Adversarial AI for Solving Complex Security Problems in Engineered Systems

Daniel Tauritz, PhD

AI4Sec:FND
Auburn University

January 30, 2024

Part I: Engineered Systems & Security

What is an Engineered System?

NSF's Engineering Research Center website defines engineered systems as:

> "a combination of components that work in synergy to collectively perform a useful function. The engineered system could, for example, wholly or in part constitute a new technology for a new product line a new manufacturing process, a technology to improve the delivery of a service, or an infrastructure system."

What is an Engineered System?

NSF's Engineering Research Center website defines engineered systems as:

> "a combination of components that work in synergy to collectively perform a useful function. The engineered system could, for example, wholly or in part constitute a new technology for a new product line a new manufacturing process, a technology to improve the delivery of a service, or an infrastructure system."

Examples:

- Modern Planes, Trains, and Automobiles
- Industry 4.0: Chemical Plant, Biotechnology, Agriculture
- Modern Utilities: Electric, Water, Gas, Oil
- Satellite Constellations (e.g., Starlink)
- Internet, Enterprise Computer Networks, Cloud Computing

Critical Infrastructure Sectors

DHS' Cybersecurity and Infrastructure Security Agency (CISA) lists 16 critical infrastructure sectors:

- Chemical
- Commercial Facilities
- Communications
- Critical Manufacturing
- Dams
- Defense Industrial Base
- Emergency Services
- Energy
- Financial Services
- Food and Agriculture
- Government Facilities
- Healthcare and Public Health
- Information Technology Sector
- Nuclear Reactors, Materials, and Waste
- Transportation Systems
- Water and Wastewater Systems

The Problem

- Modern engineered systems tend to be cyber/cyber-physical in nature

The Problem

- Modern engineered systems tend to be cyber/cyber-physical in nature
- Cyber & Cyber-physical engineered systems are extremely vulnerable to attack

The Problem

- Modern engineered systems tend to be cyber/cyber-physical in nature
- Cyber & Cyber-physical engineered systems are extremely vulnerable to attack
- Cyber & Cyber-physical engineered system attack surfaces tend to be astronomically large and infeasible to fully secure

The Problem

- Modern engineered systems tend to be cyber/cyber-physical in nature
- Cyber & Cyber-physical engineered systems are extremely vulnerable to attack
- Cyber & Cyber-physical engineered system attack surfaces tend to be astronomically large and infeasible to fully secure
- Only AI is capable of examining the combinatorially large number of unique attacks and defenses on modern engineered systems

Engineered System Security as a Game

- Two or more adversaries: one defender and one or more attackers

Engineered System Security as a Game

- Two or more adversaries: one defender and one or more attackers
- Attackers range from so-called "script-kiddies" to organized crime, terrorist organizations, and adversarial nation states

Engineered System Security as a Game

- Two or more adversaries: one defender and one or more attackers
- Attackers range from so-called "script-kiddies" to organized crime, terrorist organizations, and adversarial nation states
- Attacker goals are diverse

Engineered System Security as a Game

- Two or more adversaries: one defender and one or more attackers
- Attackers range from so-called "script-kiddies" to organized crime, terrorist organizations, and adversarial nation states
- Attacker goals are diverse
- Defender needs to simulatenously defend against this wide variety of attackers

Engineered System Security as a Game

- Two or more adversaries: one defender and one or more attackers
- Attackers range from so-called "script-kiddies" to organized crime, terrorist organizations, and adversarial nation states
- Attacker goals are diverse
- Defender needs to simulatenously defend against this wide variety of attackers
- Asymetric non-zero sum game

Engineered System Security as a Game

- Two or more adversaries: one defender and one or more attackers
- Attackers range from so-called "script-kiddies" to organized crime, terrorist organizations, and adversarial nation states
- Attacker goals are diverse
- Defender needs to simulatenously defend against this wide variety of attackers
- Asymetric non-zero sum game
- Game theory allows for mathematical analysis of adversarial models

Engineered System Security as a Game

- Two or more adversaries: one defender and one or more attackers
- Attackers range from so-called "script-kiddies" to organized crime, terrorist organizations, and adversarial nation states
- Attacker goals are diverse
- Defender needs to simulatenously defend against this wide variety of attackers
- Asymetric non-zero sum game
- Game theory allows for mathematical analysis of adversarial models
- Classic game theory does not scale to complex, real-world systems

Engineered System Security as a Game

- Two or more adversaries: one defender and one or more attackers
- Attackers range from so-called "script-kiddies" to organized crime, terrorist organizations, and adversarial nation states
- Attacker goals are diverse
- Defender needs to simulatenously defend against this wide variety of attackers
- Asymetric non-zero sum game
- Game theory allows for mathematical analysis of adversarial models
- Classic game theory does not scale to complex, real-world systems
- Computational game theory achieves scalability by approximating Nash equilibria

Part II: AI Armsraces

Computational Problem Solving

- Step 1: build abstract/computational model of the real-world

[1] https://quoteinvestigator.com/2011/05/13/einstein-simple/

Computational Problem Solving

- Step 1: build abstract/computational model of the real-world
- Step 2: solve computationally in abstract model

Computational Problem Solving

- Step 1: build abstract/computational model of the real-world
- Step 2: solve computationally in abstract model
- "Everything Should Be Made as Simple as Possible, But Not Simpler"[1]
- Step 3: map solution back to real-world

[1] https://quoteinvestigator.com/2011/05/13/einstein-simple/

Terminology

- Many computational problems can be formulated as **generate-and-test** search problems

Terminology

- Many computational problems can be formulated as **generate-and-test** search problems
- A **search space** contains the set of all possible solutions

Terminology

- Many computational problems can be formulated as **generate-and-test** search problems
- A **search space** contains the set of all possible solutions
- A **search space generator** is *complete* if it can generate the entire search space

Terminology

- Many computational problems can be formulated as **generate-and-test** search problems
- A **search space** contains the set of all possible solutions
- A **search space generator** is *complete* if it can generate the entire search space
- An **objective function** tests the quality of a solution

Terminology

- Many computational problems can be formulated as **generate-and-test** search problems
- A **search space** contains the set of all possible solutions
- A **search space generator** is *complete* if it can generate the entire search space
- An **objective function** tests the quality of a solution
- A **heuristic** is a problem-dependent rule-of-thumb

Terminology

- Many computational problems can be formulated as **generate-and-test** search problems
- A **search space** contains the set of all possible solutions
- A **search space generator** is *complete* if it can generate the entire search space
- An **objective function** tests the quality of a solution
- A **heuristic** is a problem-dependent rule-of-thumb
- A **meta-heuristic** is a general heuristic to determine the sampling order over a search space with the goal to find a near-optimal solution (or set of solutions)

Terminology

- Many computational problems can be formulated as **generate-and-test** search problems
- A **search space** contains the set of all possible solutions
- A **search space generator** is *complete* if it can generate the entire search space
- An **objective function** tests the quality of a solution
- A **heuristic** is a problem-dependent rule-of-thumb
- A **meta-heuristic** is a general heuristic to determine the sampling order over a search space with the goal to find a near-optimal solution (or set of solutions)
- A **hyper-heuristic** is a meta-heuristic for a space of programs

Algorithmic Toolbox

- A **Black-Box Search Algorithm (BBSA)** is a meta-heuristic which iteratively generates trial solutions employing solely the information gained from previous trial solutions, but no explicit problem knowledge

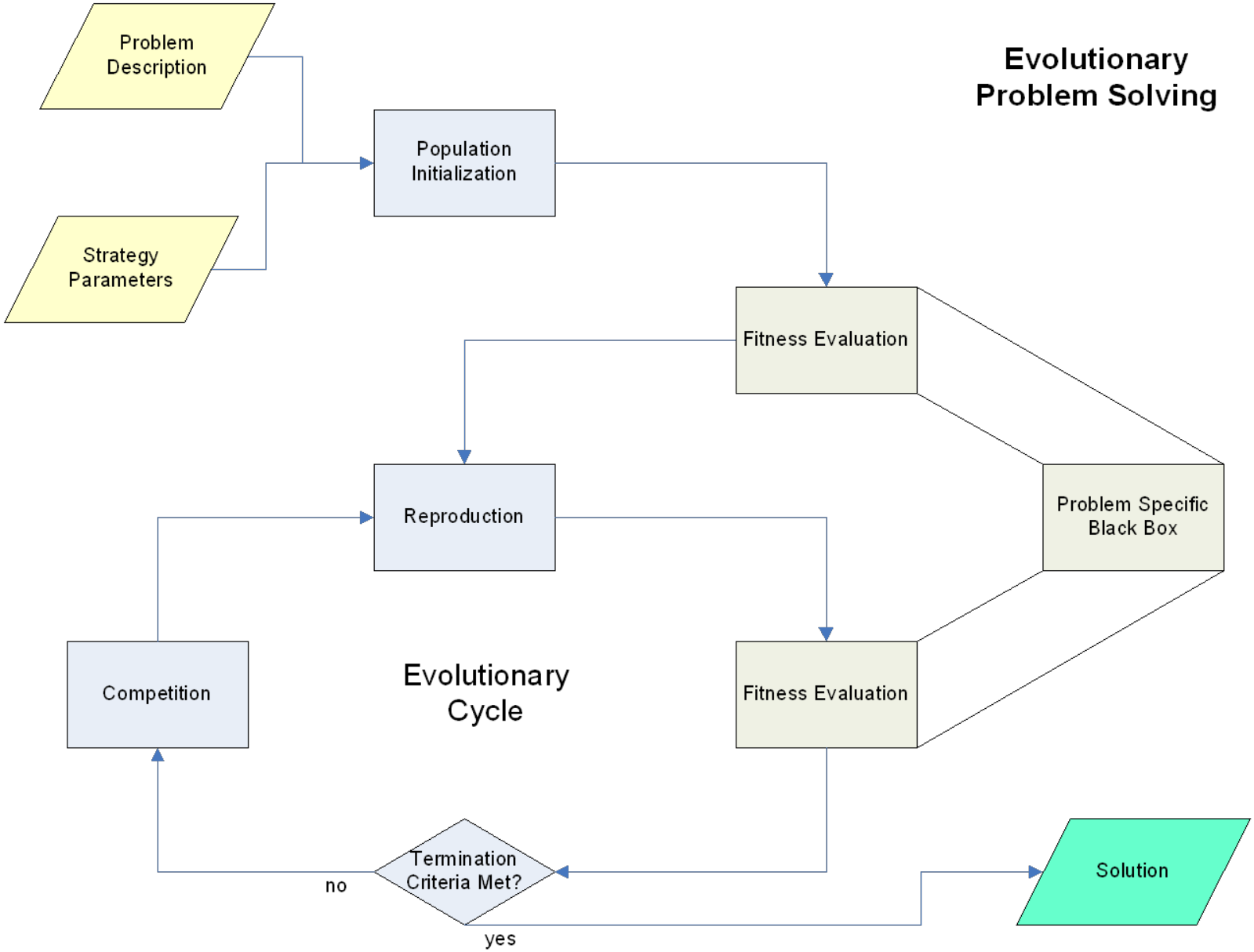
Algorithmic Toolbox

- A **Black-Box Search Algorithm (BBSA)** is a meta-heuristic which iteratively generates trial solutions employing solely the information gained from previous trial solutions, but no explicit problem knowledge
- **Evolutionary Algorithms (EAs)** can be described as a class of *stochastic*, *population-based* BBSAs inspired by *Evolution Theory*, *Genetics*, and *Population Dynamics*

Algorithmic Toolbox

- A **Black-Box Search Algorithm (BBSA)** is a meta-heuristic which iteratively generates trial solutions employing solely the information gained from previous trial solutions, but no explicit problem knowledge
- **Evolutionary Algorithms (EAs)** can be described as a class of *stochastic*, *population-based* BBSAs inspired by *Evolution Theory*, *Genetics*, and *Population Dynamics*

Evolutionary Cycle



Genetic Programming

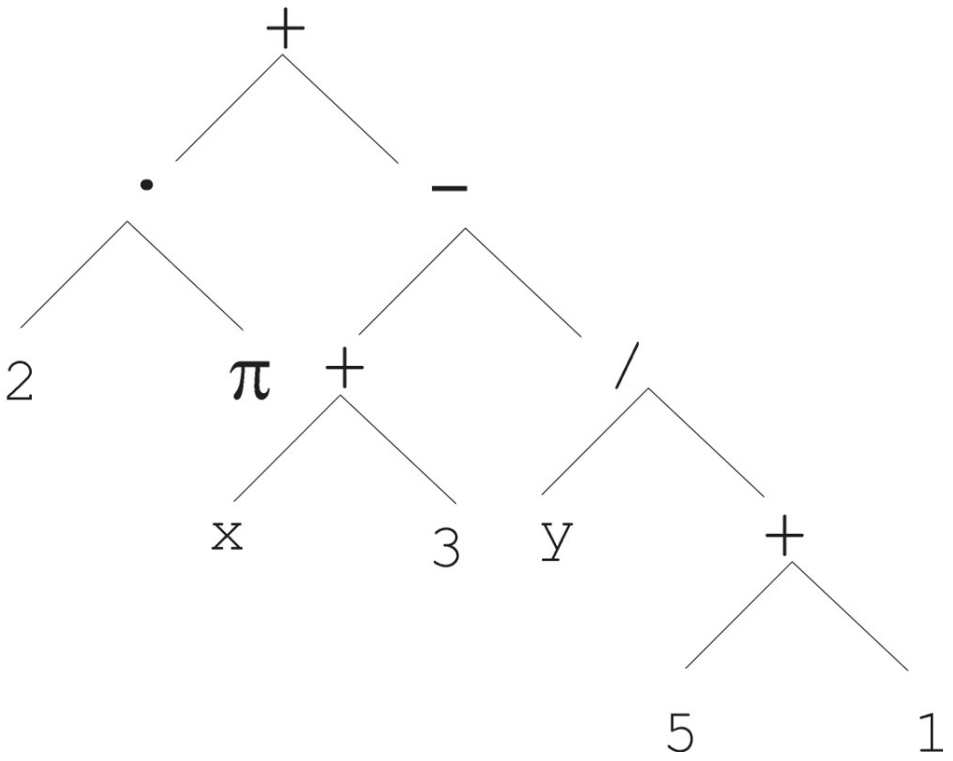
- EA with Hierarchical Representation for Model Identification

Genetic Programming

- EA with Hierarchical Representation for Model Identification
- Koza style Tree GP is the most prevalent

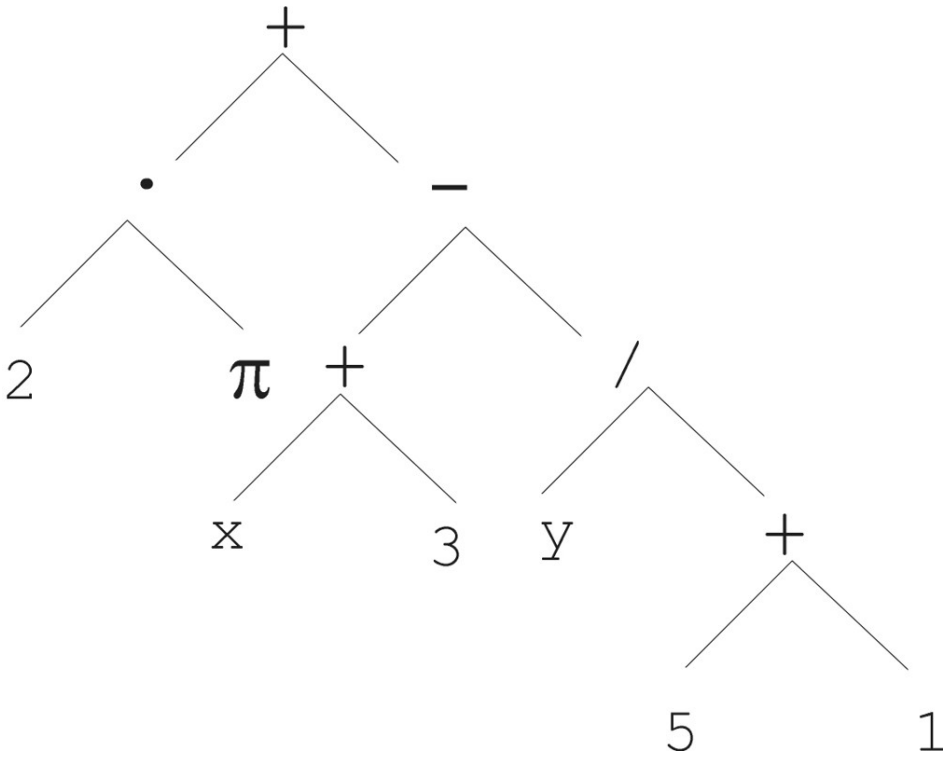
Genetic Programming

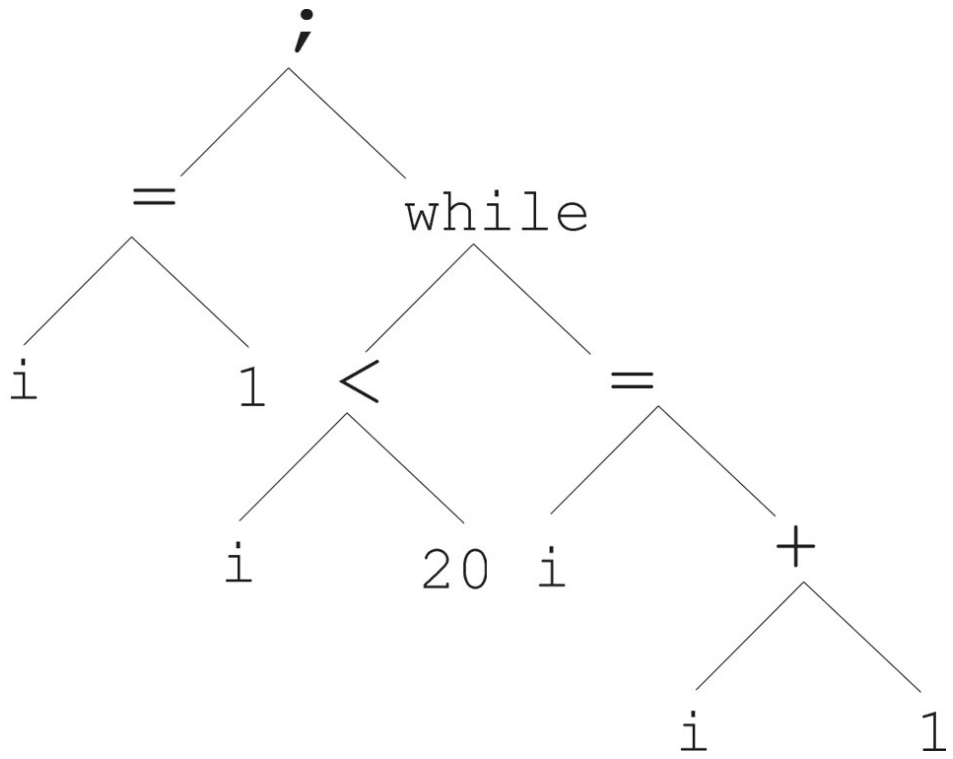
- EA with Hierarchical Representation for Model Identification
- Koza style Tree GP is the most prevalent



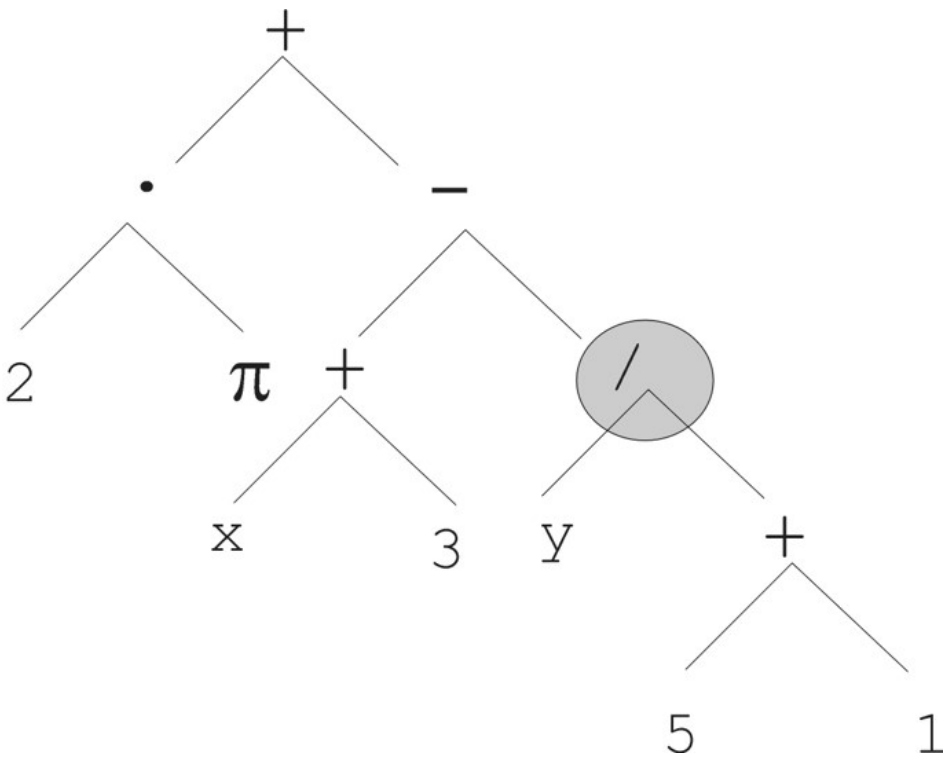
Genetic Programming

- EA with Hierarchical Representation for Model Identification
- Koza style Tree GP is the most prevalent

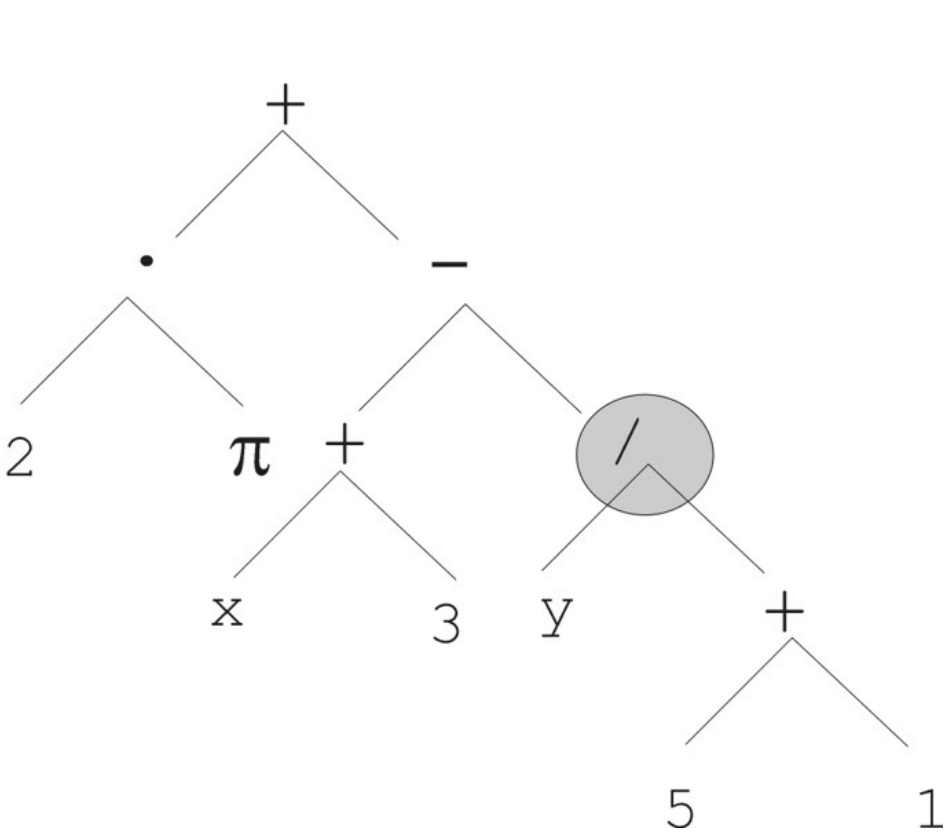


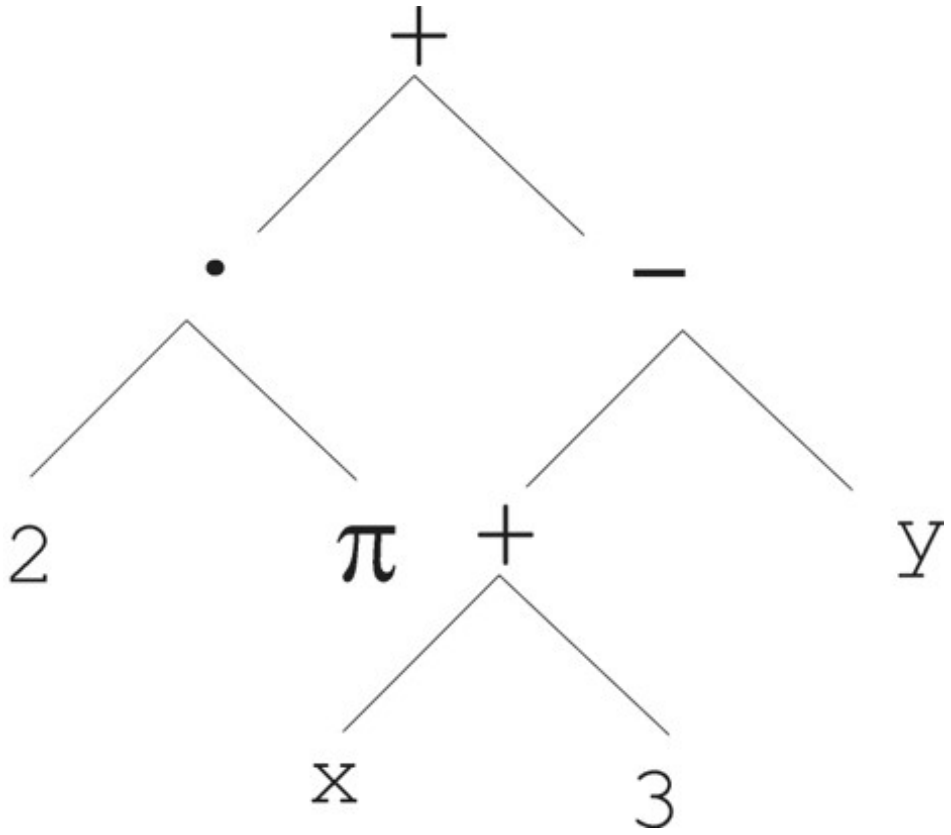


Genetic Programming - Mutation

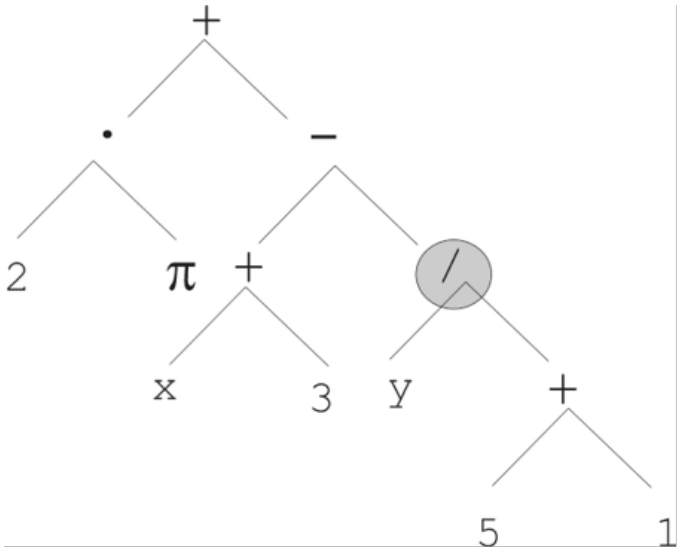


Genetic Programming - Mutation

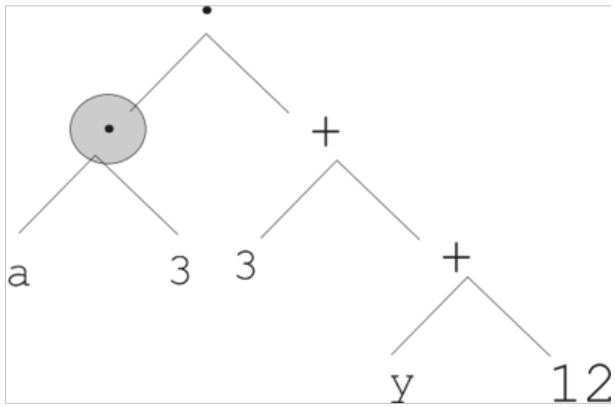




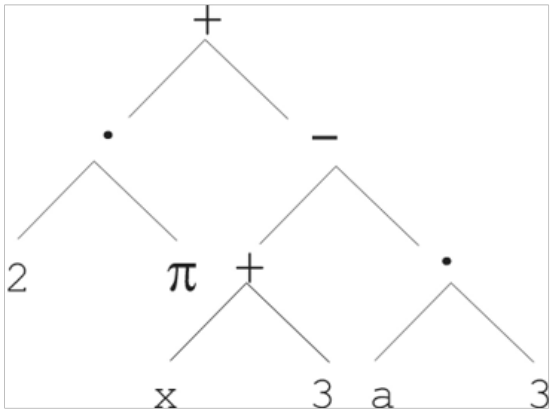
Genetic Programming - Recombination



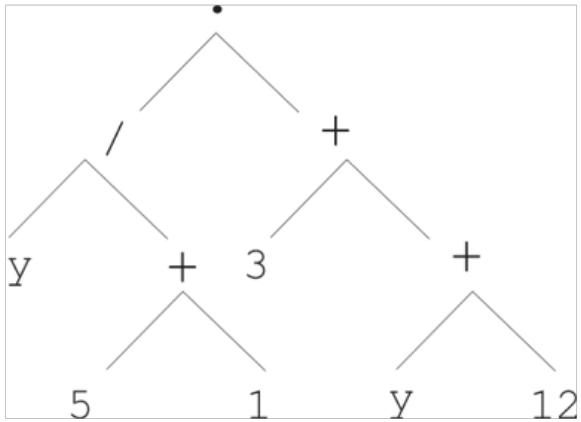
Parent 1



Parent 2



Child 1



Child 2

Real-World Game-Theoretic Problems

- Game Theory: multi-agent problem with conflicting utility functions

Real-World Game-Theoretic Problems

- Game Theory: multi-agent problem with conflicting utility functions
- Real-world examples:
  - economic & military strategy

Real-World Game-Theoretic Problems

- Game Theory: multi-agent problem with conflicting utility functions
- Real-world examples:
  - economic & military strategy
  - arms control

Real-World Game-Theoretic Problems

- Game Theory: multi-agent problem with conflicting utility functions
- Real-world examples:
  - economic & military strategy
  - arms control
  - auctions

Real-World Game-Theoretic Problems

- Game Theory: multi-agent problem with conflicting utility functions
- Real-world examples:
  - economic & military strategy
  - arms control
  - auctions
  - cyber security

Real-World Game-Theoretic Problems

- Game Theory: multi-agent problem with conflicting utility functions
- Real-world examples:
  - economic & military strategy
  - arms control
  - auctions
  - cyber security
- Common problem: real-world games are typically incomputable

Real-World Game-Theoretic Problems

- Game Theory: multi-agent problem with conflicting utility functions
- Real-world examples:
  - economic & military strategy
  - arms control
  - auctions
  - cyber security
- Common problem: real-world games are typically incomputable
- Solution: Computational Game Theory

Approximating Incomputable Games

- Consider the space of each user's actions

Approximating Incomputable Games

- Consider the space of each user's actions
- Perform local search in these spaces

Approximating Incomputable Games

- Consider the space of each user's actions
- Perform local search in these spaces
- Solution quality in one space is dependent on the search in the other spaces

Approximating Incomputable Games

- Consider the space of each user's actions
- Perform local search in these spaces
- Solution quality in one space is dependent on the search in the other spaces
- The simultaneous search of co-dependent spaces is naturally modeled as an armsrace

Classical Computational Solver Limitations

Complex real-world problems can be (practically) unsolvable with classic approaches

- Black box

Classical Computational Solver Limitations

Complex real-world problems can be (practically) unsolvable with classic approaches

- Black box
- "Ill-behaved" search space

Classical Computational Solver Limitations

Complex real-world problems can be (practically) unsolvable with classic approaches

- Black box
- "Ill-behaved" search space
- Intractable

Classical Computational Solver Limitations

Complex real-world problems can be (practically) unsolvable with classic approaches

- Black box
- "Ill-behaved" search space
- Intractable
- Evolution has a demonstrated ability to solve very complex problems

Coevolutionary Algorithm (CoEA)

CoEAs are a special type of EAs where the fitness of an individual is dependent on other individuals (i.e., individuals are explicitly part of the environment)

Coevolutionary Algorithm (CoEA)

CoEAs are a special type of EAs where the fitness of an individual is dependent on other individuals (i.e., individuals are explicitly part of the environment)

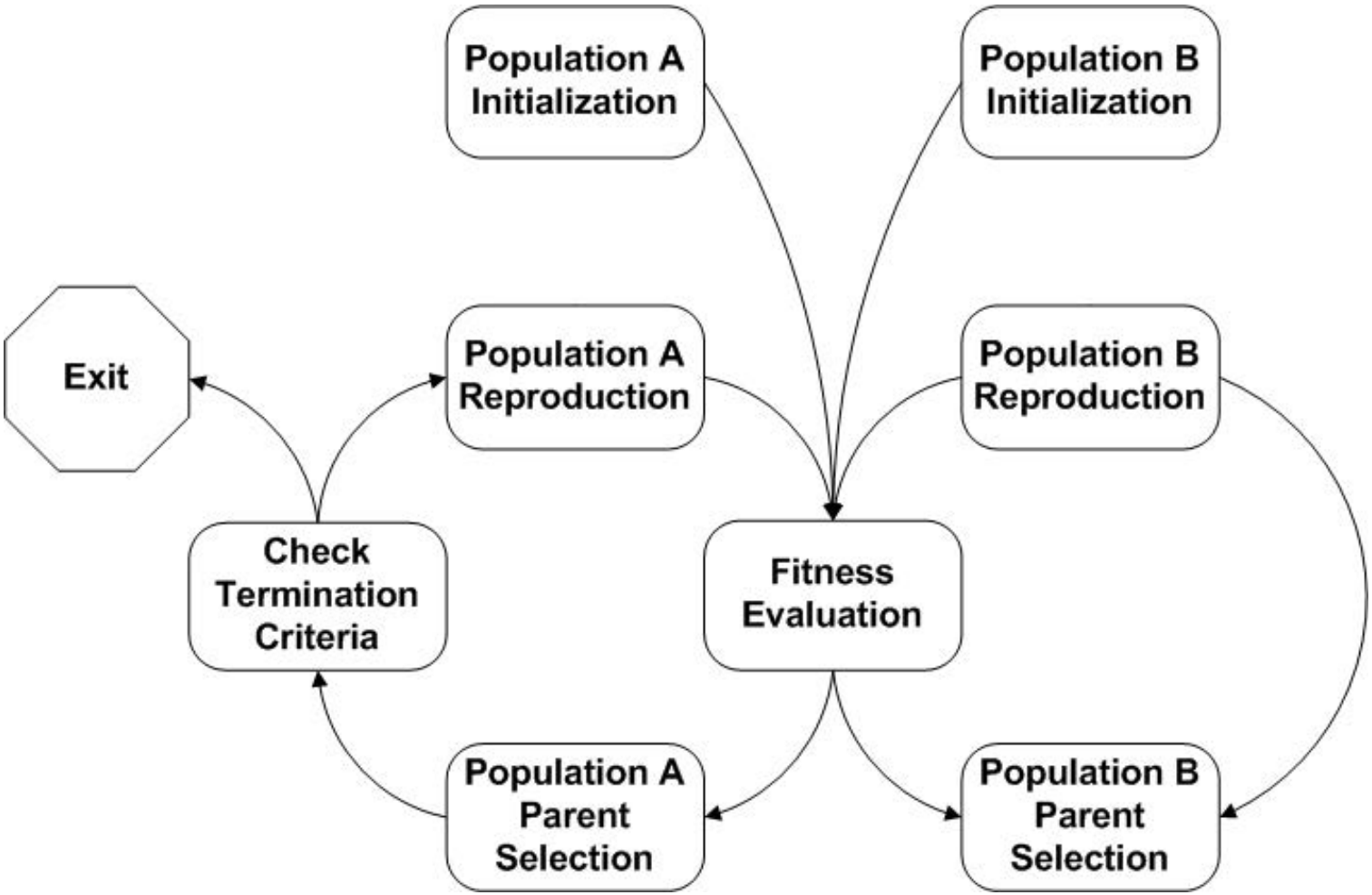
- Single species vs. multiple species

Coevolutionary Algorithm (CoEA)

CoEAs are a special type of EAs where the fitness of an individual is dependent on other individuals (i.e., individuals are explicitly part of the environment)

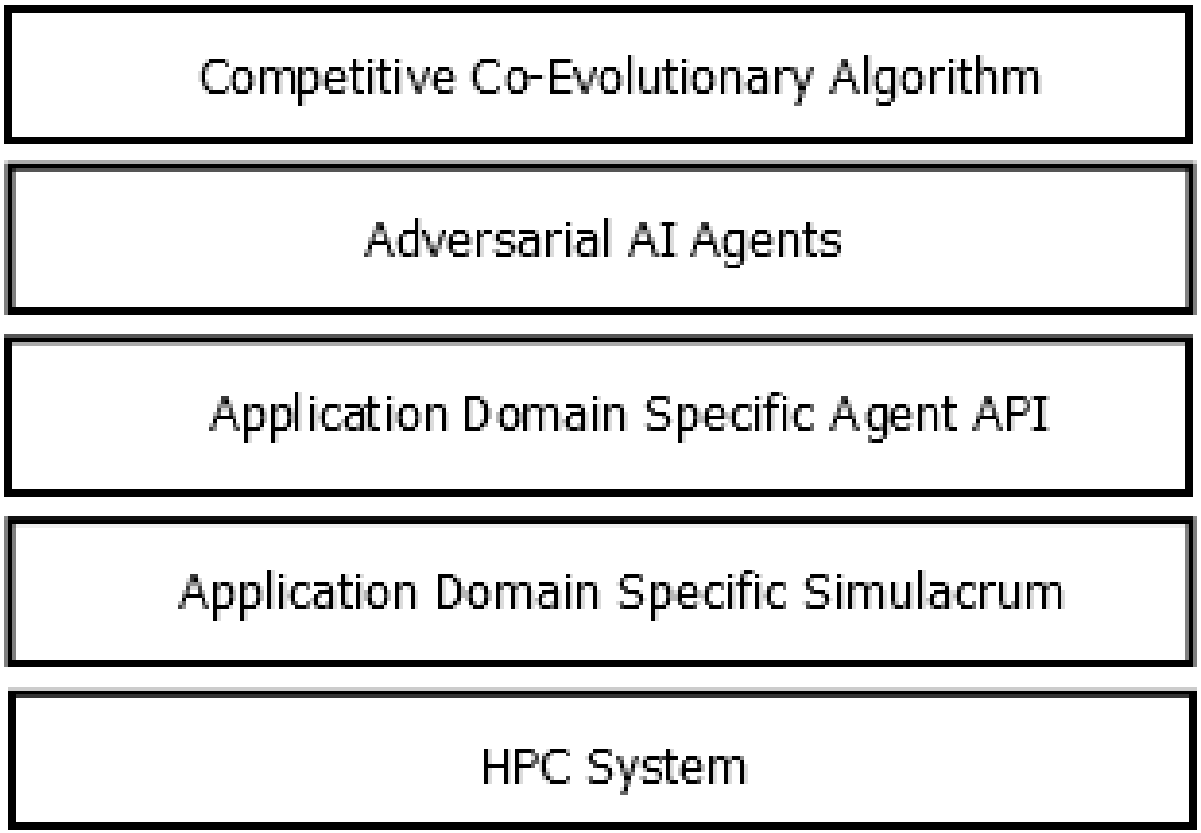
- Single species vs. multiple species
- Cooperative vs. competitive coevolution

Two-Population Competitive Coevolutionary Cycle

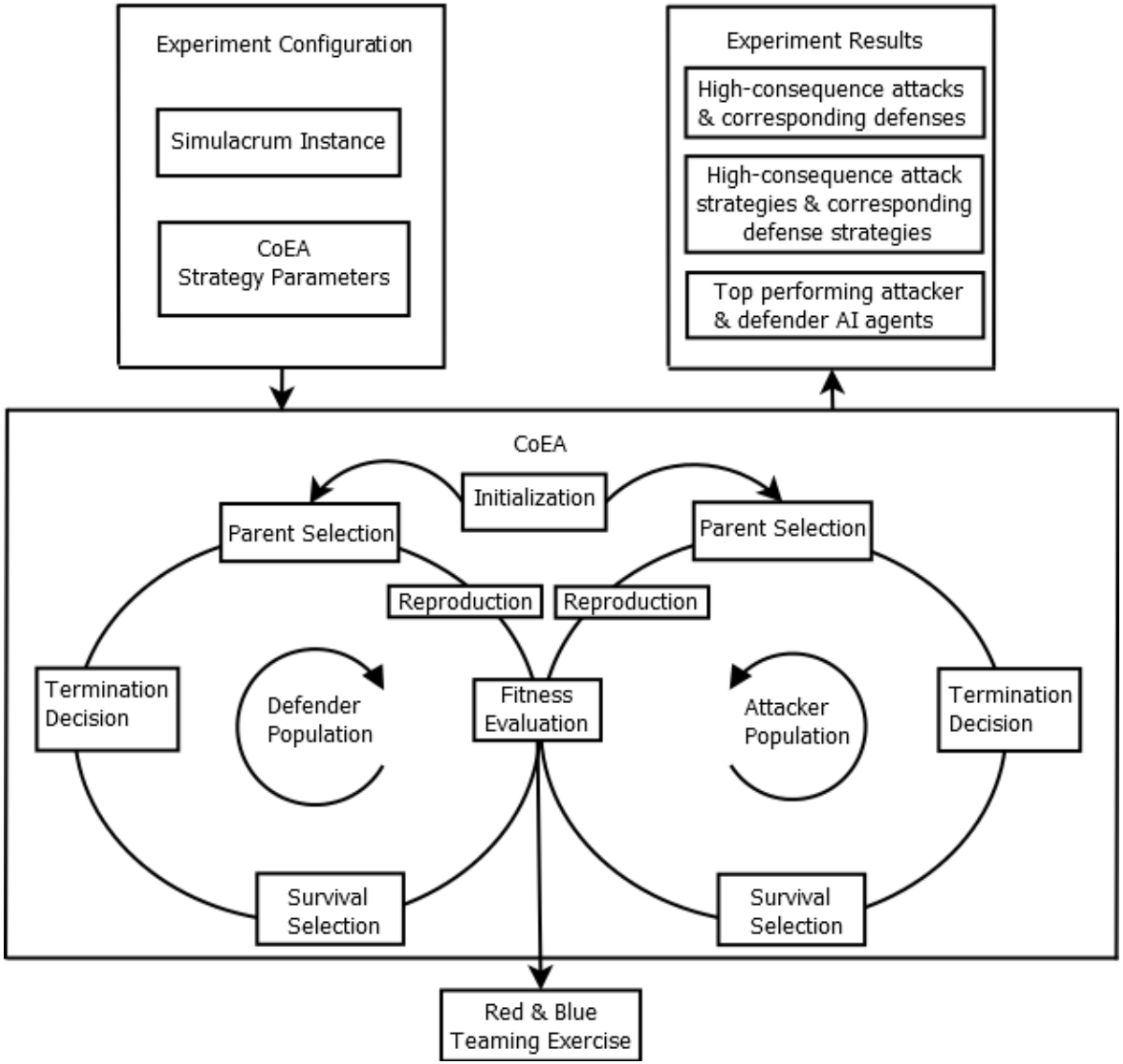


Part III: Engineered System Security through AI Armsraces

CEADS system diagram



CEADS CompCoEA operation



Outcomes

Coevolving attacker and defender AI agents can produce three distinct capabilities:

Outcomes

Coevolving attacker and defender AI agents can produce three distinct capabilities:

Attacks & Defenses Automated identification of vulnerabilities and candidate mitigations that are already tested against a large set of attacks.

Outcomes

Coevolving attacker and defender AI agents can produce three distinct capabilities:

Attacks & Defenses Automated identification of vulnerabilities and candidate mitigations that are already tested against a large set of attacks.

Attack & Defense Strategies Automated wargaming in order to identify high-consequence attack strategies and corresponding defense strategies.

Outcomes

Coevolving attacker and defender AI agents can produce three distinct capabilities:

Attacks & Defenses Automated identification of vulnerabilities and candidate mitigations that are already tested against a large set of attacks.

Attack & Defense Strategies Automated wargaming in order to identify high-consequence attack strategies and corresponding defense strategies.

Attacker & Defender AI Agents Automated generation of highly-trained AI agents that can be deployed in live systems to augment human operators, or even autonomously engage in real-time with adversaries, both human and AI.

How to Apply CEADS to an Engineered System

- Create simulacrum

How to Apply CEADS to an Engineered System

- Create simulacrum
- Design representation for AI agent actions

How to Apply CEADS to an Engineered System

- Create simulacrum
- Design representation for AI agent actions
- Create AI controller logic including sensory inputs

How to Apply CEADS to an Engineered System

- Create simulacrum
- Design representation for AI agent actions
- Create AI controller logic including sensory inputs
- Define attacker & defender fitness functions

How to Apply CEADS to an Engineered System

- Create simulacrum
- Design representation for AI agent actions
- Create AI controller logic including sensory inputs
- Define attacker & defender fitness functions
- Execute AI Armsrace

AU CEADS Efforts

CEADS-LIN Coevolving Attacker & Defender Strategies for Large Infrastructure Networks

AU CEADS Efforts

CEADS-LIN Coevolving Attacker & Defender Strategies for Large Infrastructure Networks

ATLAS-N Adversarial Threats to LArger Satellite Networks

AU CEADS Efforts

CEADS-LIN	Coevolving Attacker & Defender Strategies for Large Infrastructure Networks
ATLAS-N	Adversarial Threats to LArger Satellite Networks
Sat-Tycoon	Satellite Constellation Cyber-Economic Security

AU CEADS Efforts

CEADS-LIN	Coevolving Attacker & Defender Strategies for Large Infrastructure Networks
ATLAS-N	Adversarial Threats to LArger Satellite Networks
Sat-Tycoon	Satellite Constellation Cyber-Economic Security
AMSec-AI-Sabotage	Additive Manufacturing Security

Questions?

Questions?