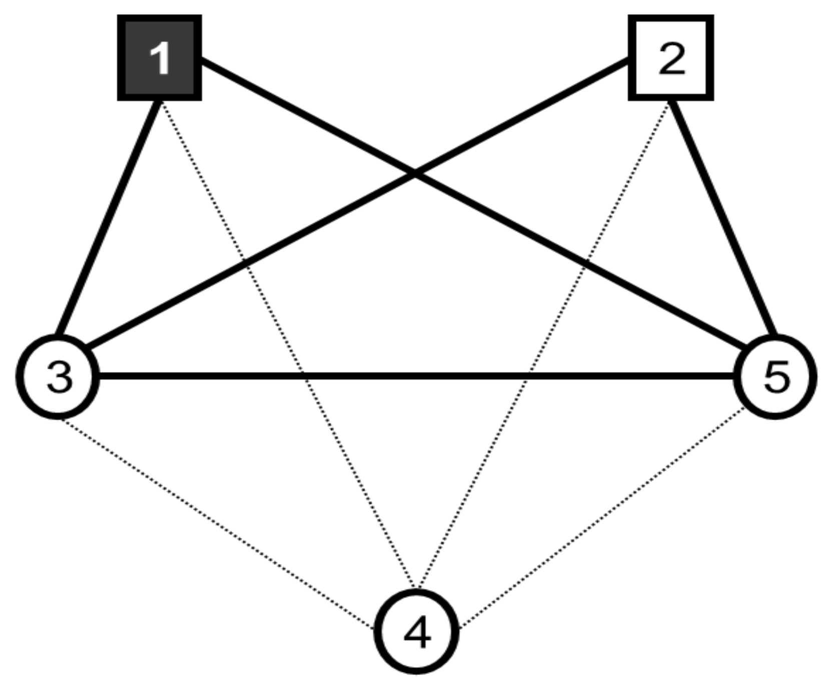
1. Motivation & application areas
2. Background
3. Problem description
4. Introduction to rationalizability
5. Methodology and solution approach
6. Cyber-security MOG example
7. Algorithms and Results
8. Conclusions & future work

The Suggested HoF-based Algorithm - Overview

- Key Features:
  - **Co-evolutionary Algorithm**
  - **Selection by:**
    - Non-domination among sets!
      - Front-ranking
      - Front-crowding
  - **Reproduction operators**
    - Adjusted to combinatorial MOGs
  - **Hall of Fame (HoF)**
    - A kind of a long memory of evolution
    - Each strategy in the HoF has a score
  - **Alternatively: Elite archive** (one generation memory)

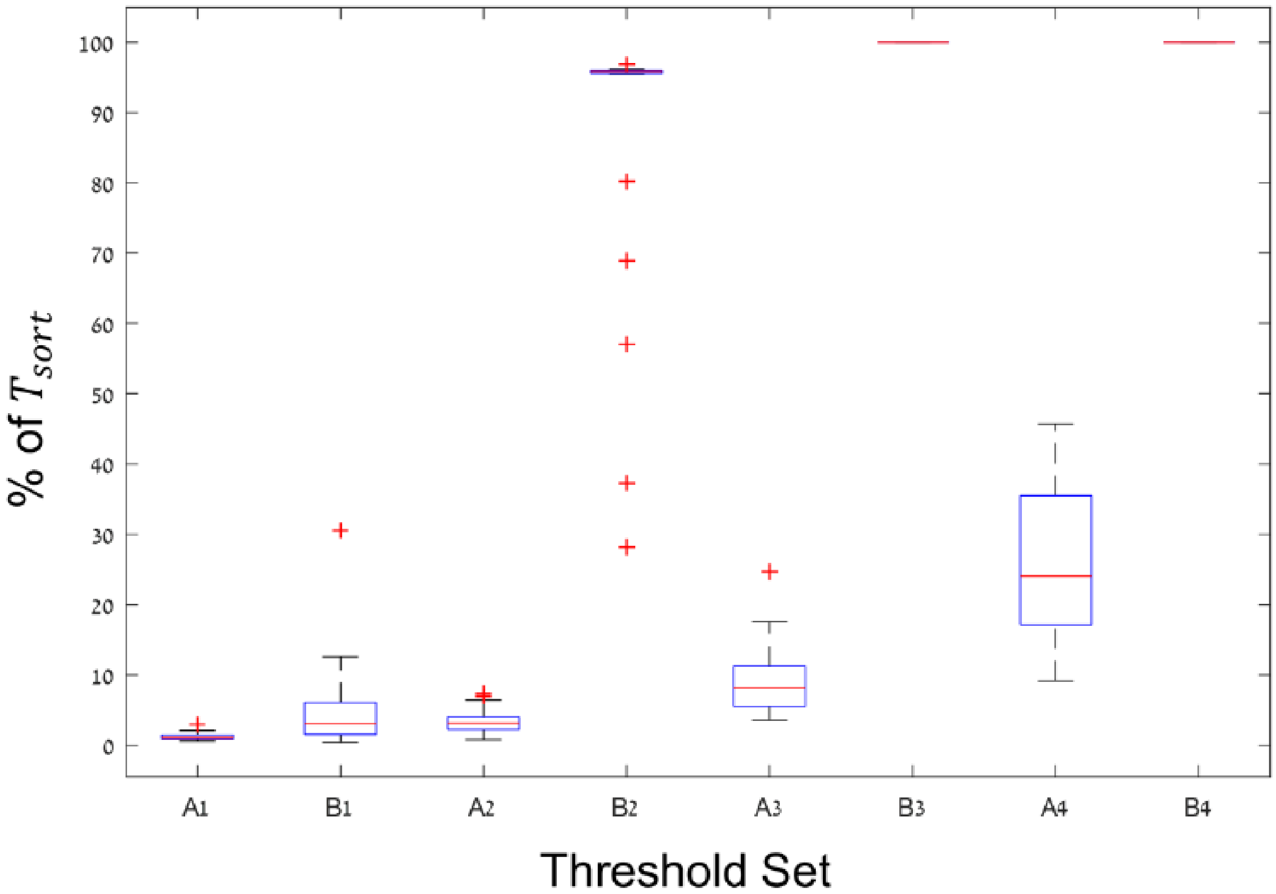
Validation and Comparison Studies - Case A

- 208 X 192 interactions
- Standard laptop
- Reference SRS by full sorting:
  - **6 strategies for the defender**
  - **11 strategies for the attacker**
- Comparing the obtained SRS with the reference one
  - HoF vs. Elite-based algorithm



Run-time Results – Case A

Threshold set #	1	2	3	4
Attacker's threshold number	3	6	9	11
Defender's threshold number	5	10	15	19



A: HoF
B: Elitism

The Relative Evaluation Method for Case B

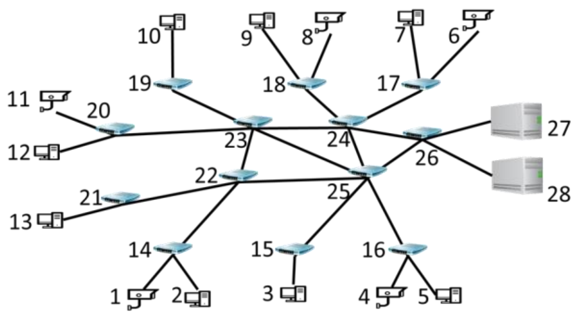
- Hip – the set obtained for player p by the i-th run using Alg-H
- Ejp - the set obtained for player p by the j-th run using Alg-E
- 30 runs per algorithm
- Create 900 union sets per each player :

$$\mathrm{UA_{ij}}^{p} = \mathrm{H_{ip}} \cup \mathrm{E_{jp}}$$

- Sort each union to find the set of 1st rank strategies:

$$\mathrm{UA}_{ij}^{*P} \subseteq \mathrm{UA}_{ij}^{P}$$

- Two measures are calculated (ideally = one):



$$\mathrm{h}_{ij}^{p} = \frac{\left|\mathrm{H}_{ip} \cap UA_{ij}^{*P}\right|}{\left|\mathrm{H}_{ip}\right|}, \mathrm{e}_{ij}^{p} = \frac{\left|\mathrm{E}_{jp} \cap UA_{ij}^{*P}\right|}{\left|\mathrm{E}_{jp}\right|}$$

Results for the attacker

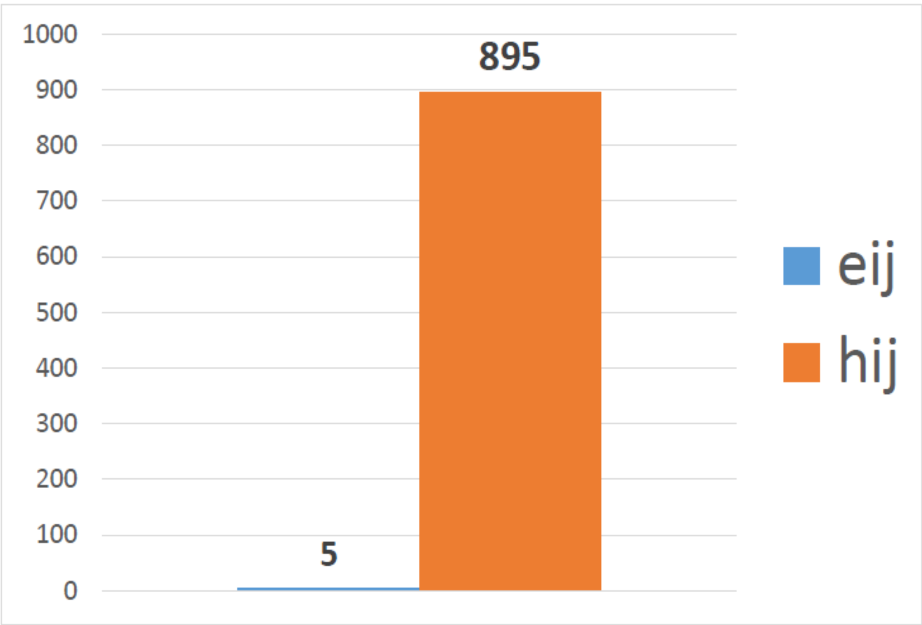
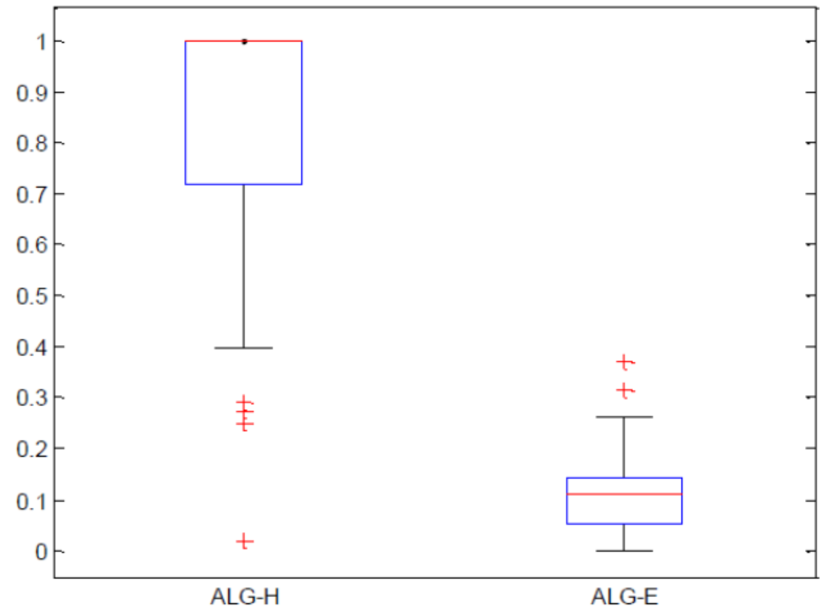
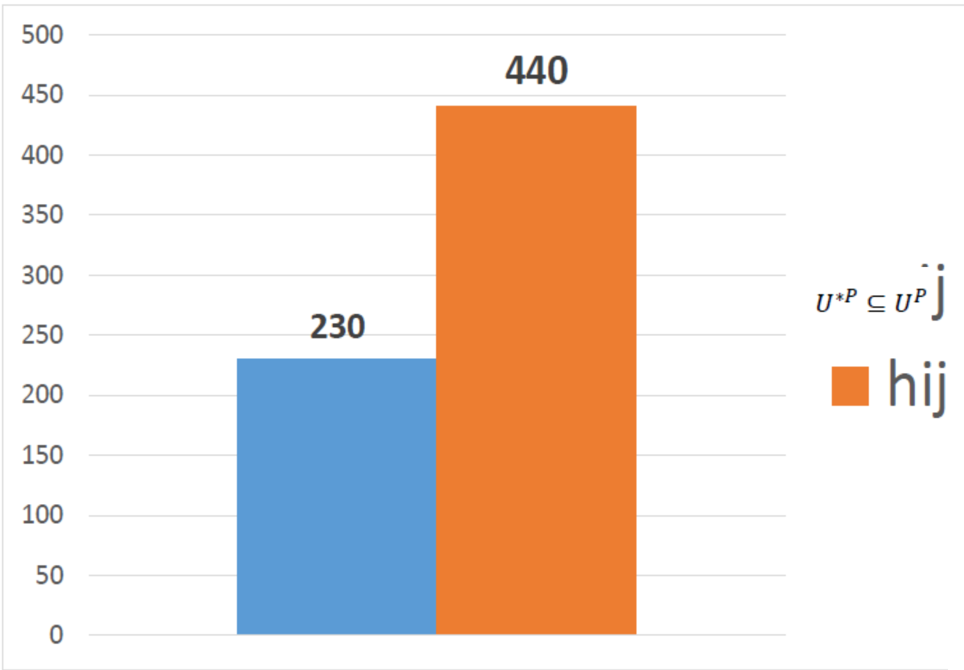


Figure 21: Case Study B: Comparison between h_{ij} and e_{ij} of the attacker



Results for the defender



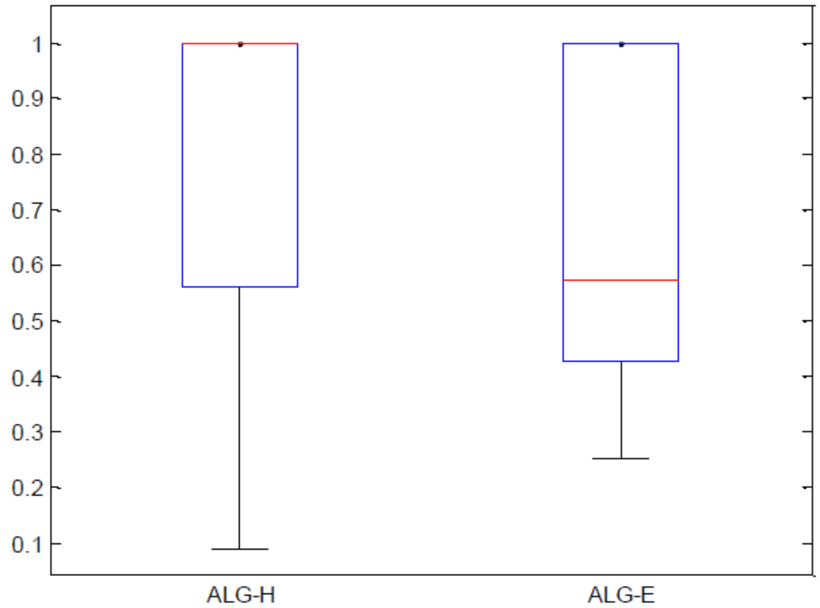


Figure 18: Case Study B: Comparison between h_{ij} and e_{ij} of the defender

Consistency Study

- Let U^P be a multiset from the union of all HoFs of the 30 runs
- Let $U^{*P} \subseteq U^P$ be the set of 1st rank strategies of the union
- Is there a correlation between 1st rank strategies and strategies with high multiplicities in the union of the HoFs.

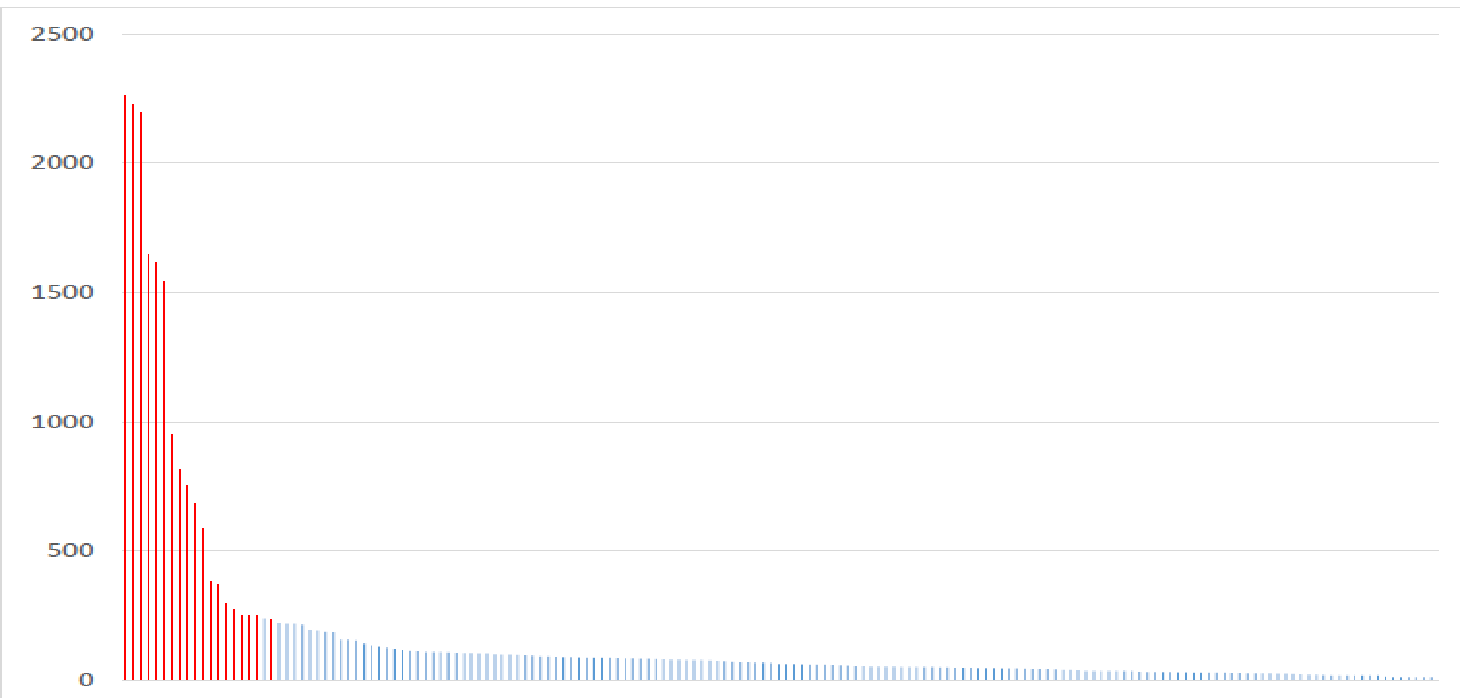


Figure 24: Case Study B: Results for the attacker in consistency study

Summary & Future work

- A non-traditional solution approach to MOGs has been suggested and formulated
- A Cyber-security MOG has been presented
- Methods to compare algorithms have been presented
- HoF-based algorithm was found to be superior
- Other MOGs that we have suggested:
  - Aeronautical MOGs
  - Competing TSP-MOGs
- Under various stages of development:
  - Proofs of related theorems
  - Alternative algorithms
  - Measures to evaluate and compare algorithms/runs
  - Alternative MCDM approaches for selecting a strategy
  - New MOGs (e.g., Colonel Blotto as a MOG, revised TSP)
  - Other types of MOGs (e.g., non-zero-sum MOGs, mixed strategy)
    - …


Questions?


thank you