Computational Challenges in Multiple Adversarial Multi-Agent Teams

Jose B. Cruz, Jr.* and Genshe Chen**

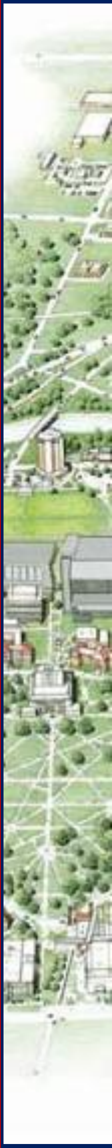
*The Ohio State University, Columbus, Ohio USA

**Intelligent Automation, Inc., Rockville, MD USA

2007 International Conference on Parallel Processing
Xi’an, China. September 12, 2007

Outline of Presentation

- Introduction
- Mathematical dynamical models
- Approaches from game theory; decision and control; and operations research
- Research challenges
- Computational challenges in determining strategies for multi-agent teams.
- Representative application areas



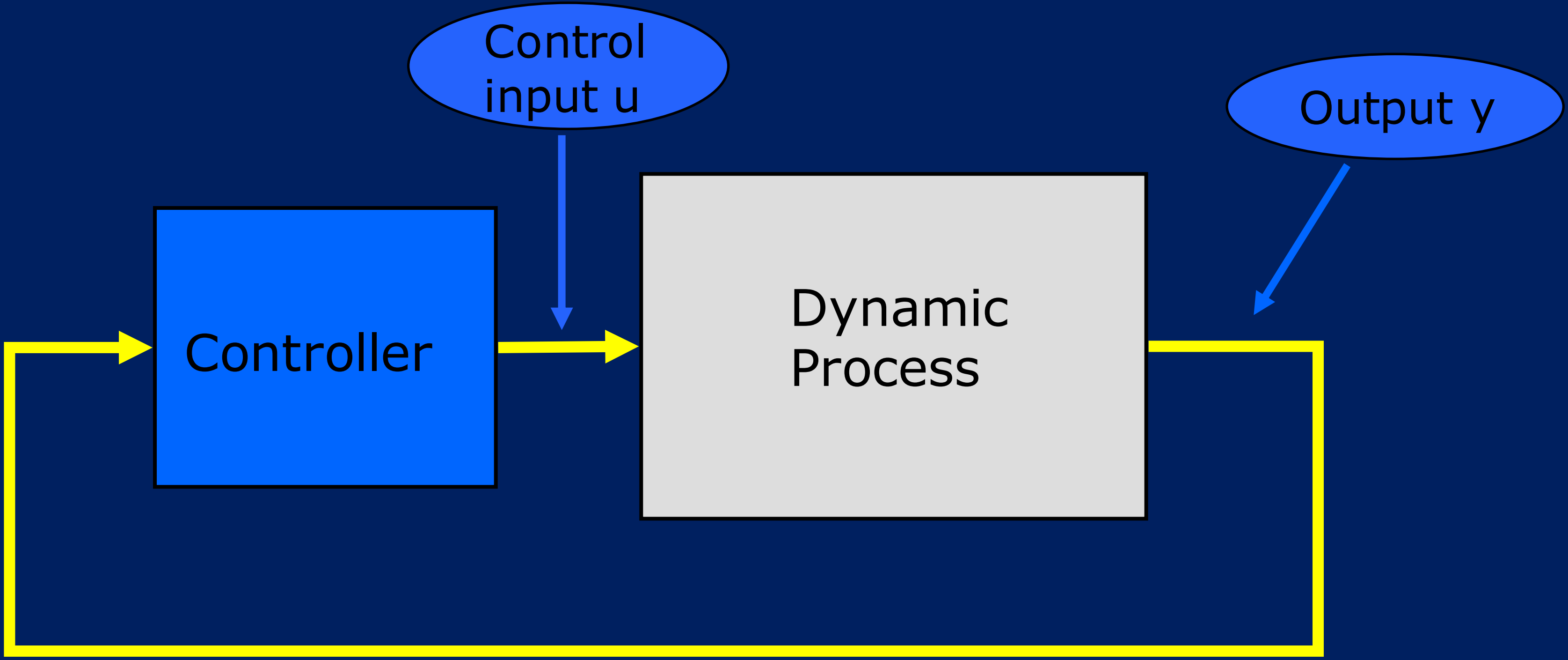
Adversarial Multi-Agent Teams

Sample Application Areas

- Wireless sensor networks in hostile environments
- Warfare on terror
- Anti-Missile Defense
- Battle engagement
- Cyber security of the internet

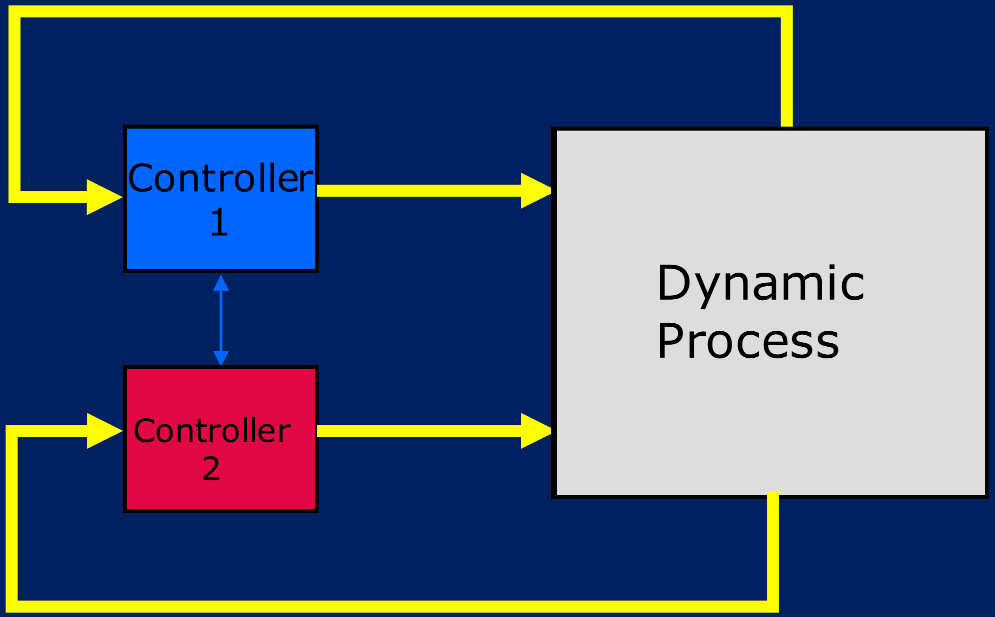
Traditional Centralized Control

- Computer control of industrial process



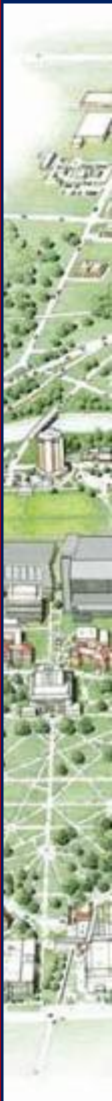
Distributed Control

- *Distributed Control:* There may be a single decision maker or control authority, but there are at least two controllers or agents.



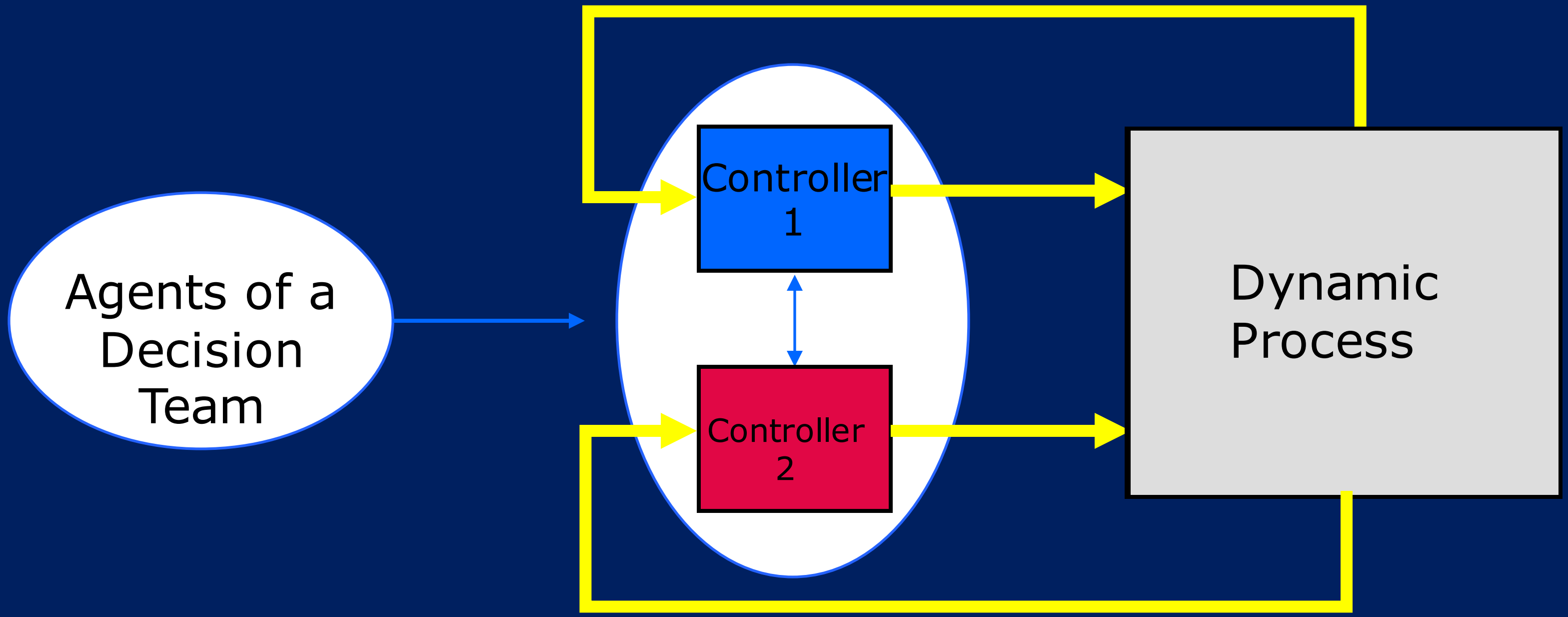
Distributed Control

- Each distributed controller may be a dynamic mapping from a localized output space to a localized control input space.
- There may be limited communication between the distributed controllers.
- Each distributed controller may have a separate objective but it is generally aware of the objectives of the other controllers.
- The distributed controllers are agents of a decision team of a single control authority.



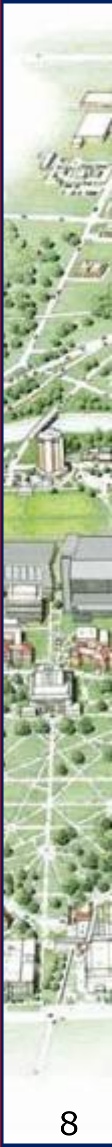
Hierarchical Team Control

- Distributed decision team of multiple agents.
- Agents are parallel controllers

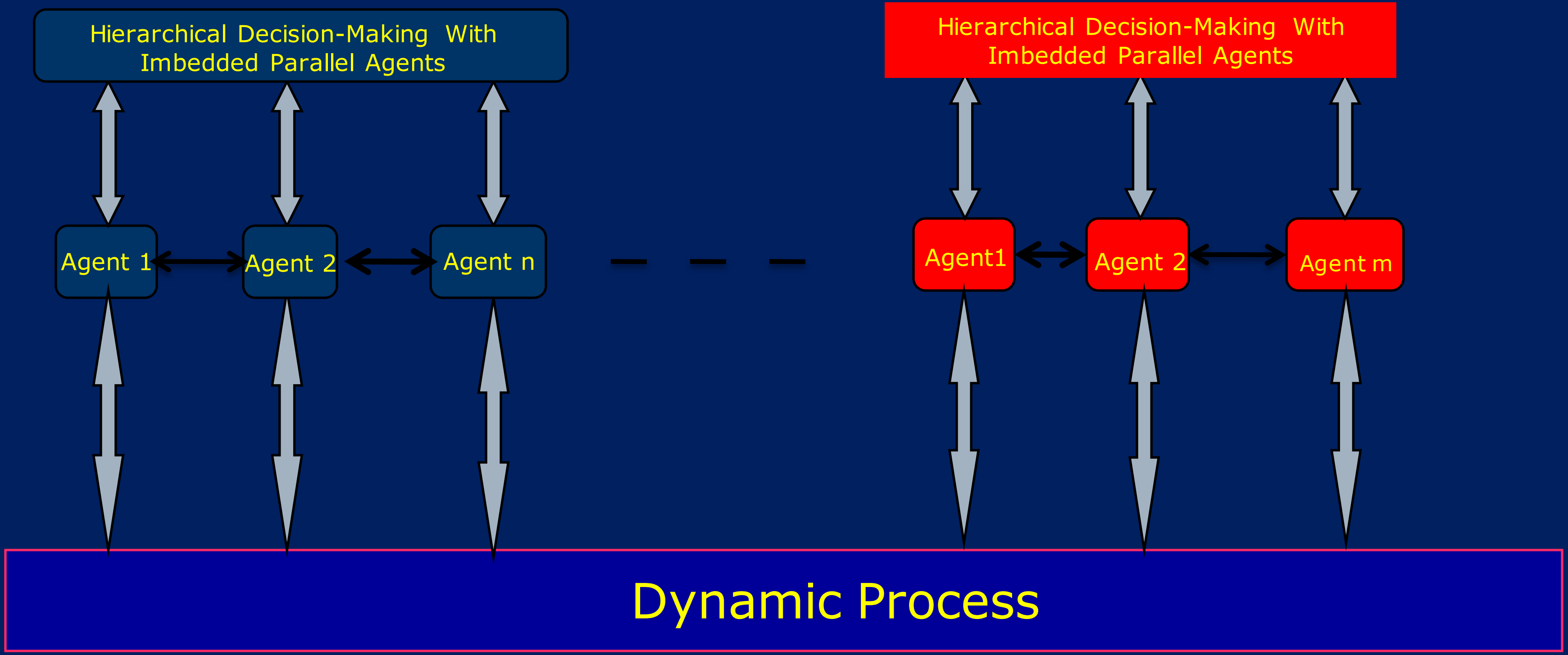


Status of Multi-Agent Team Control Theory and Applications

- Decision and control theory are adequate for planning, system design, and operations of systems with a single team.
- Applications tend to ignore presence of adversarial or competitive elements in system design.
  - Satisfactory when adversaries or competitors are not organized.
  - Not satisfactory in the presence of intelligent adversaries.



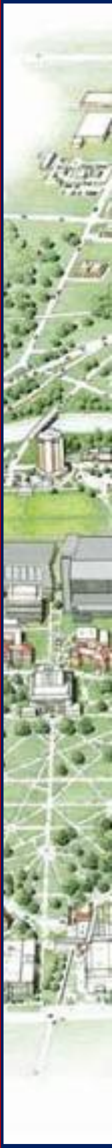
Multiple Decision Making Teams With Hierarchical Parallel Agents


Hierarchical Decision-Making With Imbedded Parallel Agents
Agent 1
Agent 2
Agent n
Hierarchical Decision-Making With Imbedded Parallel Agents
Agent1
Agent 2
Agent m
Dynamic Process

Recent Advances in Control

- Applications drive need for research in control
- Recent application areas involve multiple teams with each team involving multiple agents
- Teams are adversarial or competitive
- Cooperation of team agents needs to be coordinated
- Strategy development and operation is highly computation-intensive

Multiple Adversarial Teams With Multiple Parallel Agents : Scenarios



- Contemporary military conflicts occur in other than level playing fields.
- Surprise, deception, getting destroyed with and by the enemy, asymmetric threats, asymmetric actions and reactions are common occurrences.
- Decision making teams are hierarchical with parallel agents.
- Presence of some agents may be unknown.

Adversarial Hierarchies

- A group called the Blue group may have several cooperating teams,

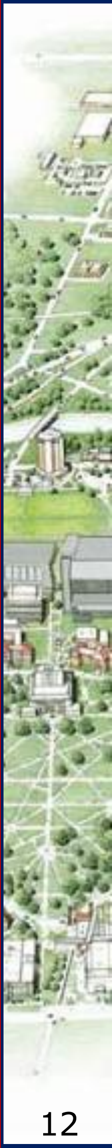
  $T_i^B, \ i = 1, \ 2, \ \ldots \ n_B$
- Each T_i^B may consist of several sub-teams, $T_{ij}^B, \ j = 1, \ 2, \ \ldots m_i^B$
- Each sub-team may consist of individual assets such as UAVs.
- Associated with each T_i^B and T_{ij}^B is an objective function

  J_i^B or J_{ij}^B to be optimized in cooperative fashion:
- $J^B = \sum_{i=1}^{n_B} \alpha_i J_i^B, \quad J_i^B = \sum_{j=1}^{\mathrm{m}_\mathrm{i}^\mathrm{B}} \alpha_{ij} J_{ij}^B$
- A second group called the Red group may have a similar hierarchy.

The Blue and Red groups are adversarial..

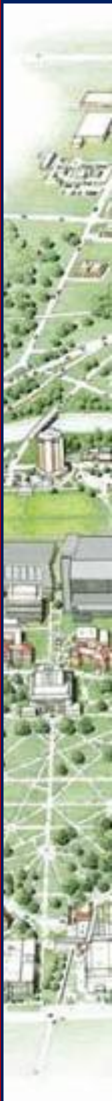
- There may be a "neutral" group (in some applicatin areas

this is the civilian population) and it may have a hierarchy.



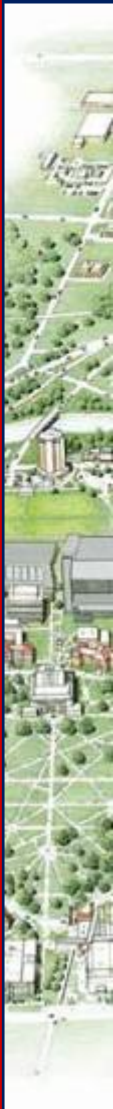
Team Dynamics and Tactics

- Each Blue team may be engaged with an adversarial Red team.
- Each Blue team or sub-team may be reassigned during battle to join a new team and assume new tasks and new schedules.
- Non-cooperative Nash game strategies are used to deal with an adversary.
- Pareto-optimality strategies are used for team and agent coordination.



Cooperation Among Parallel Agents

- Limited communication among parallel agents.
- Limited communication with higher level decision maker.
- Utilization of Pareto-optimality.



Mathematical Model for State Evolution

- There is a state vector that evolves in time (discrete or continuous)

$x(t)$ or $x(k)$

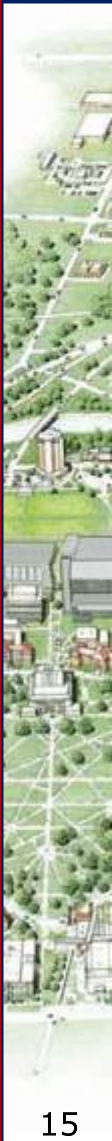
- There is a state transition rule that de termines the next state as a function of the current state and the co ntrol vectors of the blue group, the red group, and the white group

$x(k+1) = f[x(k), u^B(k), u^R(k), u^W(k)]$ or

$$\frac{dx(t)}{dt} = f[x(t), u^B(t), u^R(t), u^W(t)]$$

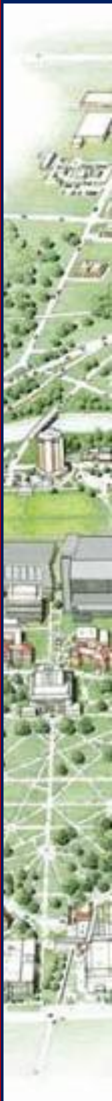
- There is an observation or measurement of the state

$y^i(k) = h^i[x(k)] + v^i(k)$, for discrete time where $v^i(k)$ is noise or uncertainty in the measurement, and $i = B, R,\ or\ W$



Nature of Objective Functions

- Each agent or team cost or benefit (pay-off) may be a sum of additive time-stage functions over a finite time horizon, N_T making the optimization a complicated dynamic optimization.
- $$J^{ij} = \sum_{m=k}^{N_T} L^{ij}[x(m), u^B(m), u^R(m), u^W(m)]$$

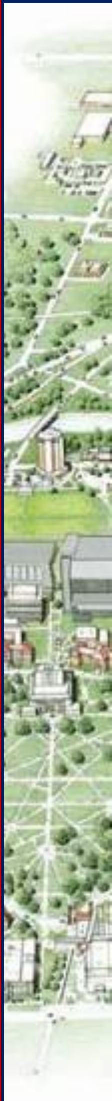


N-person Dynamic Game Theory

- The body of knowledge for the study of such systems is called dynamic game theory and the participants or decision-makers are called players. In our formulation there are three groups of players but in general this may be N.
- For continuous time the game is called a differential game
- Differential games were first studied by Isaacs in the context of two-person pursuit-evasion games.

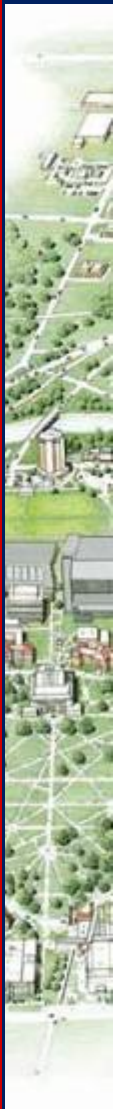
Pursuit-Evasion Game

- A two-person pursuit evasion game ends when the pursuer captures the evader.
- For multiple pursuers and multiple evaders the game becomes extremely difficult
  - When does the game end?
  - Which pursuer pairs with which evader?
  - Can two or more pursuers go after an evader?
  - Computation is extensive.
  - General theory is only beginning to be developed.



Status of Research in Multi-player Pursuit Evasion Differential Games

- Starting from a valid suboptimal solution for the value of the game, a sequence of suboptimal solutions can be constructed.
- The sequence converges and its limit is the solution to the differential game.
- Each iteration requires solving a set of well-defined optimization problems, but still numerically difficult problems.
- We have a procedure for constructing the initial optimal solution.



Challenges in Theoretical Development

- Most of current theory assumes that each player knows the objective function of each player. Estimation theory needed.
- How can deception be detected early?
- How can additional teams/agents be detected?

Computational Challenges

- Need scalable software for multi-player differential games
- Need practical software for stochastic games
- Need to develop an efficient algorithm for solving the optimization problems in the iteration process

War on Terror

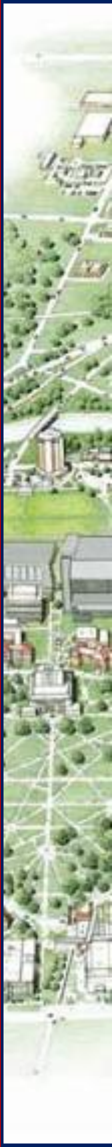
Sample application area:

- □ Asymmetric threat prediction using spatial and time features prediction with game theory
- □ Minimizing incidence of terror attacks
- □ Minimizing damage from terror attacks

Using features in prediction

- Very early prediction models (Model Type I)
  - Calculate crime frequencies
- Later models (Model Type II)
  - Analyze possible crime preferences or features, such as population density, income per capita, distance to police station, etc.
  - Fuse such analyses in prediction, typically
    - Statistically summarize features
    - Statistically apply features in probability models
  - Achieved great improvement on accuracy of city district crime predictions
- Features can greatly refine the predictions.

When to employ game theory



- When the enemies are unorganized and non-intelligent, Courses of Action (COAs) will be somewhat independent, of enemy activity.
- If the enemy is well-structured and has an intelligent organization, the scenario will be largely different.
  - Intelligent enemy’s behavior might not have strong randomness.
  - The enemy might purposely choose COA time and site, perform such COA, calculate the loss and gain of the last stage, then determine the next stage’s action.
  - The enemy might choose a different site for every stage, which will not display any traditional “geographical preference”.
- Intelligent enemy might (suddenly) change preferences or behavior features.

Why do we need game theory?

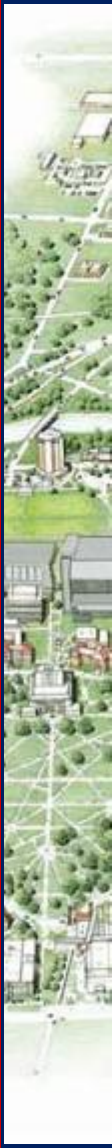
- Model Type II assumes that the features are fixed once they are identified:
    - If “the distance to gas station” is an effective feature, this feature will always be taken into account even if later terrorists change their pattern so that “the distance to a school” should be the new feature.
    - Terrorists assume that the old feature is known and continuing the old Course of Action (COA) pattern entails too high risks, thus changing their patterns of behavior.

Time to abandon old method

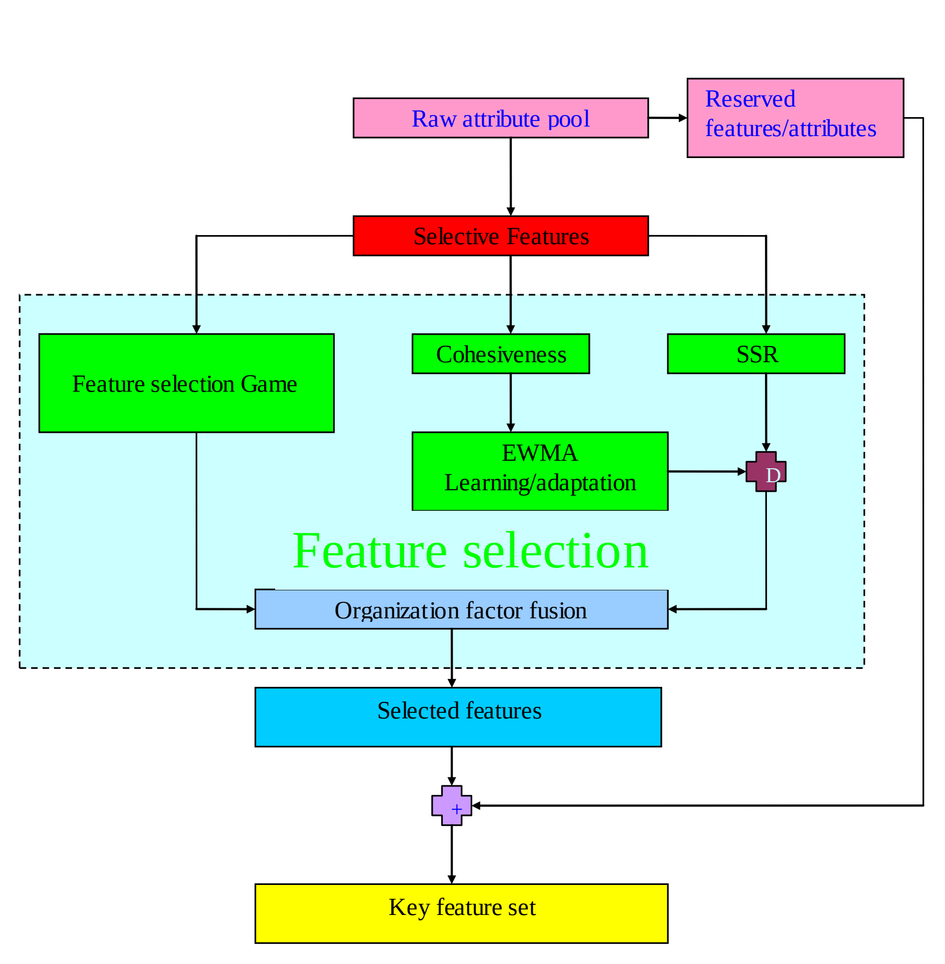
- Model type II can not efficiently deal with possible changes of COA features.
  - Even if after each time step the features should be chosen again, there will still be significant delay in identifying such changes of features, because the old method for identifying effective features is based on statistical data.
  - Only after the changes happen long enough is it possible to detect such changes.

Advantages of game theory

- Applying game theory can help predict possible changes of features
  - The basic logic of game theory is to predict ahead via all available information, including past data and possible choices at current stage.
  - It does not need to wait for the enemy's change happening first thus no delay.
  - Such prediction is often self-enforcing due to the properties of Nash solutions.
- Via game theory, surprise attacks can be reduced.



Advanced Hybrid Feature Selection

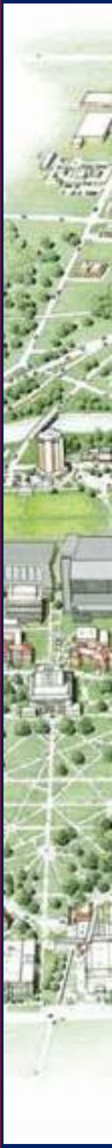


Key feature set consists of :

- Reserved Feature Subset
- Selective Feature Subset

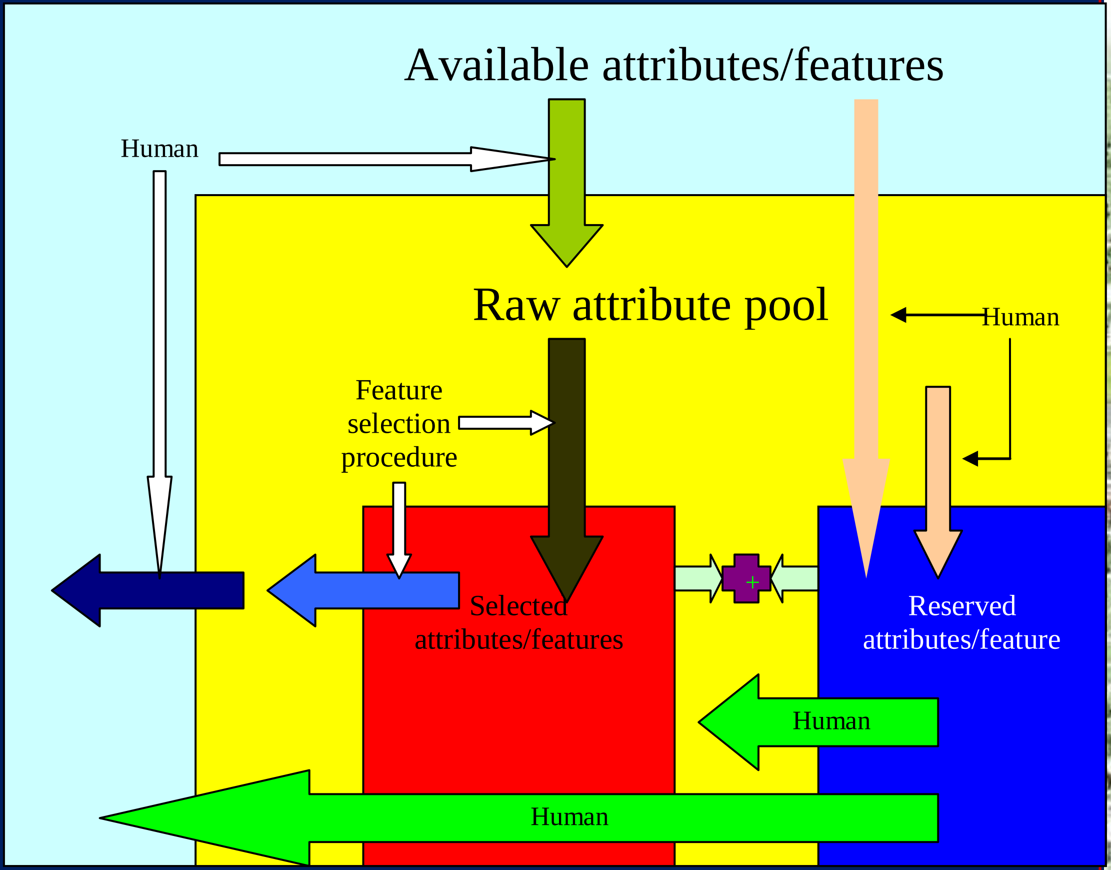
Reserved feature subset is composed of very important features which should not be ignored at any time.

Selective feature subset is automatically selected by the feature search algorithm in the “Feature selection” block.



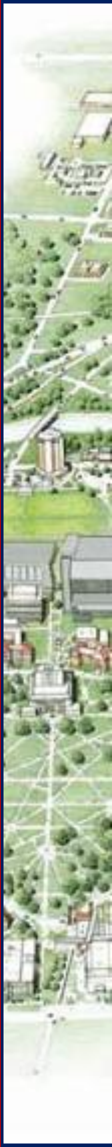
Feature Storage

- Selected features will be placed in the inner core of the ontology which stores the features and the relationships among them.


Available attributes/features
Human
Raw attribute pool
Human
Feature selection procedure
Selected attributes/features
+
Reserved attributes/feature
Human
Human

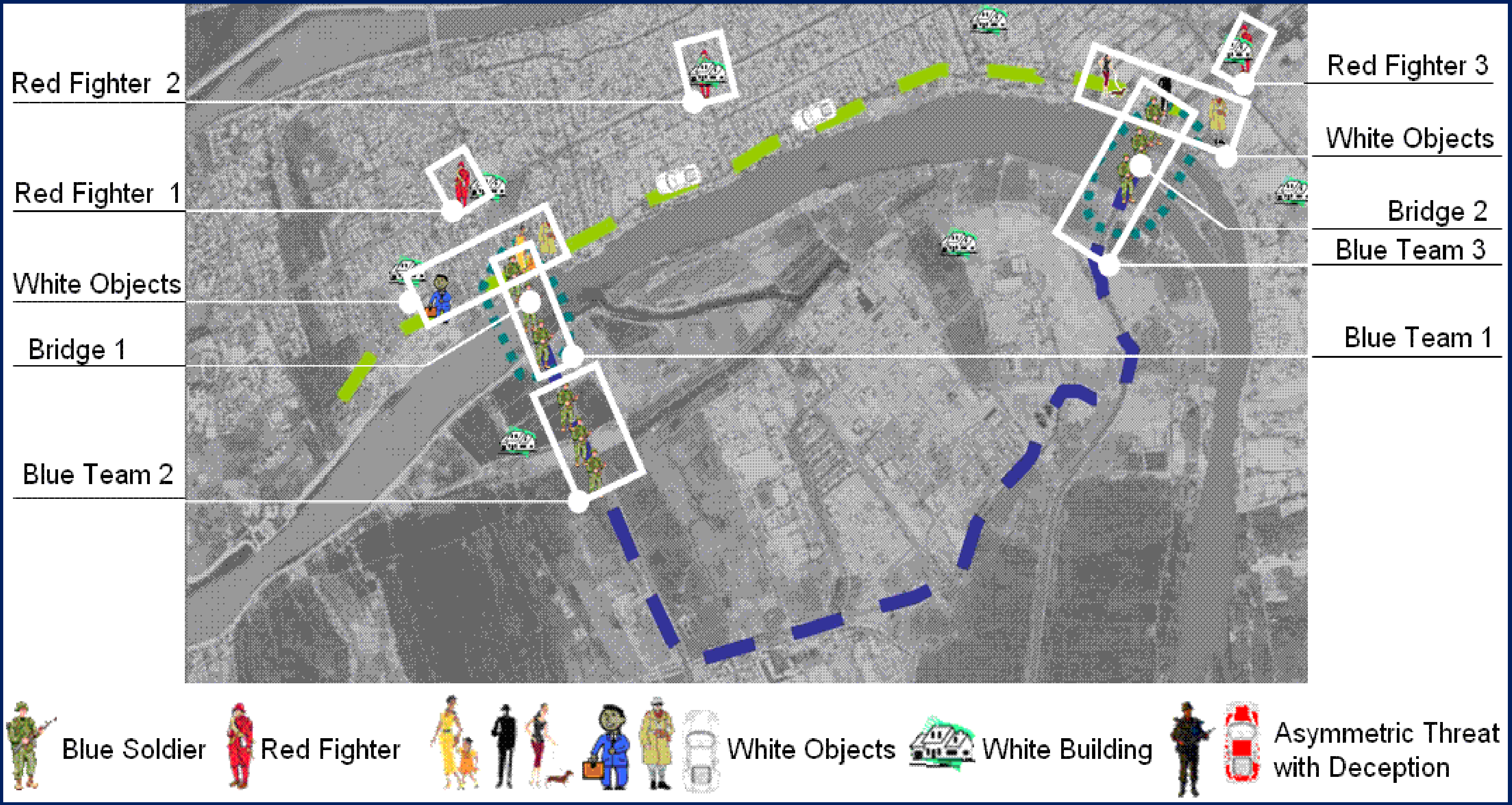
Advantages over traditional approaches

- No need to discard features “that do not exhibit enough variation in the event feature data set”
  - Such features are not convenient for traditional probability approach
  - High concentration does not necessarily mean low prediction
  - Our game method can make use of such features
- Refined kernel probability functions in estimation
  - Problems of traditional Gaussian distribution approach (use last event feature value as the center-point)
    - Event distribution might be severely asymmetric
    - Many feature values are even one sided
  - Problems of traditional exponential distribution approach (use last event feature value as the starting-point, then decreasing)
    - An intelligent attacker would intuitively avoid exactly the same location/time/features
    - Thus last time’s feature values do not mean the highest possibility
  - Our approach: use double-sided exponential kernel distribution.



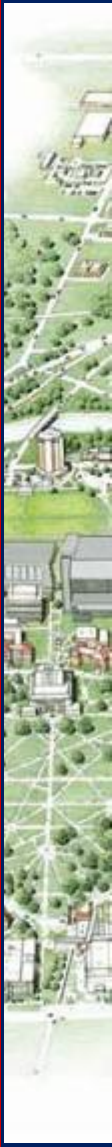
Simulation: Urban Warfare Scenario

- Typical urban warfare scenario to illustrate our dynamic adaptive hierarchical game theoretic approach for modeling and prediction of asymmetric threat learning processes.



Urban Warfare Scenario

- The blue force’s missions: secure the whole area: urban districts, bridges, mains roads and blocks.
- The red force (terrorist and/or insurgent forces) includes armed fighters and some hiding in and acting like white objects (the civilians).
- When battles are long-lasting and the battlefields are heavily populated by civilians
  - Civilian interest: desire “participation”.
  - Civilian intelligence: capable of “participating”.
  - Biased civilians can affect COA success probabilities.



Urban Warfare Scenario: Detailed Strategy description

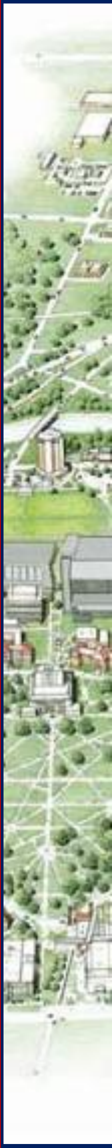
- In urban scenario, we predict the changes in enemy strategies before such changes are fully implemented.
- We present a primitive prediction of ECOAs by following the pattern/feature recognition model.
- Based on such prediction, some associated best response strategies of the Blue side can be recommended.
- If the primitive prediction is almost correct, there are two possible response strategies for the blue group according to different goals.

Urban Warfare Scenario

- If the purpose of blue force is to stop the red forces’ actions, the recommended COA of the blue force is to publicly send a message to the red forces, and suggest that their actions will not work. As a consequence, probably the red forces will change their proposed actions.
- However, if the purpose of the blue force is to set up a trap, the blue group should only maneuver secretly.
- In such cases not only might the red use deceptions, the blue might also use some counter deceptions.
- If the first guess is incorrect (the attack pattern might be new and unknown), our game theoretic data fusion module and dynamic learning module will refine the primitive prediction and update the feature/pattern records.

Urban Warfare Scenario: features

- Classify and identify different ECOAs into a small number of types of surprise attacks with associated features.
- After deciding which type of attacks will likely occur at the next stage with what probability, we develop an appropriate resource allocation algorithm.
- Considering information from different resources (papers, newspapers, reports from Department of Defense: Navy, Marines, Army, Air Force), typical surprise attacks are:
  - Type 1: Gun Fighter/Mortar/Small Arms
  - Type 2: IED (Improvised Explosive Device)
  - Type 3: Kidnap/Hijack
  - Type 4: Robbery/Stealing
  - Type 5: “Dirty” bomber/Bio-attacks



Urban Warfare Scenario: features

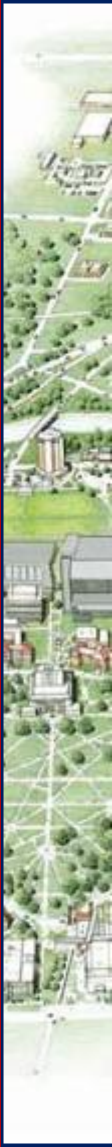
- Any possible attribute (or feature) might be related to another attribute, which means any attribute can serve as a potential feature or pattern.
- Due to real limits such as computation requirements, we can only choose some measurable, available, and “probably” related attributes and place them in a pool of “raw attributes.”
- In such a raw attribute pool there might still exist hundreds or even thousands of attributes, which would greatly exceed the computation capability of existing computer systems since each attribute will serve as a dimension, and when the number of attributes increases the computation will also increase.
- As a result, before associating features into the system, a much smaller key feature set should be dynamically selected from the raw attribute pool.

Partial List of Raw Attribute Pool: Example

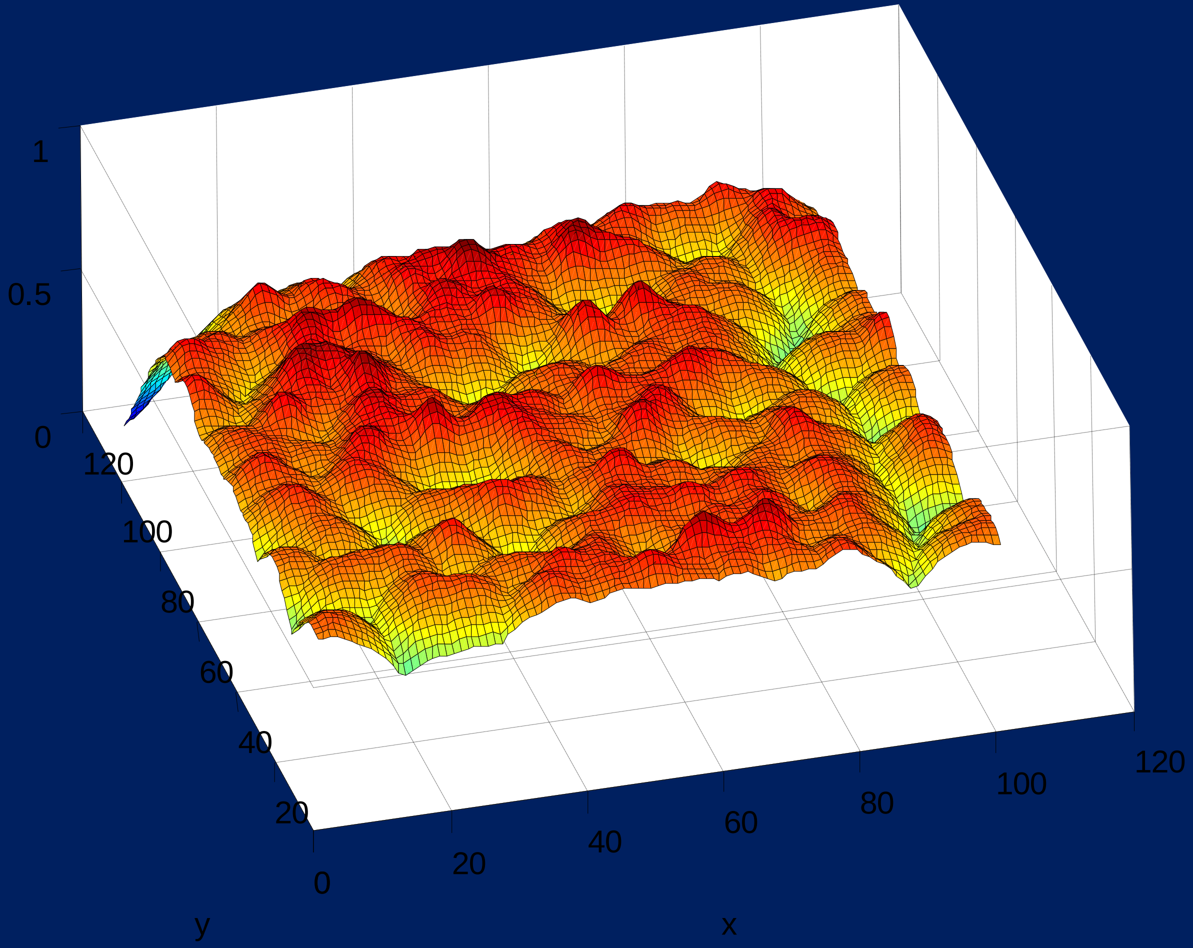
- Population density per square mile
- Religious intensity
- Male people population density per square mile
- Average family size
- Young people (from 11 to 29) population density per square mile
- Average salary per year
- Average price of houses
- Ratio of children in school and out of school
- Percentage of people who were once involved in crimes
- Percentage of people who are in debt
- Average percentage of people who have children
- Distance to nearest soldier/policemen station
- Distance to nearest hospital
- Distance to nearest highway
- Distance to nearest church/school/library
- The time difference from the previous attack
- Distance to nearest location of previous attacks
- Morale of insurgents
- Average wellness of public utilities
- Distance to nearest desert/woods
- Average expenditure on alcohol beverages, tobacco, and smoking

Simulation results for the scenario

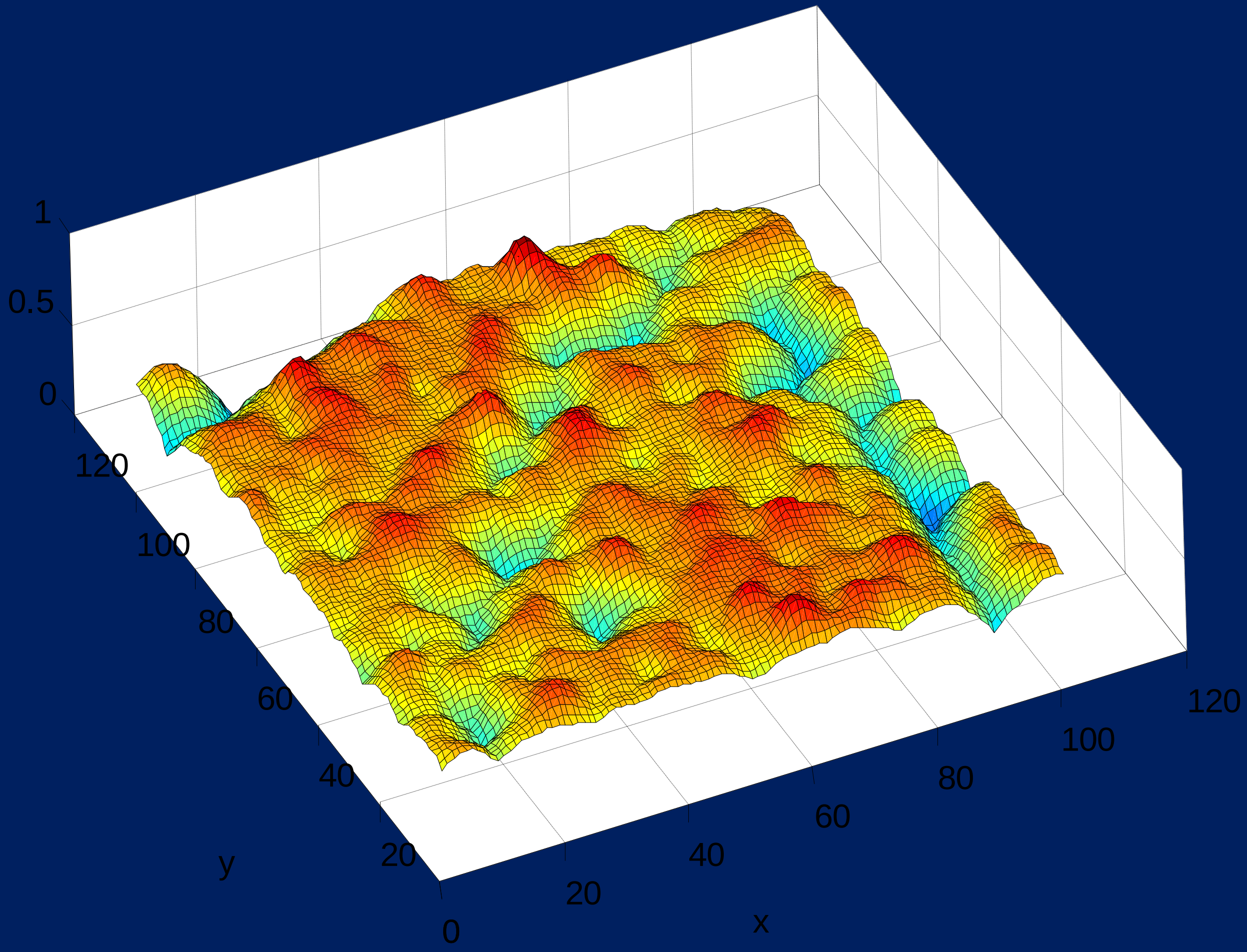
- The final comprehensive probability prediction results (probability maps) in a long duration battle (which can be divided to three time-continuous stages) can be demonstrated in following figures.
- Indices of these three probability prediction maps are arranged in time sequence.
- All the strategies discussed are fused to produce the ECOA threat probabilities over city districts.
- Over the time horizon, new events are fed to the system to update the identified and/or predicted event features/patterns, and finally update the probability predictions.



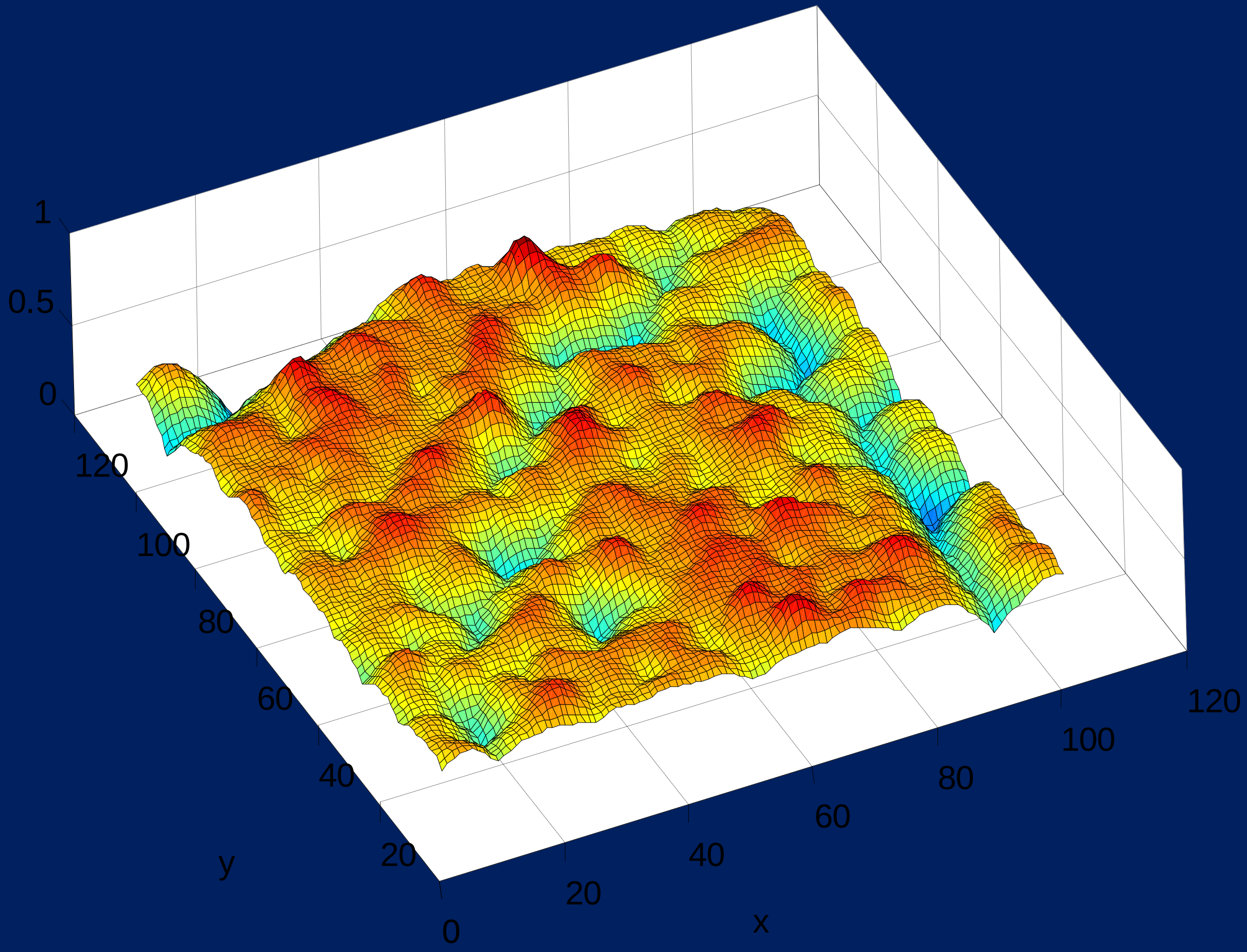
Simulations for Urban Warfare Scenario



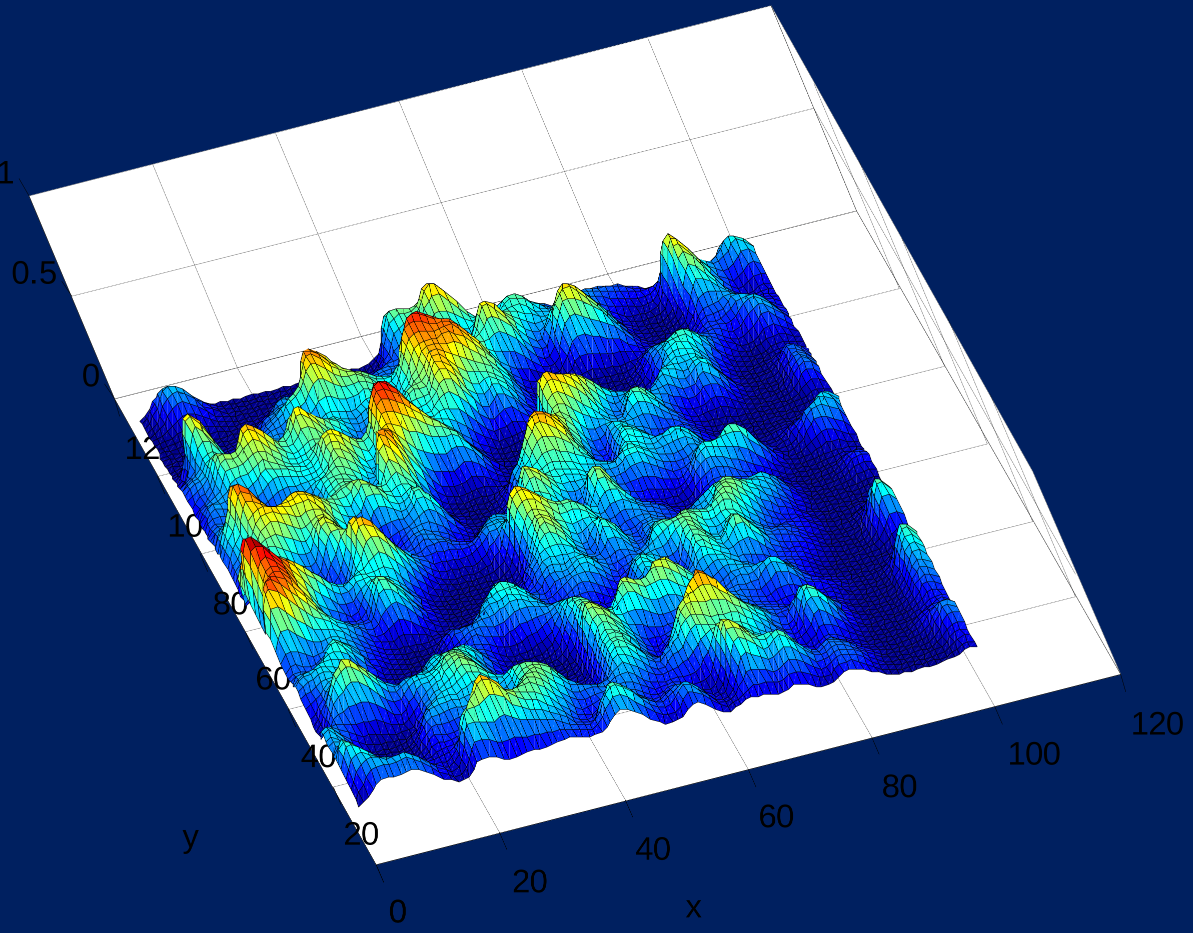
Simulations for Urban Warfare Scenario



Simulations for Urban Warfare Scenario



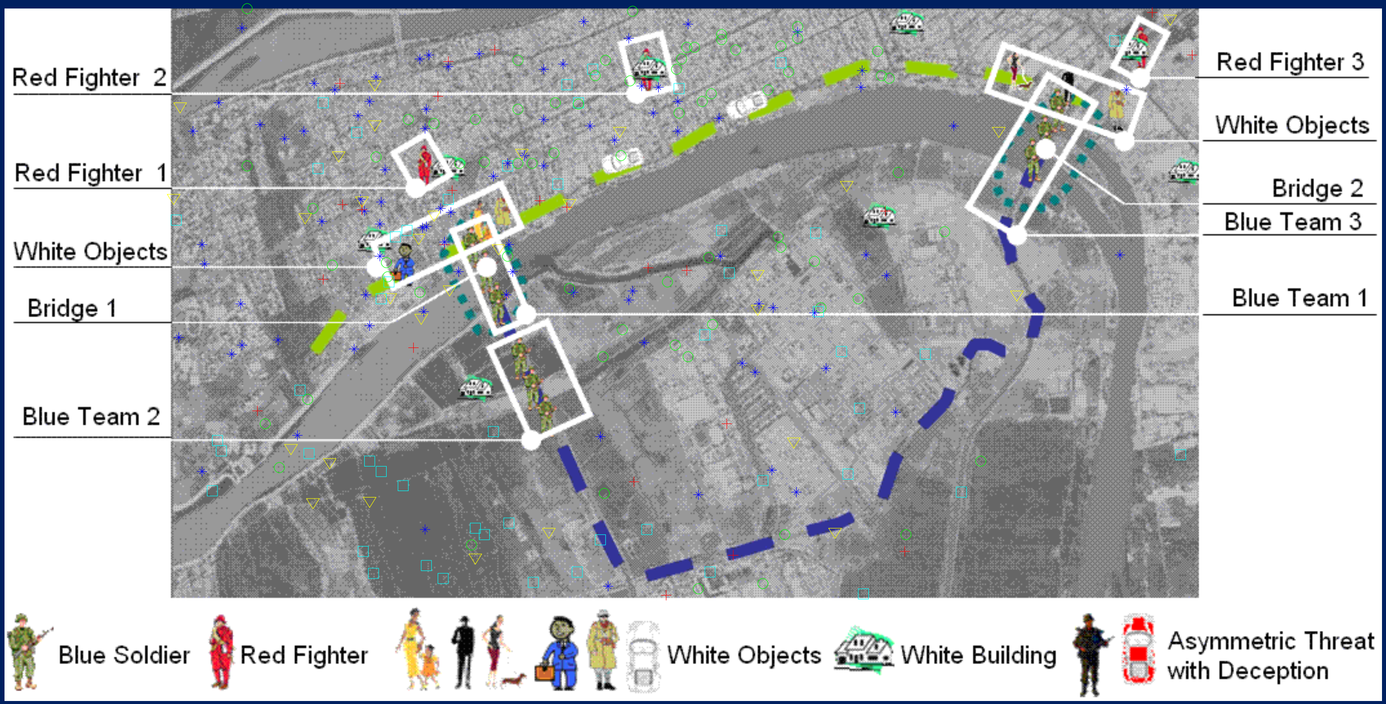
Simulations for Urban Warfare Scenario



Explanations

- Red group changed its preferences.
- Some important features such as population density and morale are always in reserved feature set.
- The Blue group successfully assigned soldier/weapon resources. In the last figure the Red group has lower morale, which is reflected as a general lower probability to have an event for most locations.
  - The river, which is generally not a favorite site for attacks, is also reflected in all three maps.
  - It is still possible to have an attack on the river, which means it might occur on a bridge or boat.

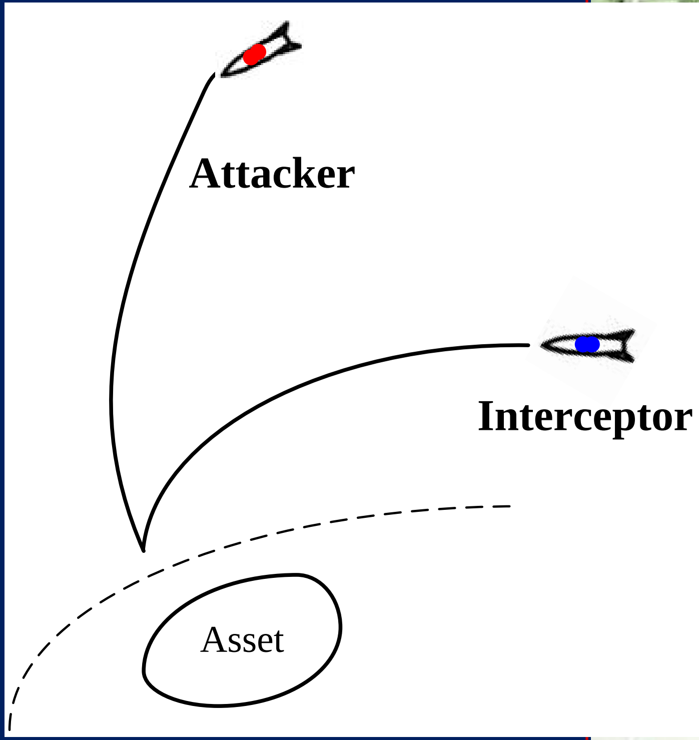
Simulations for Urban Warfare Scenario



Anti-Missile Defense: *sample application area*

- □ Objective: to develop an interception strategy for the pursuer to catch the missile threat before it reaches the asset (missile’s target).
- □ Problem Model
  - ■ Pursuer P; evader E; Asset A
  - ■ The game terminates when the distances

    $d(P,E) \le \varepsilon_P$ or $d(E,A) \le \varepsilon_E$
- □ This is a PE game problem with a state constraint, and solution theory is unavailable.



LQ Formulation

- Game Space $\mathbb{R}^{n_0}$
- Linear Dynamics of the Players
  - Pursuer $\dot{x}_p = A_p x_p + B_p' u_p$
  - Evader $\dot{x}_e = A_e x_e + B_e' u_e$
- Quadratic Objective Functional
  - Projector P: $P(x_p) \in \mathbb{R}^{n_0}$, $P(x_e) \in \mathbb{R}^{n_0}$
  - Objective

$$\begin{aligned} J \;=\; & \int_0^T (u_e^T u_e - u_p^T u_p)\mathrm{dt} + w_e \| P(x_e(T)) \|^2 \\ & - w_p \| P(x_p(T)) - P(x_e(T)) \|^2 \end{aligned}$$

$w_p, w_e > 0$ are linear weights; T is fixed.

We use soft constraints

$w_e \| P(x_e(T)) \|^2$ and $w_p \| P(x_p(T)) - P(x_e(T)) \|^2$ as penalty.

Zero-sum Game

- □ Linear state feedback strategy
- □ Zero-sum Game (**Pursuer-maximizer; Evader-minimizer**)
- □ Objective function

$$J = \int_0^T (u_e^\top u_e - u_p^\top u_p)\mathrm{dt} + x^\top(T)Q_f(w_p, w_e)x(T)$$

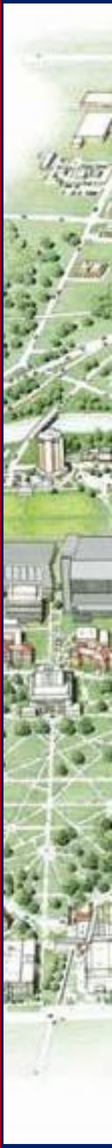
Theorem: The game admits a feedback saddle-point solution given by

$$u_e^*(t) = -B_e^\top Z(t)x(t),\ u_p^*(t) = B_p^\top Z(t)x(t),$$

where $Z(t)$ is bounded, symmetric and satisfies

$$\dot{Z} + A^\top Z + ZA - Z(B_e B_e^\top - B_p B_p^\top)Z = 0 \text{ with}$$
$$Z(T) = Q_f(w_p, w_e). \qquad (1)$$

The existence of solutions for the Riccati equation (1) can be proved under certain conditions.



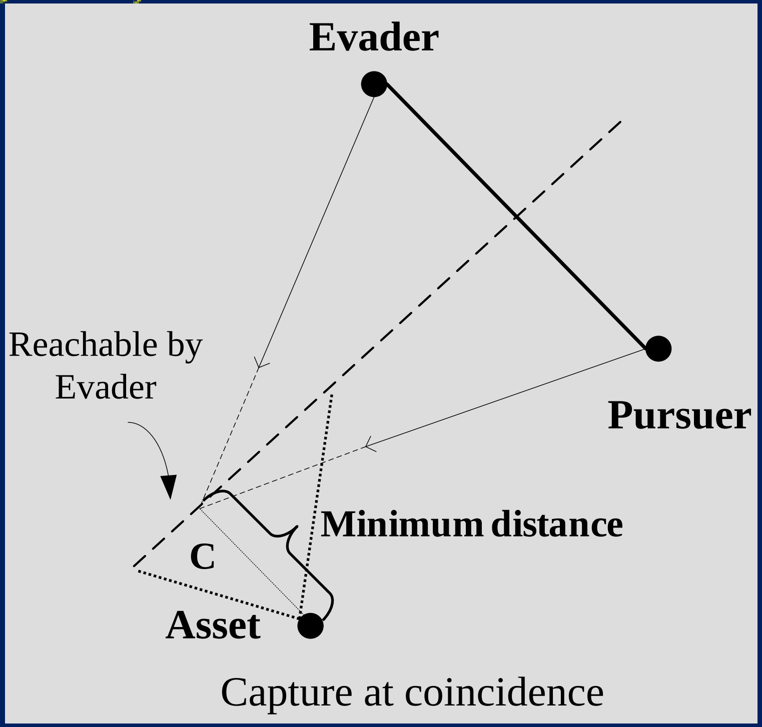
Simple Motion Dynamics

Dynamics: $(\varsigma \in \{p, e\})$

$$\dot{x} = v_\varsigma \cos\theta$$

$$\dot{y} = v_\varsigma \sin\theta$$

$$v_p = v_e = v$$



Point C is the place closest to the Asset that can be reached by the evader. P/E Optimal Strategy: proceed to C.

Use simplified dynamics

$$\dot{x} = u_x \qquad\qquad (2)$$

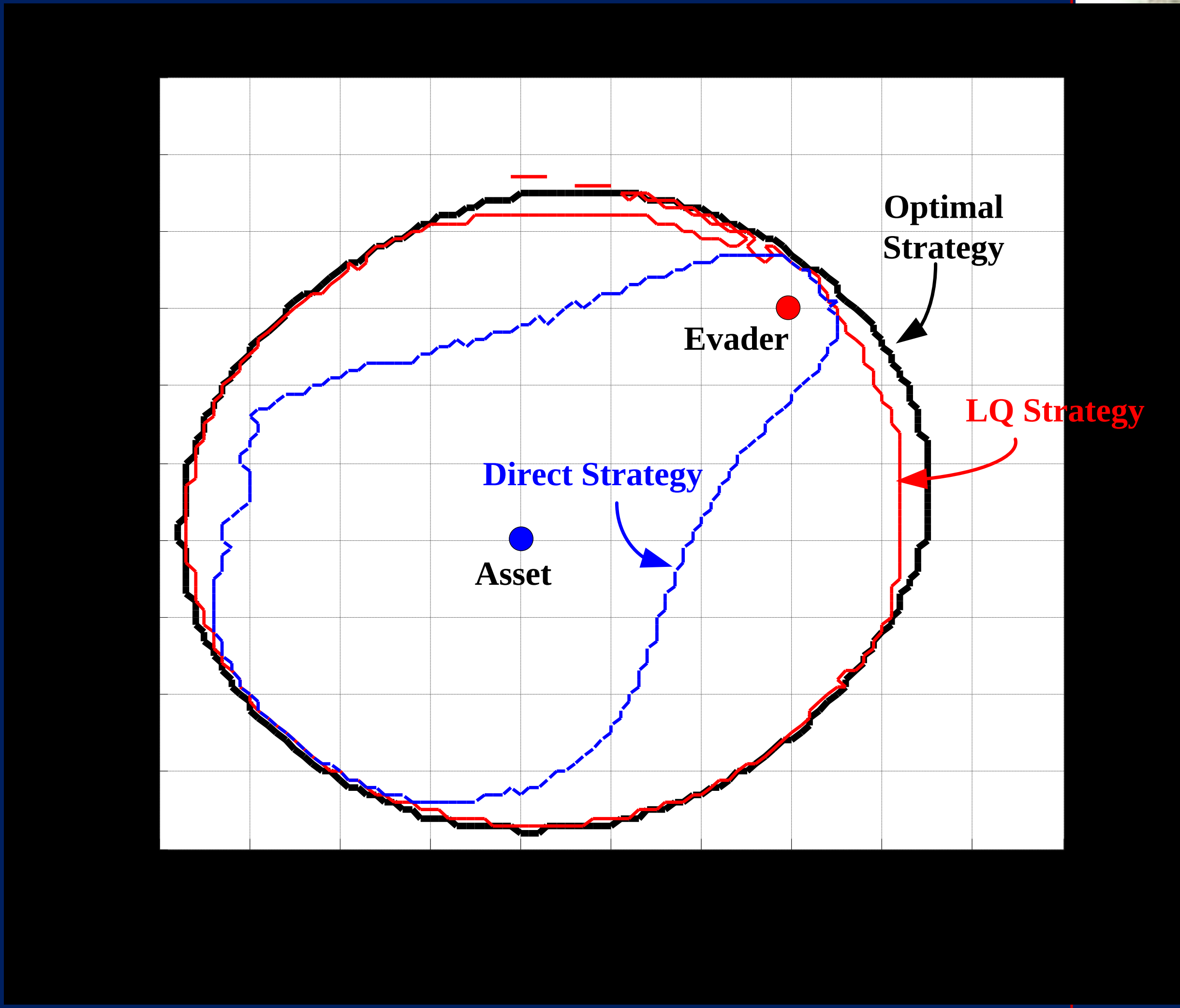
$$\dot{y} = u_y$$

Theorem (Existence of Solution)
Given the dynamics of the players in (2), the Riccati equation (1) has a bounded solution.

Applying the optimal strategy of the evader on the left, we can verify the performance of the LQ strategy.

Performance Verification Pursuer's Strategy

- The evader starts at (3,3), and uses its optimal strategy.
- The pursuer applies one of three strategies:
  1) direct strategy (line of sight):
  2) LQ Strategy;
  3) Optimal Strategy.

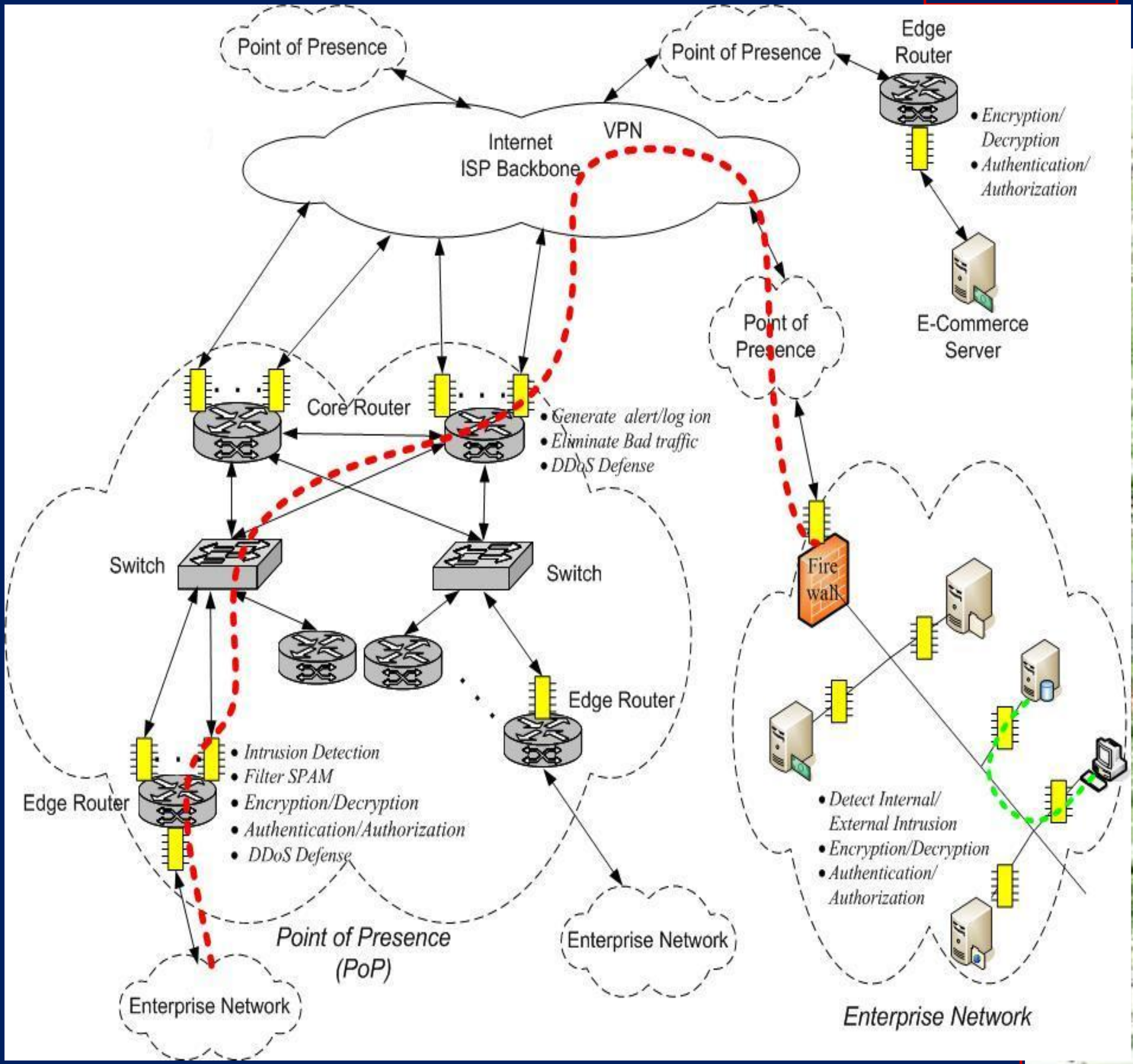


Current Complicating Challenges

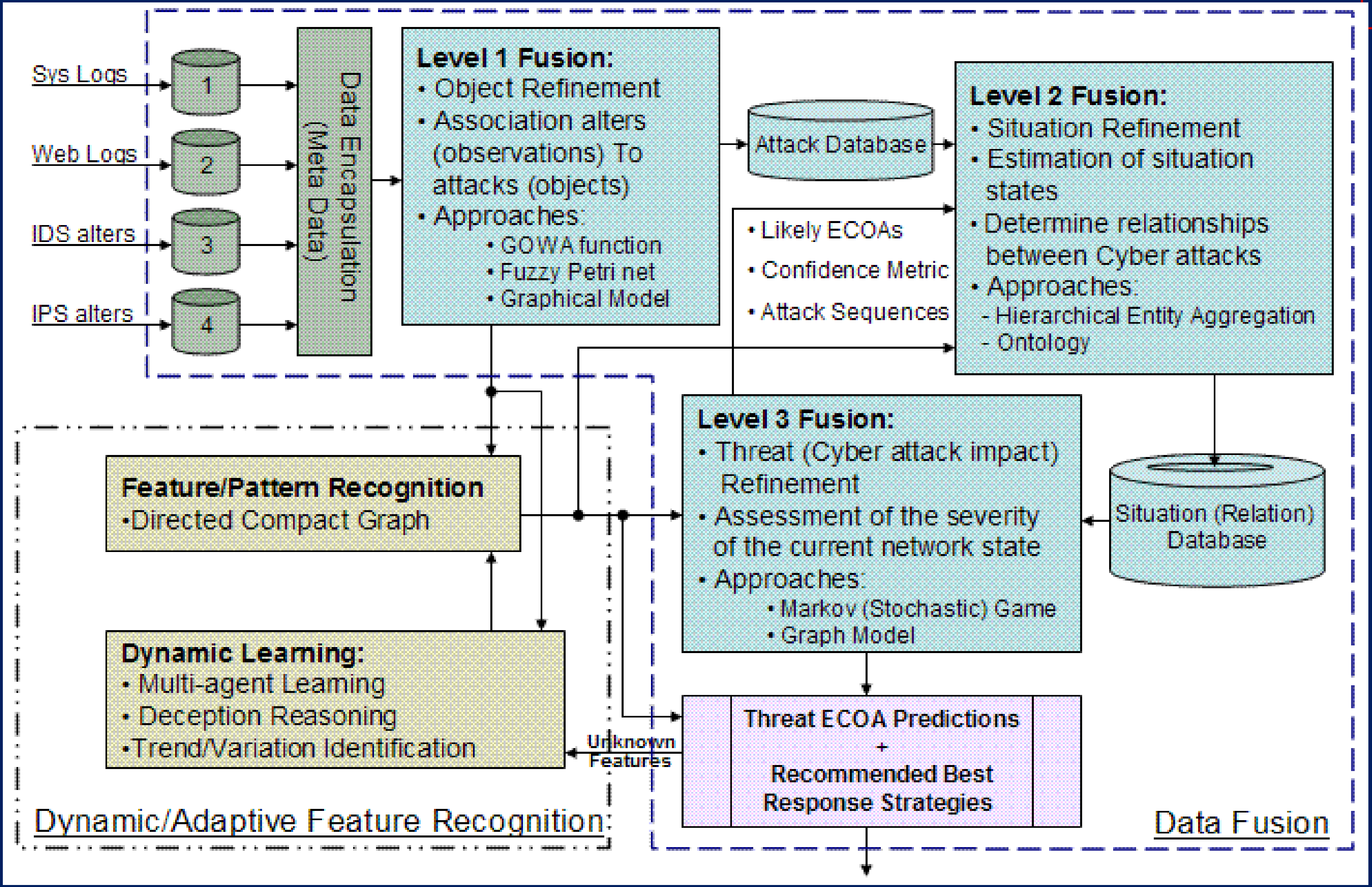
- Multiple Attackers (evaders)
- Unknown number and location of attackers
- Multiple assets to be protected
- Multiple defenders (pursuers)

Cyber Security of the Internet:
sample application area

- Cyberspace security requires next-generation network management and intrusion detection systems.
- These systems should combine both *short-term sensor information* and *long-term knowledge databases* to provide decision-support and cyberspace command and control.
- Information fusion and data mining used to detect and predict the multistage stealthy cyber attacks.



System Architecture



Sys Logs

Web Logs

IDS alters

IPS alters

1

2

3

4

Data Encapsulation (Meta Data)

Level 1 Fusion:
- Object Refinement
- Association alters (observations) To attacks (objects)
- Approaches:
  - GOWA function
  - Fuzzy Petri net
  - Graphical Model

Attack Database

- Likely ECOAs
- Confidence Metric
- Attack Sequences

Level 2 Fusion:
- Situation Refinement
- Estimation of situation states
- Determine relationships between Cyber attacks
- Approaches:
  - Hierarchical Entity Aggregation
  - Ontology

Feature/Pattern Recognition
- Directed Compact Graph

Dynamic Learning:
- Multi-agent Learning
- Deception Reasoning
- Trend/Variation Identification

Unknown Features

Level 3 Fusion:
- Threat (Cyber attack impact) Refinement
- Assessment of the severity of the current network state
- Approaches:
  - Markov (Stochastic) Game
  - Graph Model

Situation (Relation) Database

Threat ECOA Predictions
+
Recommended Best Response Strategies

Dynamic/Adaptive Feature Recognition

Data Fusion

Key components of cyberspace security system

- Our cyberspace security system has two coupled major parts:
  - Data fusion module (to refine primitive awareness and assessment; to identify new cyber attacks);
  - Dynamic/adaptive feature recognition module (to generate primitive estimations; to learn new identified new or unknown cyber attacks; to detect computer network penetration).

Key components of cyberspace security system

- □ Various logs and alerts are fed into the L1 data fusion component.
- □ Fused objects are used by a feature/pattern recognition module to generate primitive prediction of intents of cyber attackers.
- □ High-level (L2 & L3) data fusion Markov game models refine the primitive prediction.
- □ Captured unknown/new cyber attack patterns are associated with related L1 results in dynamic learning block.

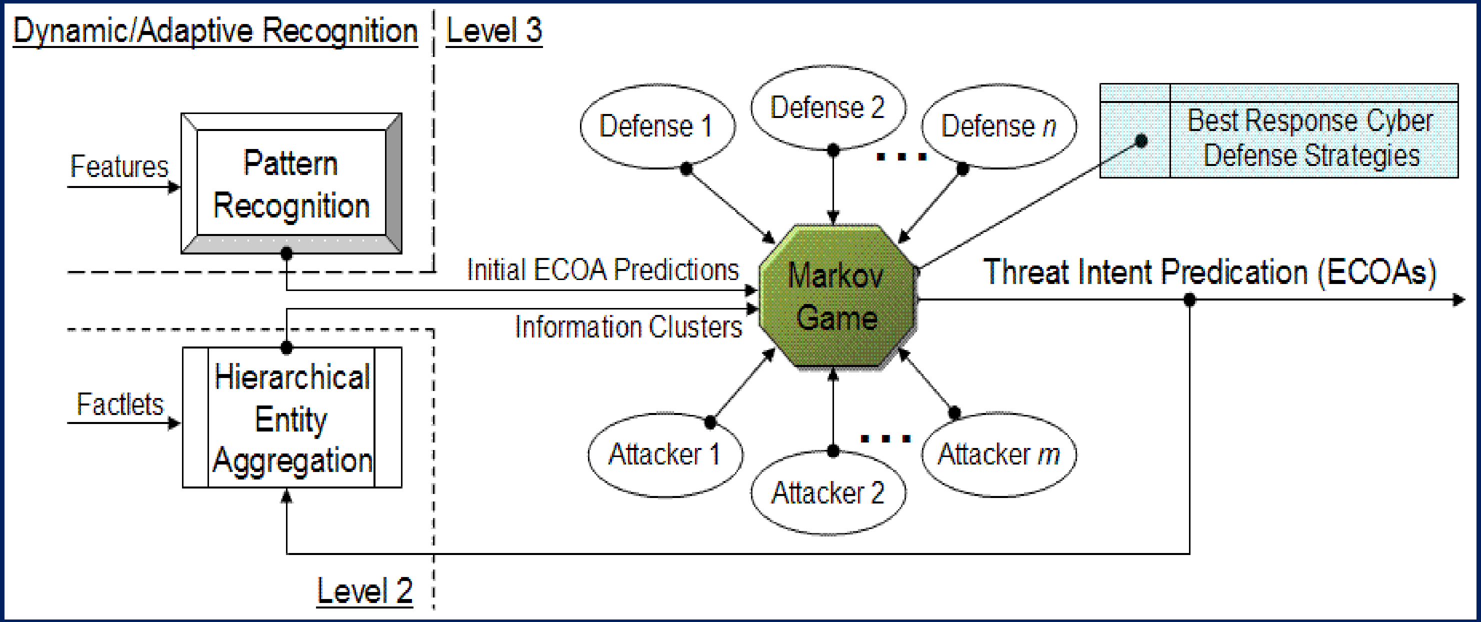
Key Features

- Recognition/Refinement/Learning Structure --- Data mining
- Decentralized multiplayer non-zero sum Markov Game
  - to estimate the belief of each possible Enemy Course of Action (ECOA).
  - white objects are modeled as the *third player* .
- A Hierarchical Entity Aggregation
  - Lower level entity aggregation
  - High level entity aggregation

A Decentralized Markov Game Model

- Previous matrix game models lack sophistication to study multi-players with relatively large actions spaces, and large planning horizons.
- Our approach has several features:
  - *Decentralized*. Each cluster of intrusion detection systems (IDSs) makes decisions based on local information. Allow autonomies in each group for more flexibility;
  - *A Markov Decision Process* (MDP) can effectively model uncertainties in the cyber network environment;
  - *A Game framework* is an effective model to capture the nature of network conflicts;
  - *White (neutral) objects* (normal network user nodes) are modeled as one of the sets of multi-players, in addition to the traditional adversarial sets of players.

A Decentralized Markov Game Model



A Markov game is specified by

- (i) a finite set of players
- (ii) a set of states
- (iii) information structure (noisy measurements)
- (iv) for every player, a finite set of available actions
- (v) a transition rule
- (vi) a payoff function for each player

Players (decision makers)

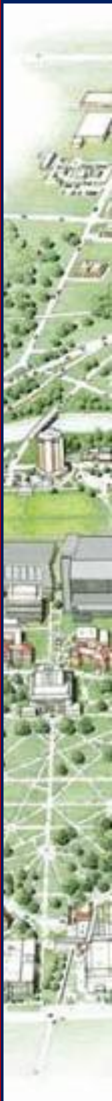
- Cyber hackers (attackers), network defense system, and normal network users are players of this Markov game model.
- We denote cyber attackers as the red team, network defense system (IDSs, Firewalls, Email-Filters, Encryption) as the blue team, normal network users as the white team.
- Cooperation within the same team is modeled by a lower level cooperative game among team members.

State Space

- □ The defense status for each network node is in the state space.
- □ For each network node (server or workstation), the state at time *k*+1 is

$$s^i(k+1) = f(s^i(k), a)$$

where *f* is the transition rule of the i^{th} network node, $s^i(k)$ is the state at time *k*, and *a* is the set of actions or control strategies by the three sets of players. The time k may be a discrete-event time.



Action Space

- The action control of the *i*th white player at time *k* is $u_w^i(k) = (t, v)^T$

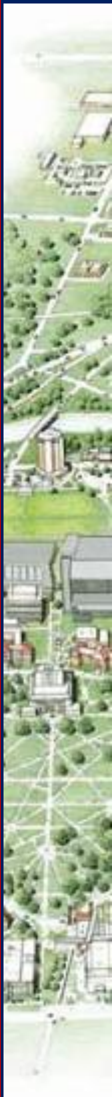
  where vector *t* is the network node providing services and *v* is the service types requested.
- For the red team, possible types of network-based attacks are: Buffer overflow, Semantic URL attack, E-mail Bombing, E-mail spam and Distributed Denial-of-Service (DDoS), penetration of military computer systems.
- For the blue team, possible defense actions are: IDS deployment, Firewall configuration , Email-filtering, and Shut down or reset servers

State Transition Example

- For example, if the state of node 1 at time *k* is [“normal”, “NULL”, “NULL”], one component of red action is “email-bombing node 1”, one component of blue action is “email-filter –configuration-no-block for node 1”, and all white actions are not related to node 1, then the probability distribution of all possible next states of node 1 is:
  - [“normal”, “email-filter-configuration”, “email-bombing”] with probability *0.4*
  - [“slow”, “email-filter-configuration”, “email-bombing”] with probability *0.3*
  - [“crashed”, “email-filter-configuration”, “email-bombing”] with probability *0.3*.

Payoff Functions

- In our decentralized Markov game model, there are two levels of payoff functions for each team (red, blue, or white):
  - lower (cooperative within each team) level
  - higher (non-cooperative between teams) level payoff functions
  - This hierarchical structure is important to model the coordinated cyber network attacks and specify optimal coordinated network defense strategies and IDS deployment.



Payoff Functions, continued

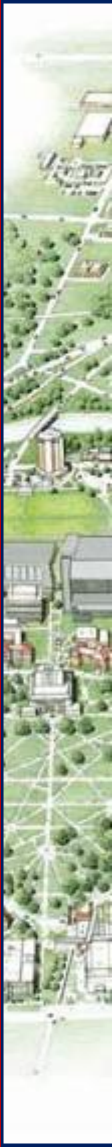
- The lower level payoff functions are used by each blue, red or white team to determine the cooperative action for each team member based on the available local information.
- The top level payoff functions at time *k* are used to evaluate the overall performance of each team.
- The lower lever payoffs are calculated distributedly by each team member and sent to the network administrator.

Strategies

- In game theory, the Nash equilibrium is an optimal collective strategy in a game where no player has anything to gain by unilaterally changing his or her own strategy.
- A mixed strategy is used in game theory to describe a strategy comprised of possible actions with associated probabilities.
- In our cyber network security application, *mixed Nash strategies* are preferred since
  - existence is guaranteed
  - the stochastic nature of mixed Nash strategy is compatible with the Markov game model
  - A mixed strategy can keep opponents off balance.

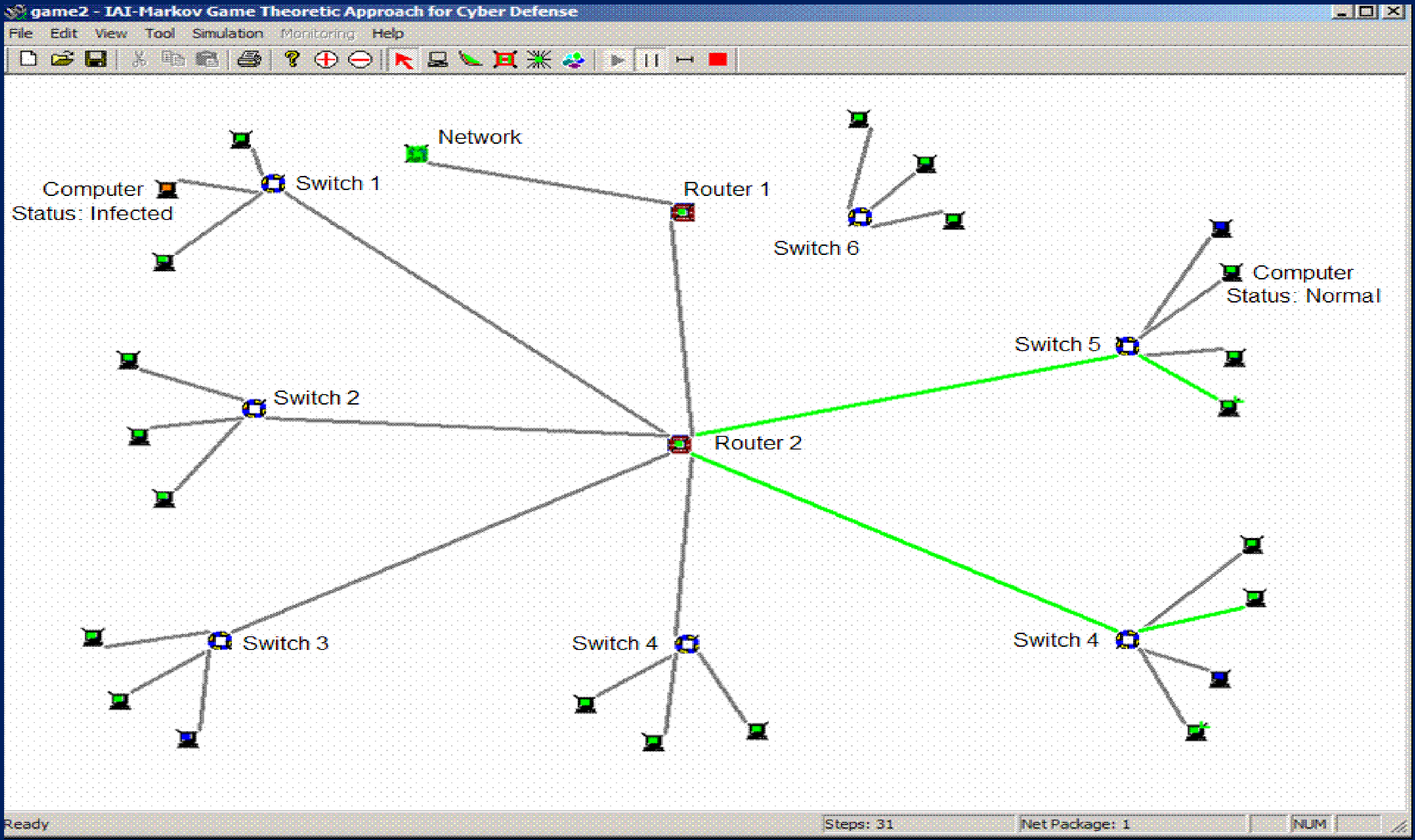
Cyber Game Simulation & Experiments

- Network components: Computer(host), Switch, Open Shortest Path First (OSPF) Router or Firewall, Link (connection), and (Sub) Network (Simulated by a node).
- Traffic volume on a link (in KBps and in Mbps).
  - Light Gray: less than 1 percent of bandwidth
  - Green: more than 1 percent of bandwidth
  - Yellow: between green and red
  - Red: more than 30 percent of bandwidth
- Host status.
  - Red: Infected node.
  - Green: Vulnerable node but not infected
  - Gray: Non-vulnerable node



Simulations and Experiments

- □ Simulation software - Cyber Game Simulation Platform (CGSP)

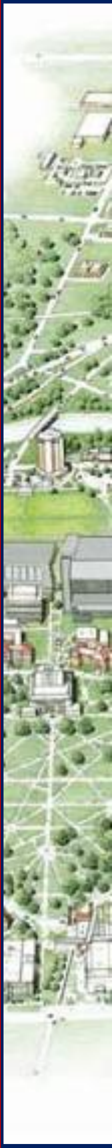


game2 - IAI-Markov Game Theoretic Approach for Cyber Defense

Ready

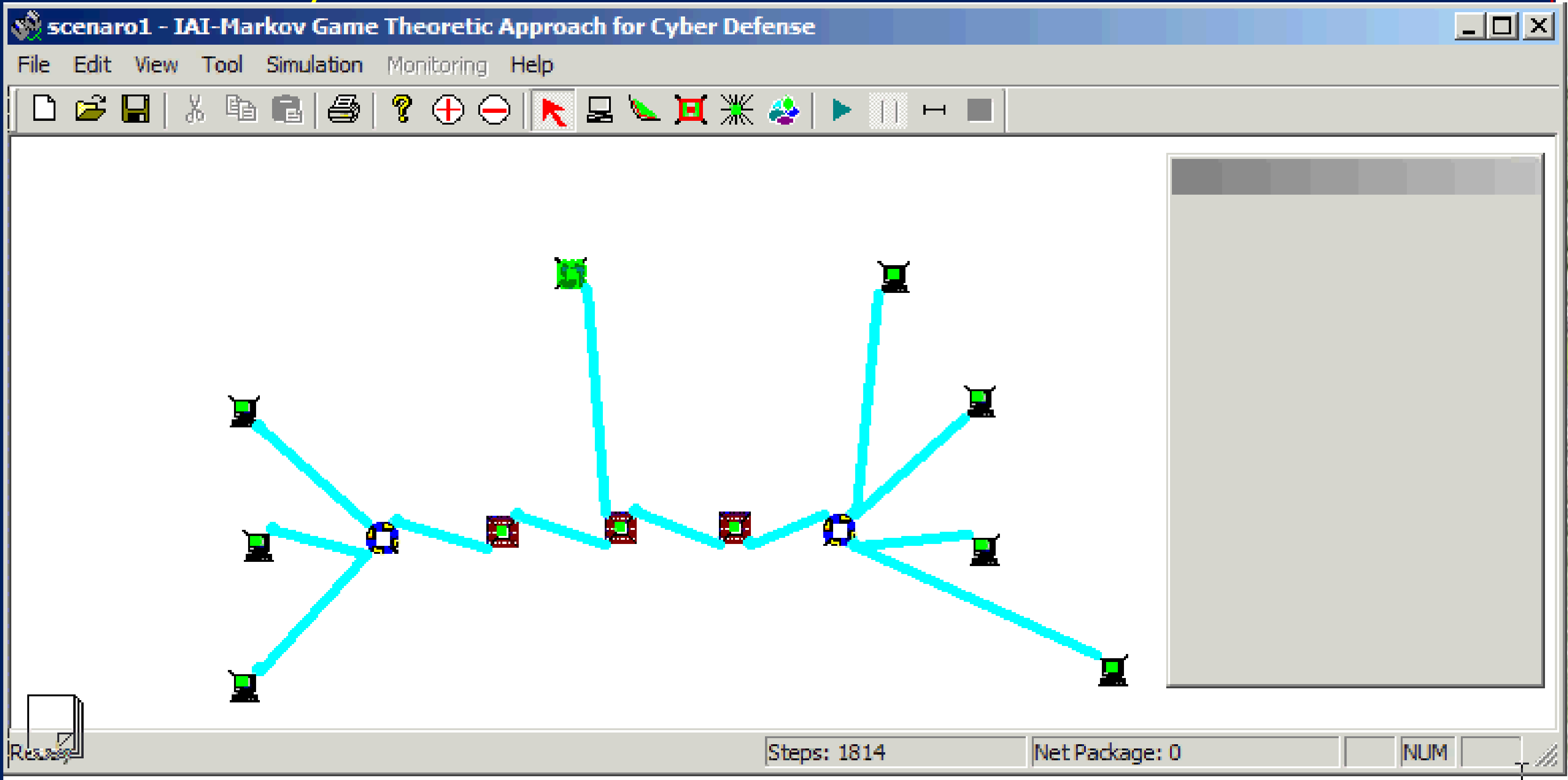
Steps: 31

Net Package: 1



Scenario 1 – “reset” enabled

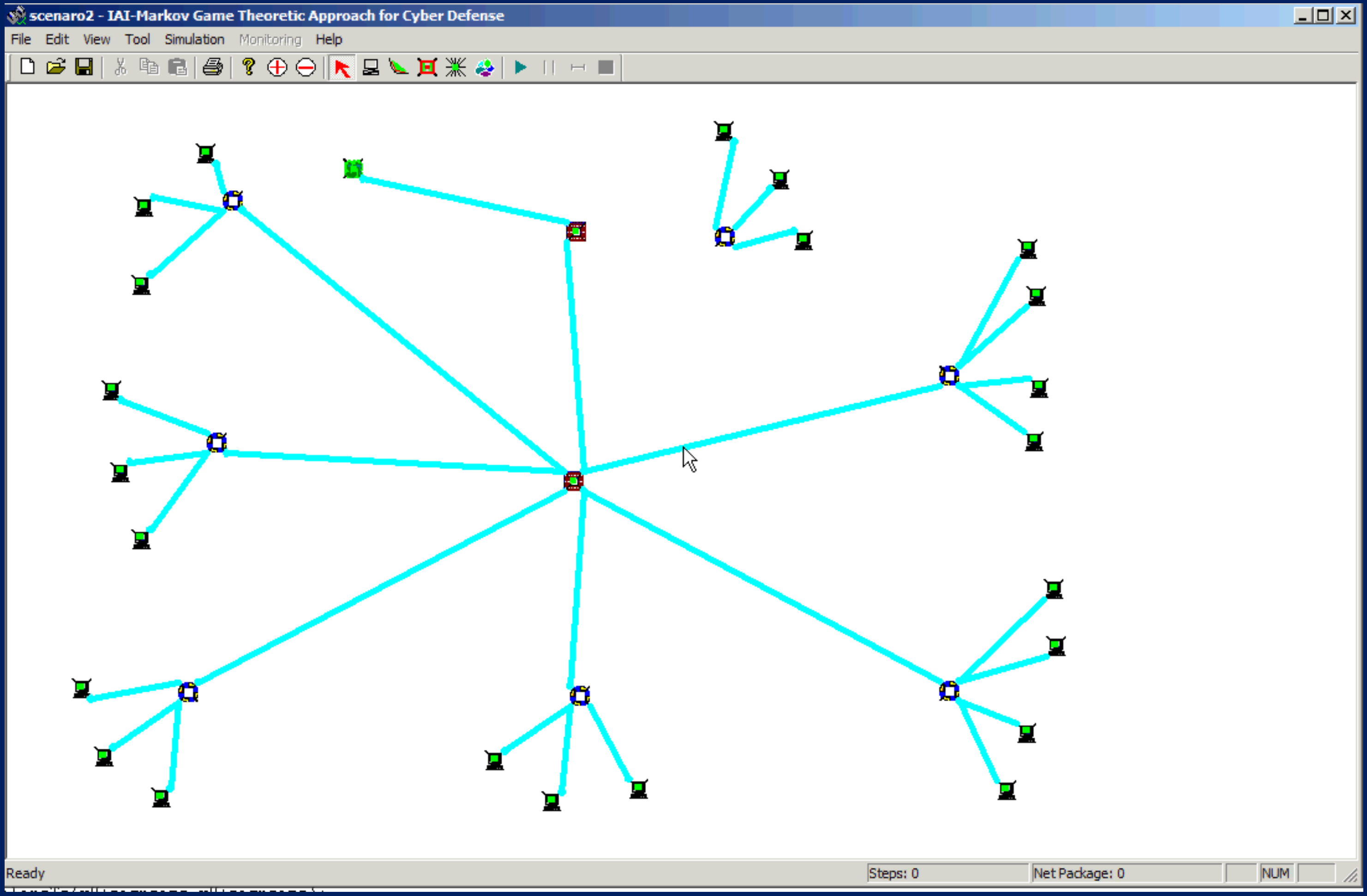
- There are 7 computers, 3 routers, 2 switches, and 1 normal outside network.



Since the network defense side can reset the computers anytime, we can see from the simulation that no servers or target computers are infected or hacked.

Scenario 2– “reset” disabled

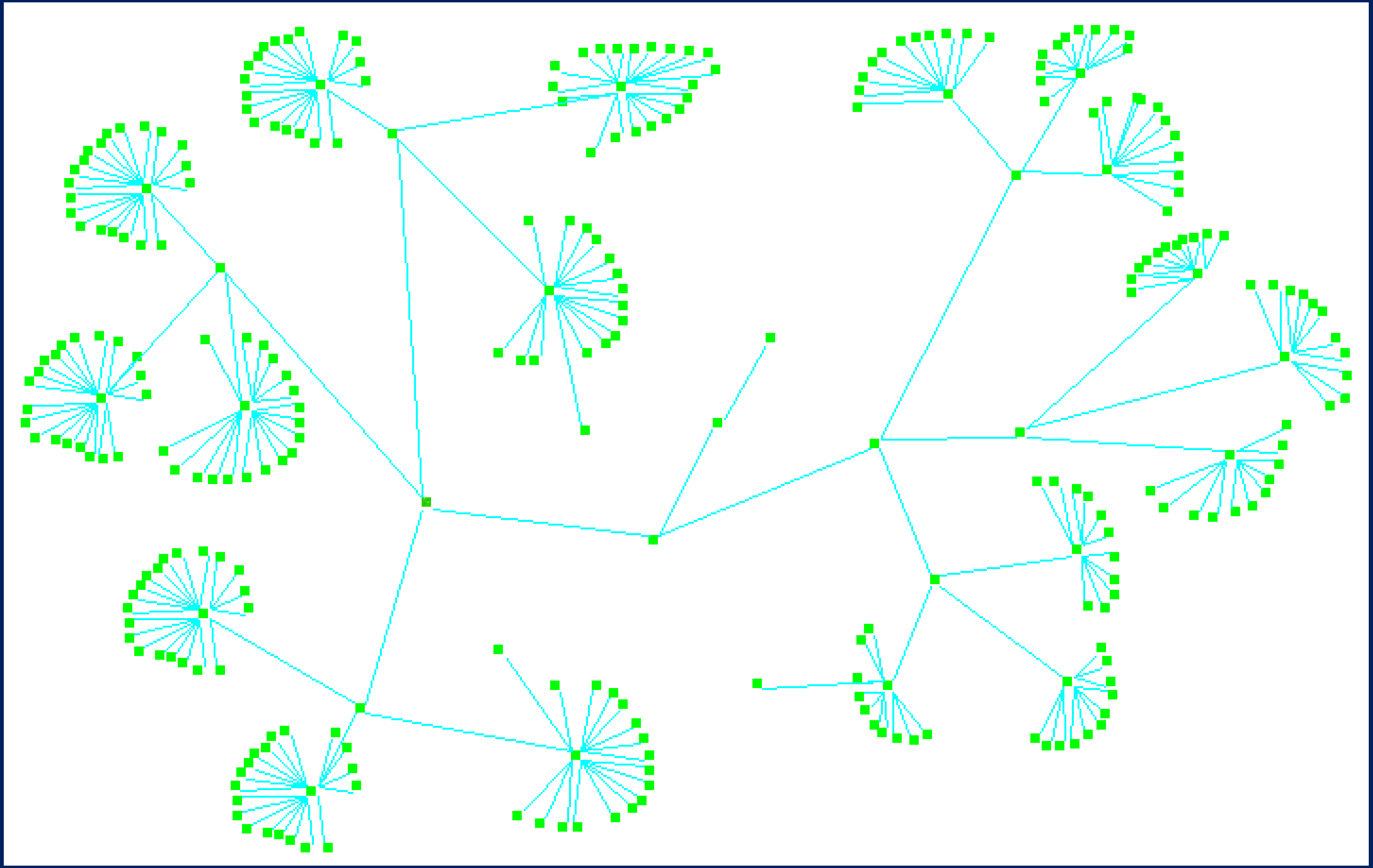
- There are 23 computers, 2 routers, 7



A target computer (web server) is infected or hacked. Then the computer (web server) will be used by attacking force to infect other more important target computers such as file servers or email servers.

Scenario 3– scalability test

- There are 269 computers, 10 routers, and 18 switches.



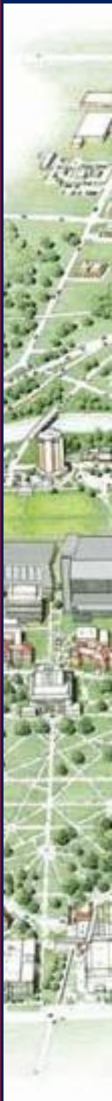
The simulation is slower than the previous two scenarios due to the increased computing work. Fortunately, the intelligent interactions between two sides are well simulated and demonstrated based on our Markov game model.

Concluding Remarks

- Many application areas of control of complex dynamic systems involve groups of controllers called teams. Each group consists of a principal controller and multiple agents.
- Some teams may be cooperative with other teams. Some may be adversarial.
- The natural framework for this field is dynamic game theory.
- The calculations and implementations of strategies are computationally intensive.

Concluding Remarks, continued

- Some recent theoretical results provide guidance in the applications
- Some application areas use ad hoc methods in the absence of additional theory.
- Research challenges remain in
  - The underlying theory
  - Developing scalable algorithms and software

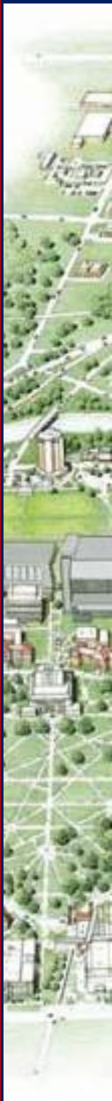


References

1. J.V. Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*, Princeton University Press, 1944.
2. R. Isaacs, *Differential Games*, John Wiley & Sons, Inc., New York, 1965.
3. T. Basar and G.J. Olsder, *Dynamic Noncooperative Game Theory, 2nd Edition (revised)*, the Society for Industrial and Applied Mathematics, 1998.
4. Y.C. Ho, “Differential Games, Dynamic Optimization, and Generalized Control Theory,” *Journal of Optimization Theory and Applications*, Vol. 6, No. 3, 1970.
5. M.H. Breitner, “The Genesis of Differential Games in Light of Isaacs’ Contributions,” *Journal of Optimization Theory and Applications,* Vol. 124, No. 3, pp. 523–559, March 2005.
6. V. Turetsky and J. Shinar, “Missile Guidance Laws Based on Pursuit-Evasion Game Formulations,” *Automatica*. 39, No. 4, pp. 607 – 618, 2003.
7. M.H. Foley and W.E. Schmitendorf, “A Class of Differential Games with Two Pursuers Versus One Evader,” *IEEE Transactions on Automatic Control*, Vol. AC-19, No. 3, 1974.

References, continued

8. A. G. Pashkov and S.D. Terekhov, “A Differential Game Approach with Two Pursuers and One Evader,” *Journal of Optimization Theory and Applications*, Vol. 55, No. 2, 1987.
9. P. Hagedorn and J.V. Breakwell, “A Differential Game with Two Pursuers and One Evader,” *Journal of Optimization Theory and Applications*, Vol. 18, No. 1, 1976.
10. M. Simaan and J. B. Cruz Jr., “On the Stackelberg strategy in nonzero-sum games,” *Journal of Optimization Theory and Applications*, V. 11, 533 - 555, No. 5, 1973.
11. D. Li, J.B. Cruz, G. Chen, C. Kwan and M.H Chang, “A Hierarchical Approach To Multi-Player Pursuit-Evasion Differential Games”, *Proceedings of Joint IEEE Conference on CDC-ECC05*, Spain, 2005.
12. D.P. Bertsekas, J.N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, Belmont, MA, 1996.
13. I.M. Mitchell, A.M. Bayen and C.J. Tomlin, “A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games,” *IEEE Trans. on Automatic Control*, V.50, 7, July 2005, pp.947 – 957.



References, continued

14. J. Hespanha, M. Prandini, and S. Sastry, “Probabilistic pursuit-evasion games: A One-step Nash Approach,” In *Proc. Of 39th IEEE Conf. on Decision and Control*, pages 2272-2277, 2000.
15. A. Antoniades, H.J.Kim and S. Sastry, Pursuit-evasion strategies for teams of multiple agents with incomplete information, in *Proceedings of 42nd IEEE Conference on Decision and Control*, 2003.
16. Mo Wei, Genshe Chen, Jose B. Cruz, Jr., Leonard Haynes, Khanh Pham, and Erik Blasch, “Multi-Pursuer Multi-Evader Pursuit-Evasion Games with Jamming Confrontation,” *Journal of Aerospace Computing, Information, and Communication*, Vol. 4, Issue No. 3, pp. 676-692, April 2007.
17. Dongxu Li and Jose B. Cruz, Jr., “Cooperative Pursuit of Multiple Evaders: On the Structure of Look-ahead Improvement and Decentralized Implementation,” *Proceedings 45th IEEE Conference on Decision and Control, San Diego, CA, December 2006.*
18. Mo Wei, Genshe Chen, Jose B. Cruz, Jr., Leonard S. Haynes, Mou-Hsiung Chang, Erik Blasch, “A Decentralized Approach to Pursuer-Evader Games with Multiple Superior Evaders in Noisy Environment,” *2007 IEEE Aerospace Conference,* March 3-10, 2007, Big Sky, MT.

References, continued

19. M. Wei, G. Chen, J. B. Cruz, L. Haynes, M. Kruger, and E. Blasch “Game theoretic modeling and control of military air operations with retaliatory civilians,” *2007 IEEE Aerospace Conf*, March 3-10, 2007, Big Sky, MT.
20. Mo Wei, Genshe Chen, J. B. Cruz, L. Haynes, M. Kruger, and E. Blasch, “Game theoretic behavior features change prediction in hostile environments,” *SPIE’s Defense and Security Symposium*, Orlando, FL, 9-13 April 2007.
21. D. Shen, G. Chen, J. B. Cruz, L. Haynes, M. Kruger, and E. Blasch, “A Markov game theoretic approach for cyber situational awareness”, *SPIE’s Defense and Security Symposium*, Orlando, FL, 9-13 April 2007.
22. Dongxu Li, Jose B. Cruz, Jr., Genshe Chen, Chiman Kwan, and Mou-Hsiung Chang, “A Hierarchical Approach to Multi-Player Pursuit-Evasion Differential Games,” *44th IEEE Conference on Decision and Control,* Seville, Spain, December 2005, CD ROM.
23. Dongxu Li and Jose B. Cruz, Jr., “Better Cooperative Control with Limited Look Ahead,” *Proceedings, 2006 American Control Conference*, Minneapolis, MN, June 14-16, 2006, CD ROM.