SAND2012-8680P

Stochastic Stackelberg Games, with Applications to Adversarial Patrolling

Yevgeniy Vorobeychik*

Sandia National Laboratories, CA

(with Bo An, Milind Tambe, USC, Santider Singh, UMich)

* Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

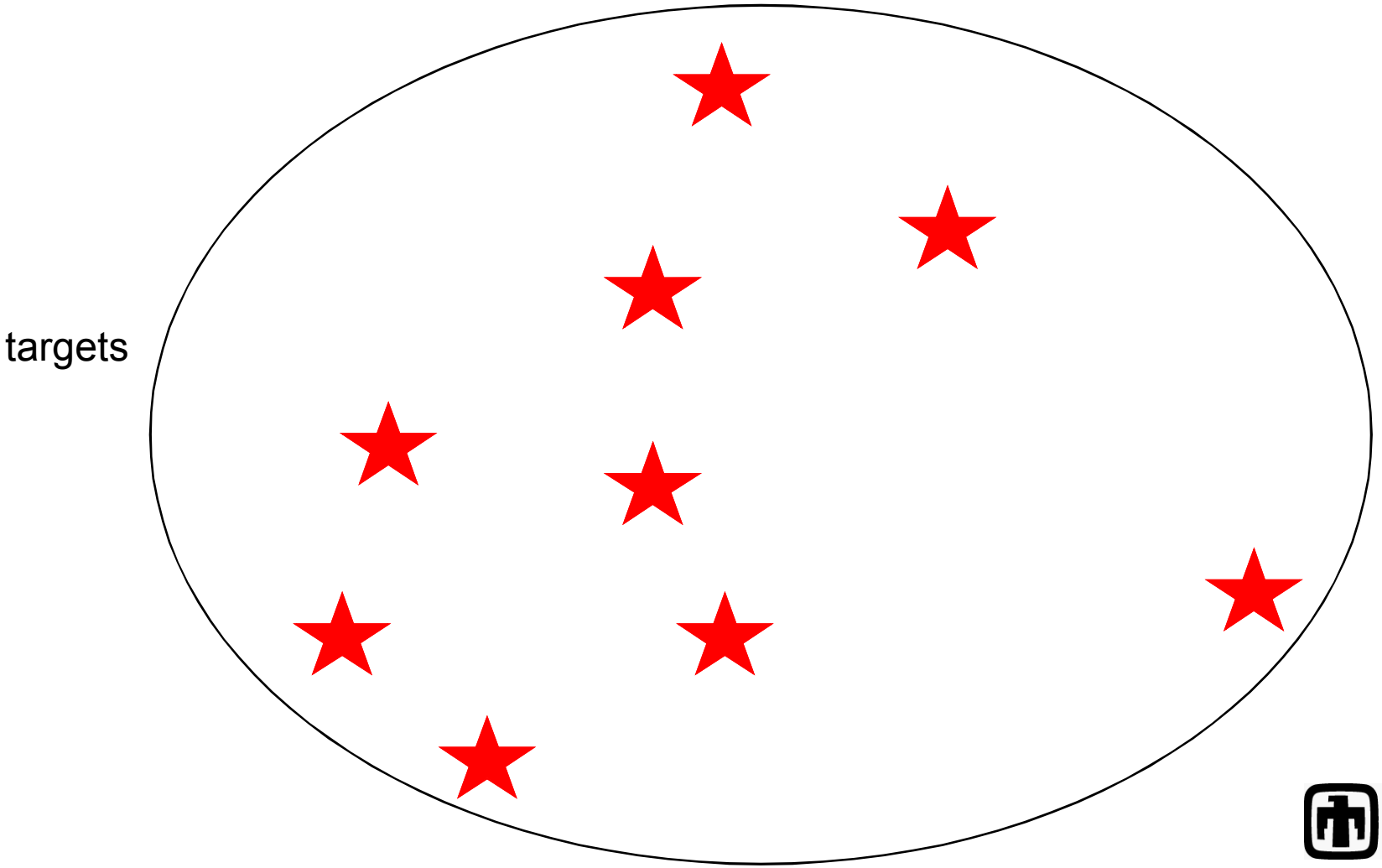
Sandia National Laboratories

Stackelberg Equilibria and Security Games

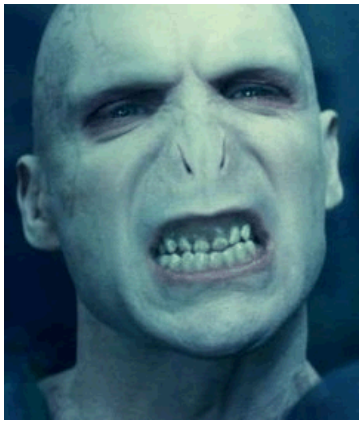
Sandia National Laboratories

Game Theoretic Model of Security

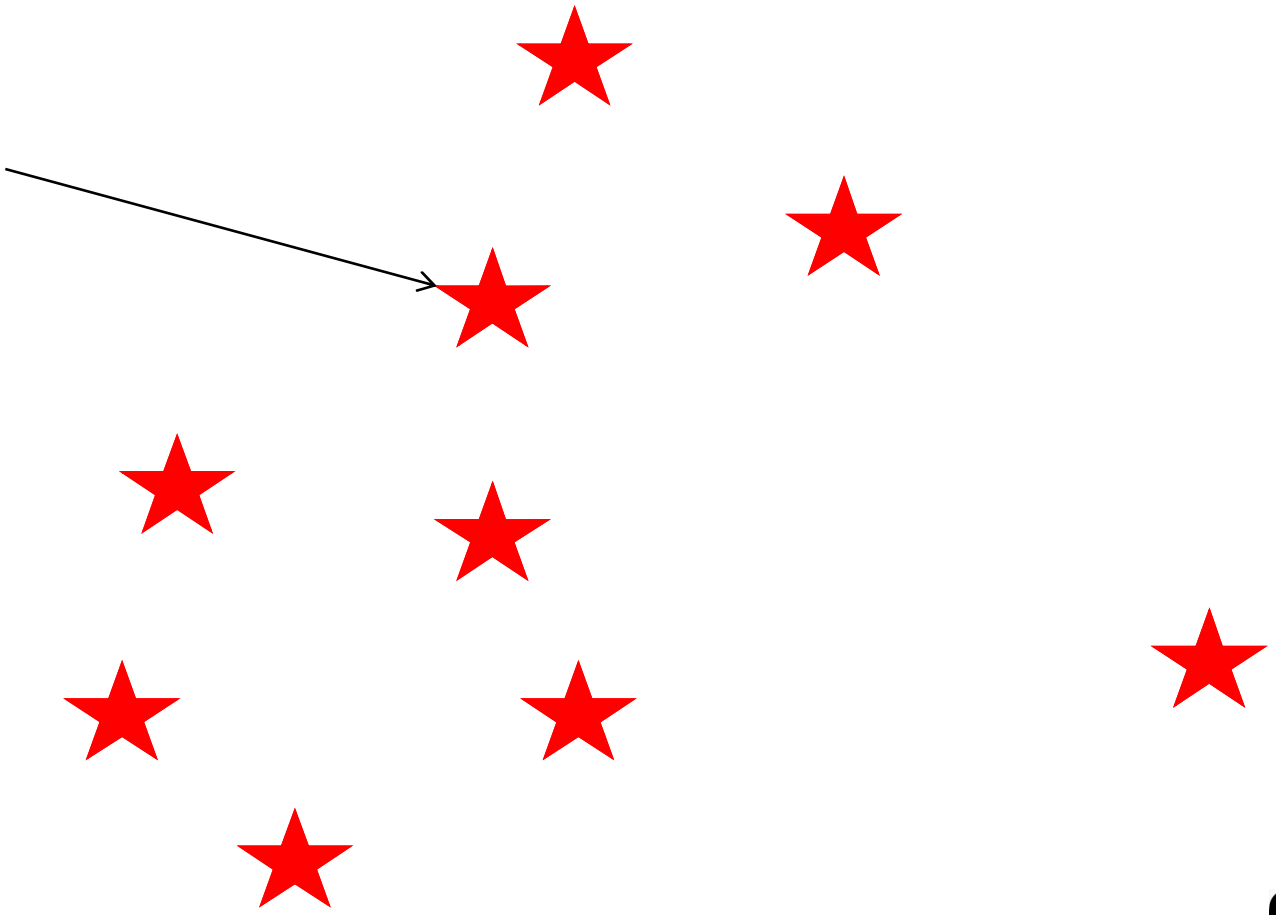
Game Theoretic Model of Security



Game Theoretic Model of Security

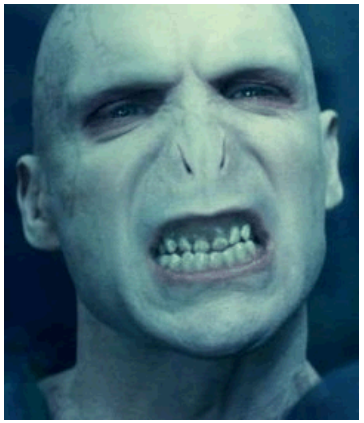


attacker



Sandia National Laboratories

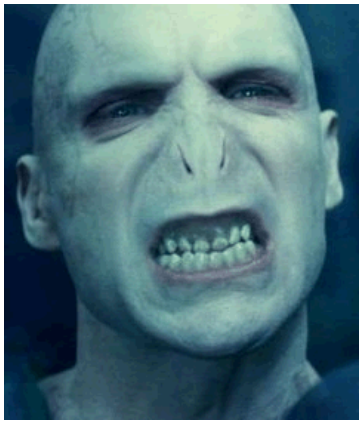
Game Theoretic Model of Security



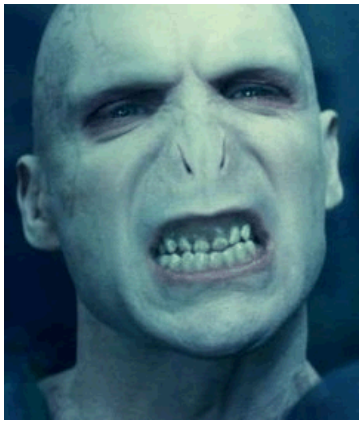
defender

protects a randomly chosen target

Game Theoretic Model of Security

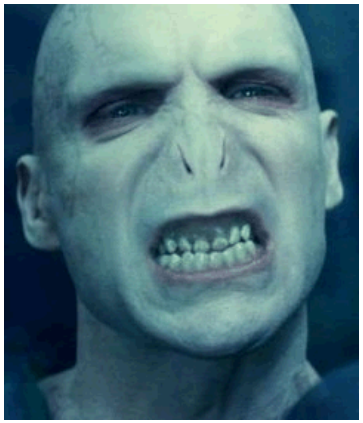


Game Theoretic Model of Security



Sandia National Laboratories

Game Theoretic Model of Security

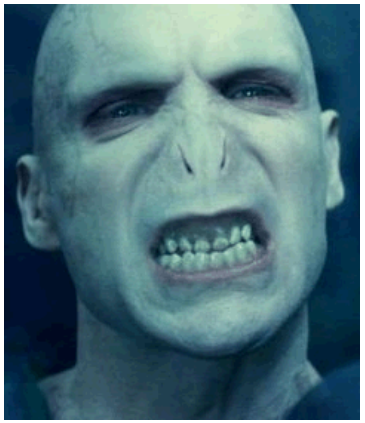


knows the probability each target is protected

Game Theoretic Model of Security

Sandia National Laboratories

Game Theoretic Model of Security



Security Games and Stackelberg Equilibria

- A security game is:
  - T: a set of targets
  - R_D/R_A: defender/attacker values for targets
  - Defender: chooses a strategy p in which each target i has the probability p_i of being covered
- Attacker: knows p; chooses a target to attack which maximizes expected utility $R_{A,i}\ (1 - p_i)$
- Stackelberg equilibrium: defender chooses *p that maximizes its utility*, ***accounting for attacker's best response to p***

MILP and Stackelberg Equilibria

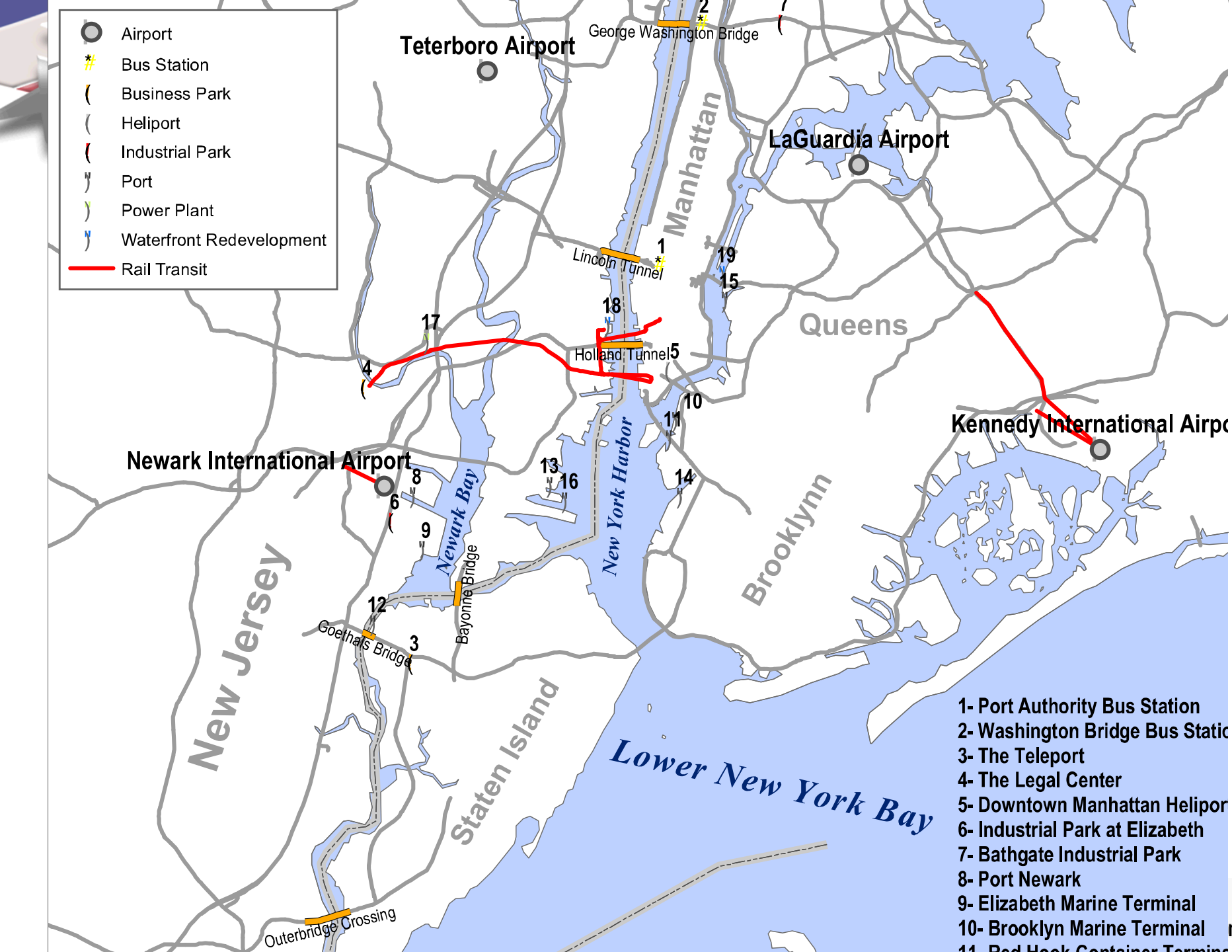
- Much previous work has focused on fast linear / integer programming techniques/formulations for such problems
- Deployed in real applications:
  - LAX canine patrol
  - federal air marshall scheduling
  - US coast guard

Adversarial Patrolling Games

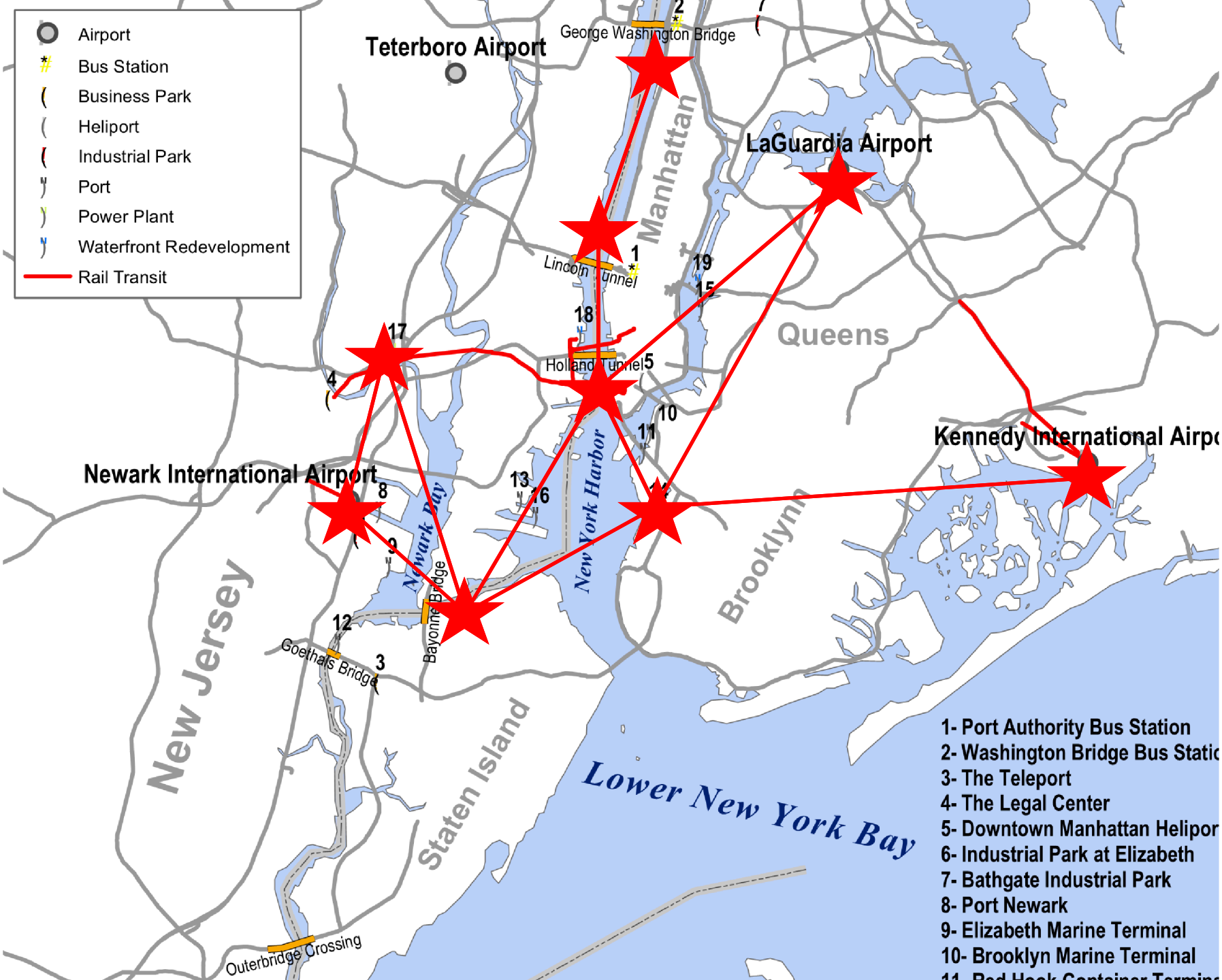
Sandia National Laboratories

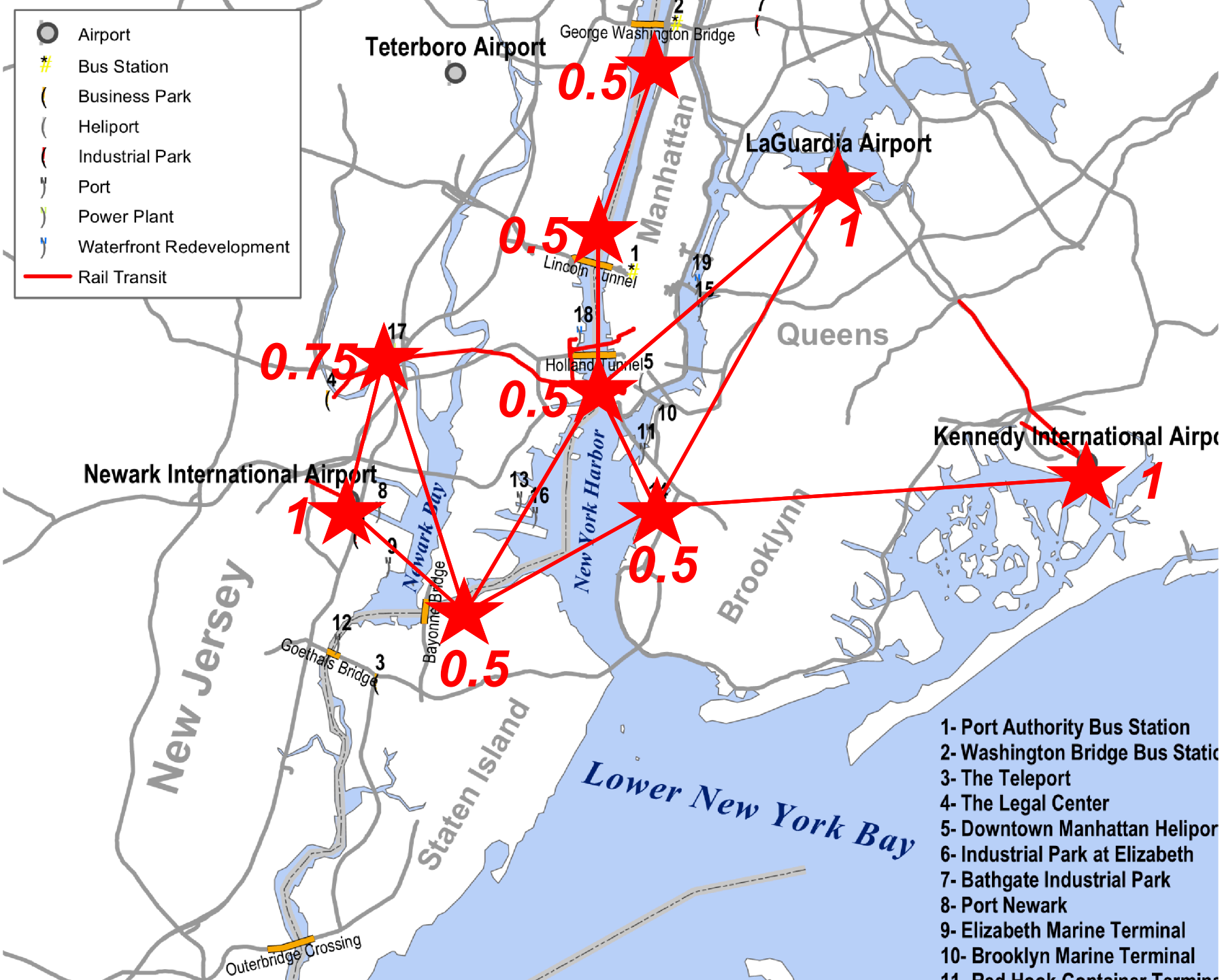
Motivation

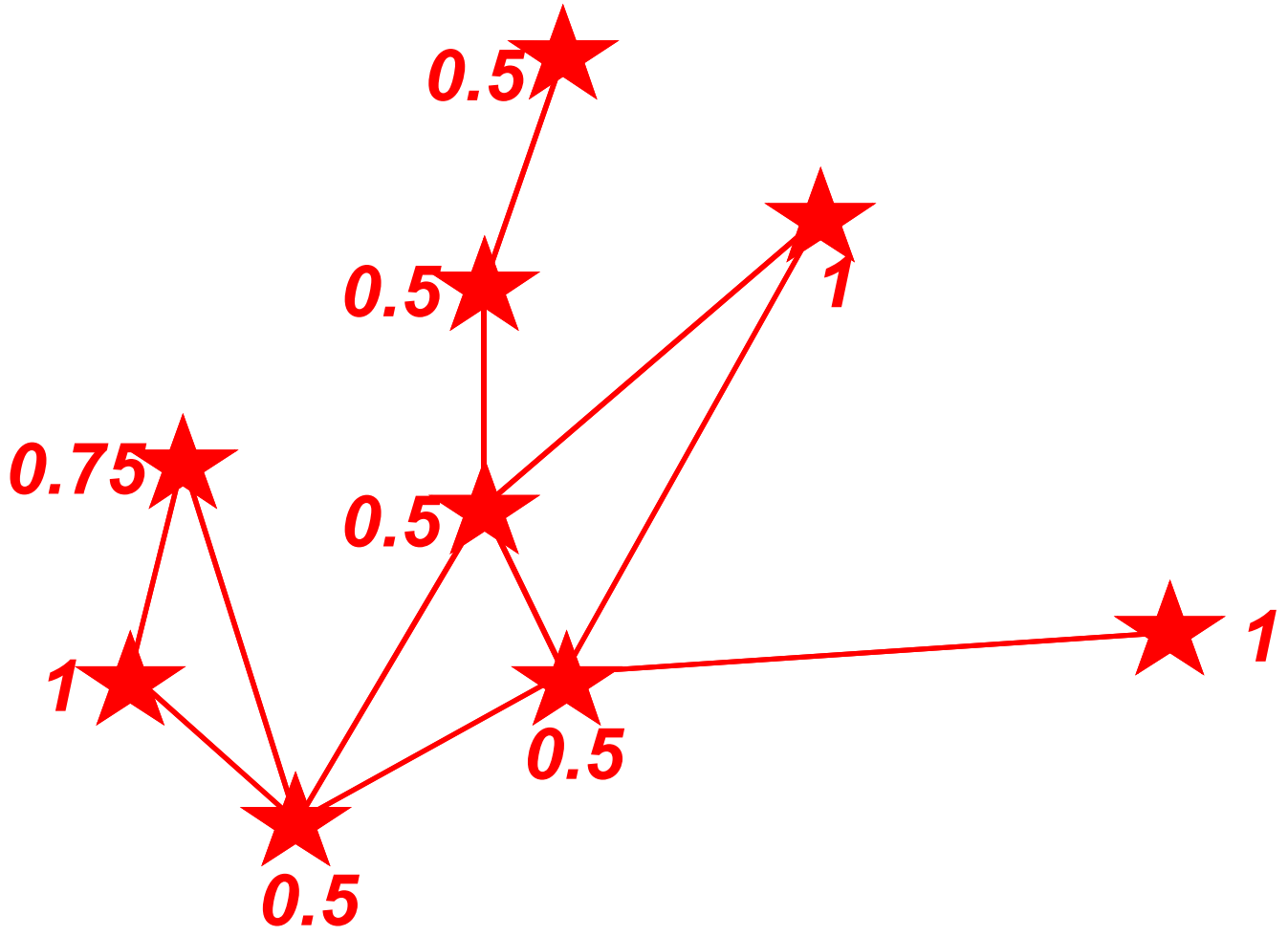
- ***Suppose the defender follows a patrolling schedule***
  - ***instead of choosing a random target to cover, defender chooses a random sequence of targets to cover***
- ***If an attacker observes defender's current location, it can reveal information about where the defender will be next***

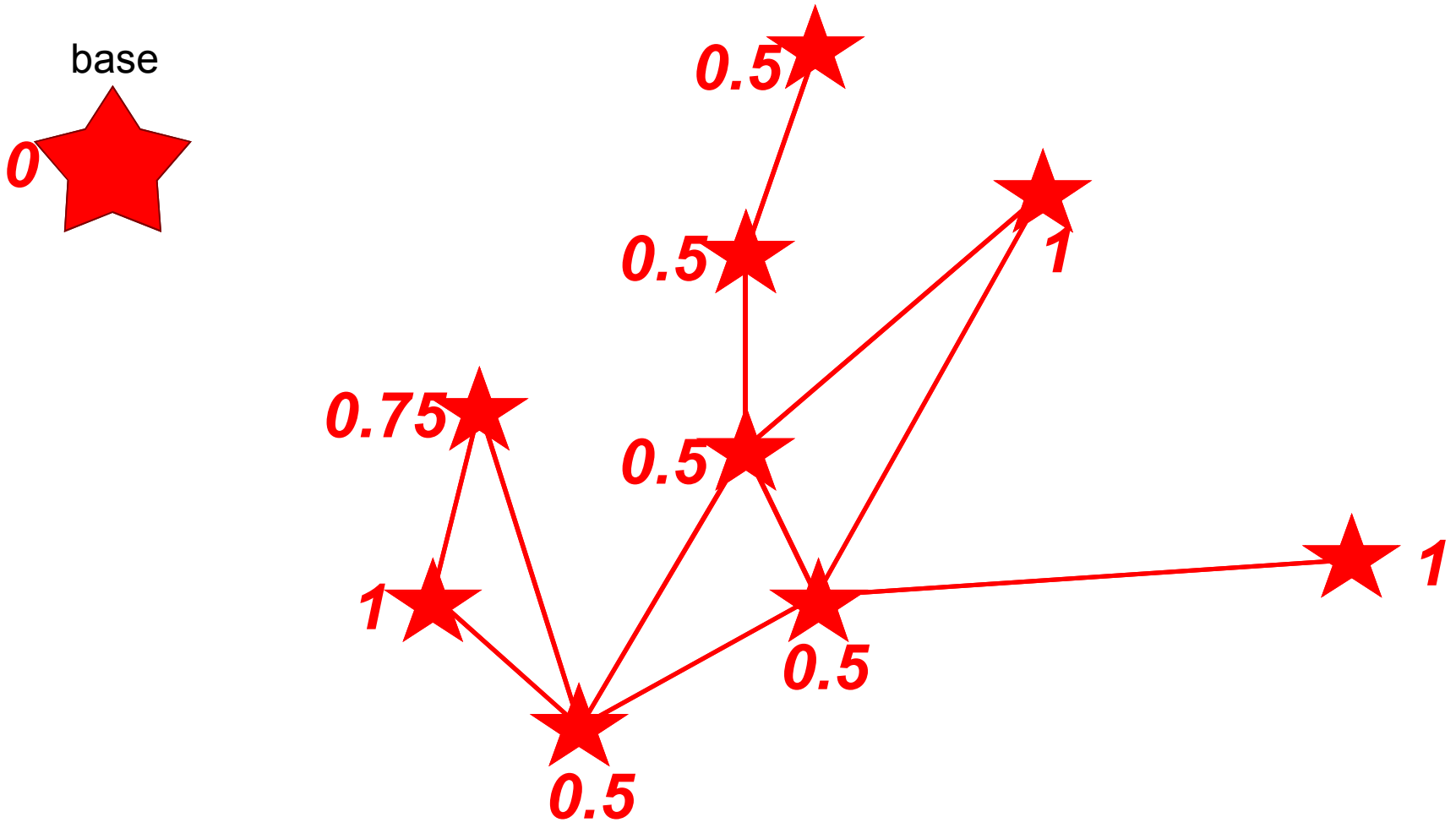

Airport
Bus Station
Business Park
Heliport
Industrial Park
Port
Power Plant
Waterfront Redevelopment
Rail Transit
Teterboro Airport
George Washington Bridge
LaGuardia Airport
Manhattan
Queens
Lincoln Tunnel
Holland Tunnel
Newark International Airport
Kennedy International Airpo
Newark Bay
New York Harbor
Brooklynn
New Jersey
Staten Island
Bayonne Bridge
Goethals Bridge
Outerbridge Crossing
Lower New York Bay
1- Port Authority Bus Station
2- Washington Bridge Bus Statio
3- The Teleport
4- The Legal Center
5- Downtown Manhattan Heliport
6- Industrial Park at Elizabeth
7- Bathgate Industrial Park
8- Port Newark
9- Elizabeth Marine Terminal
10- Brooklyn Marine Terminal

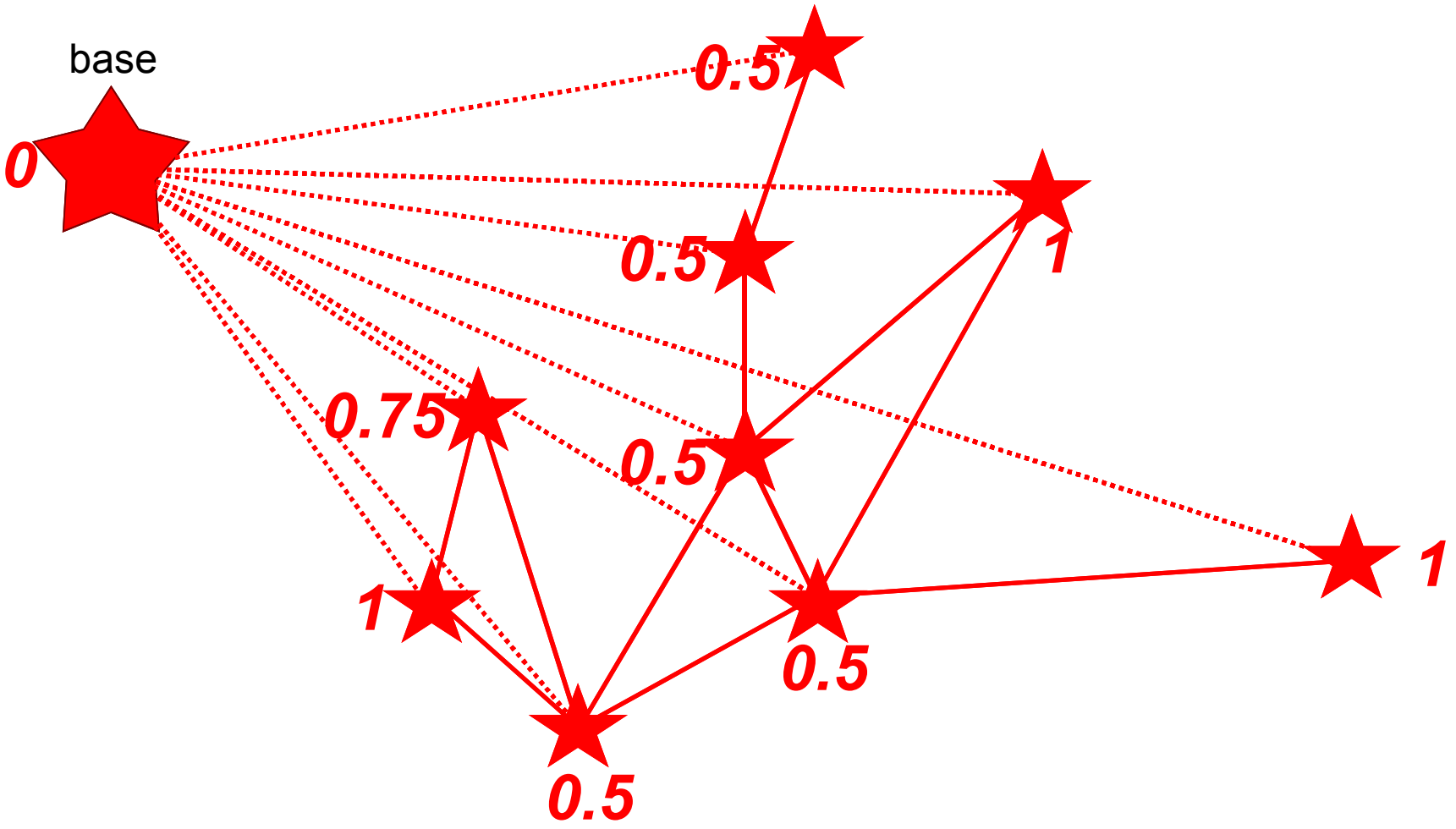

Airport
Bus Station
Business Park
Heliport
Industrial Park
Port
Power Plant
Waterfront Redevelopment
Rail Transit
Teterboro Airport
George Washington Bridge
Manhattan
LaGuardia Airport
Lincoln Tunnel
Holland Tunnel
Queens
Newark International Airport
Kennedy International Airpo
Newark Bay
New York Harbor
Brooklynn
New Jersey
Bayonne Bridge
Goethals Bridge
Staten Island
Outerbridge Crossing
Lower New York Bay
1- Port Authority Bus Station
2- Washington Bridge Bus Statio
3- The Teleport
4- The Legal Center
5- Downtown Manhattan Helipor
6- Industrial Park at Elizabeth
7- Bathgate Industrial Park
8- Port Newark
9- Elizabeth Marine Terminal
10- Brooklyn Marine Terminal

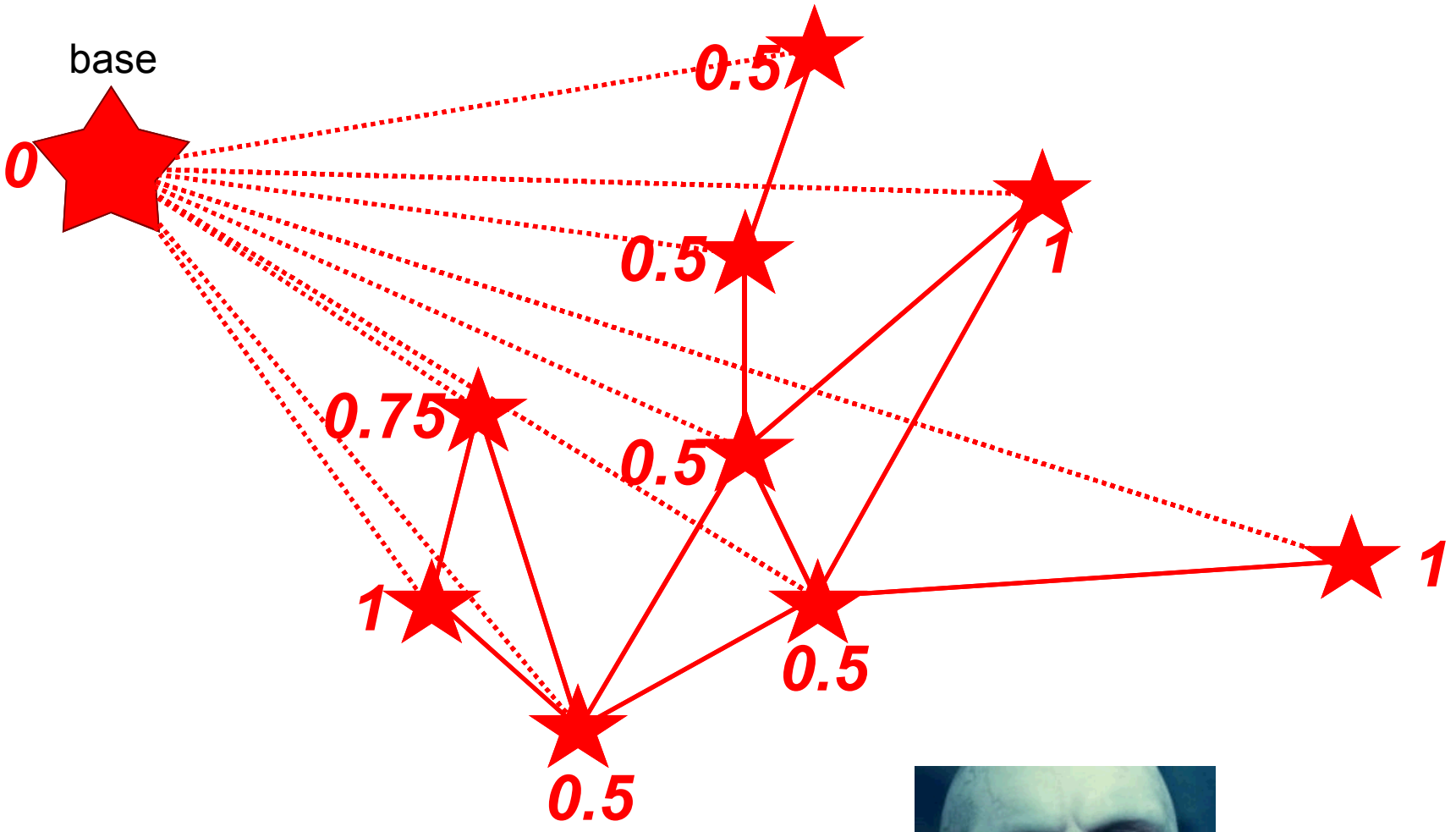

Airport
Bus Station
Business Park
Heliport
Industrial Park
Port
Power Plant
Waterfront Redevelopment
Rail Transit
Teterboro Airport
George Washington Bridge
Manhattan
LaGuardia Airport
Lincoln Tunnel
Holland Tunnel
Queens
Newark International Airport
Kennedy International Airpo
New York Harbor
Newark Bay
Bayonne Bridge
Goethals Bridge
Outerbridge Crossing
New Jersey
Staten Island
Brooklyn
Lower New York Bay
1- Port Authority Bus Station
2- Washington Bridge Bus Statio
3- The Teleport
4- The Legal Center
5- Downtown Manhattan Helipor
6- Industrial Park at Elizabeth
7- Bathgate Industrial Park
8- Port Newark
9- Elizabeth Marine Terminal
10- Brooklyn Marine Terminal

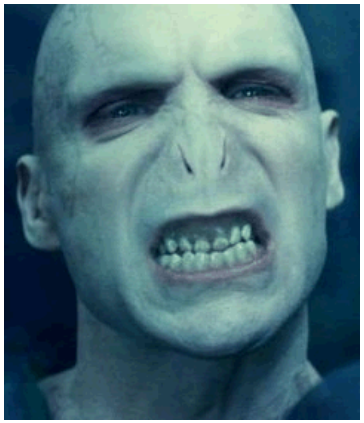

Airport
Bus Station
Business Park
Heliport
Industrial Park
Port
Power Plant
Waterfront Redevelopment
Rail Transit
Teterboro Airport
George Washington Bridge
Manhattan
LaGuardia Airport
Lincoln Tunnel
Holland Tunnel
Queens
Newark International Airport
Kennedy International Airpo
New Jersey
Newark Bay
New York Harbor
Brooklyn
Bayonne Bridge
Goethals Bridge
Staten Island
Outerbridge Crossing
Lower New York Bay
0.5
0.5
1
0.75
0.5
1
0.5
1
0.5
1- Port Authority Bus Station
2- Washington Bridge Bus Statio
3- The Teleport
4- The Legal Center
5- Downtown Manhattan Helipor
6- Industrial Park at Elizabeth
7- Bathgate Industrial Park
8- Port Newark
9- Elizabeth Marine Terminal
10- Brooklyn Marine Terminal



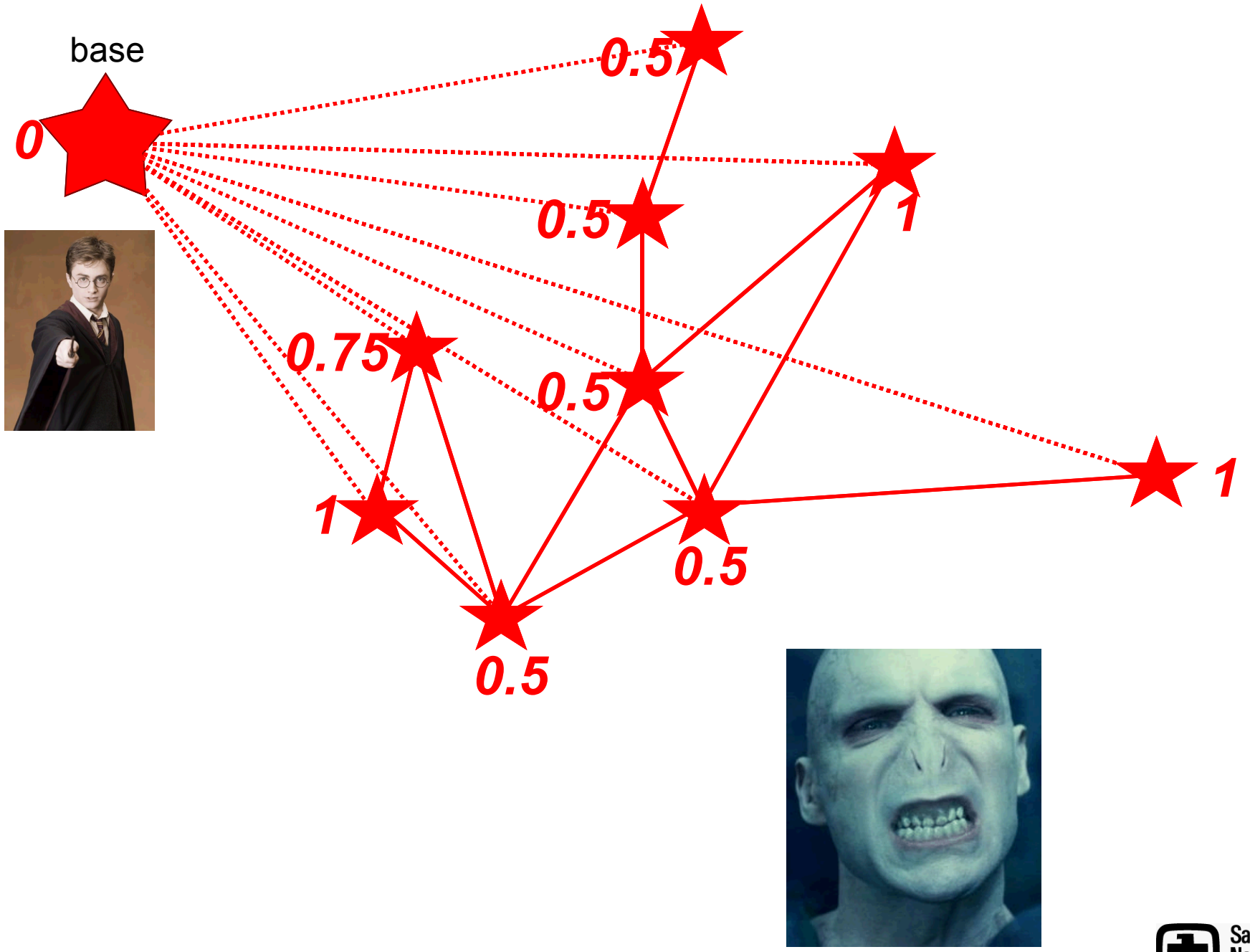




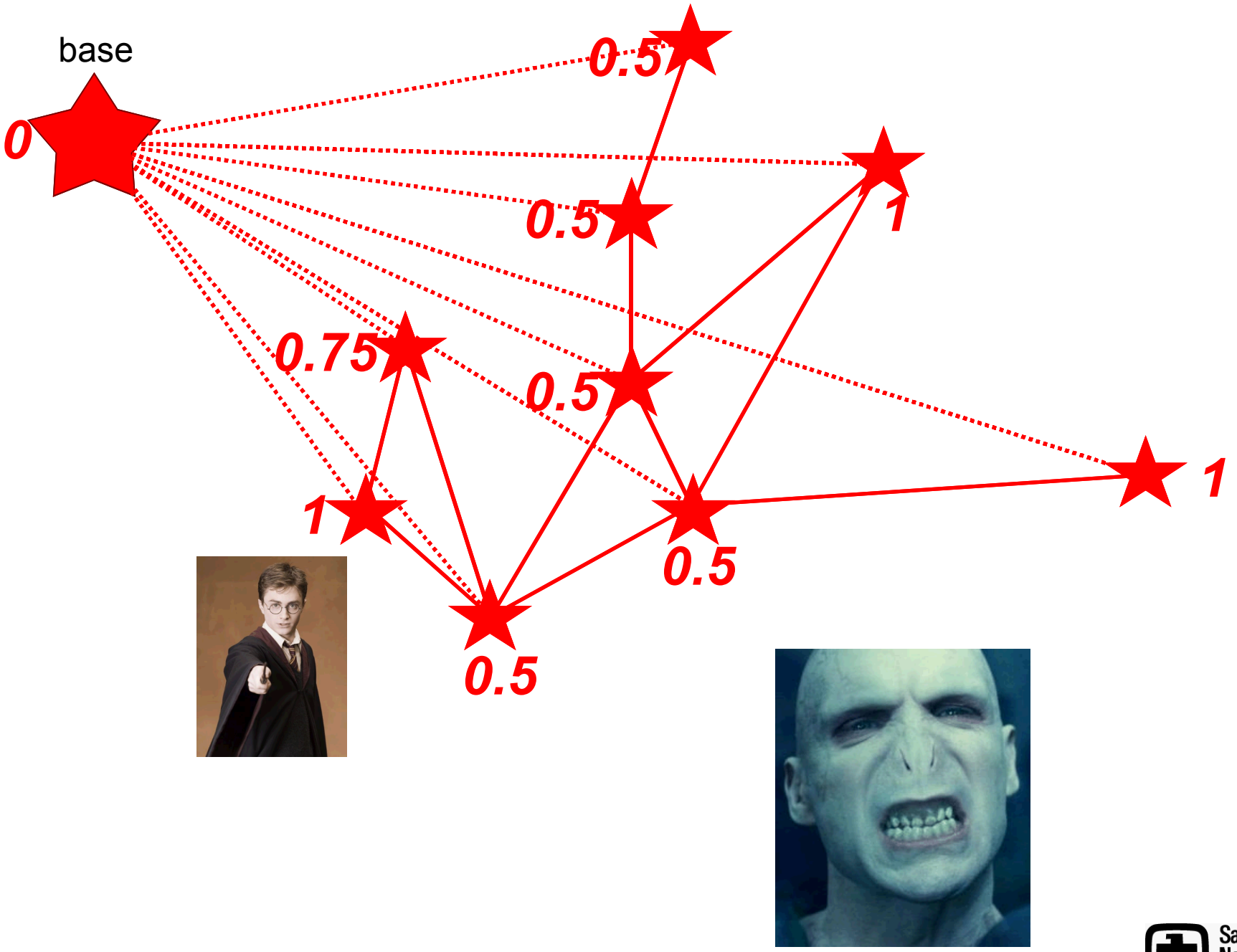


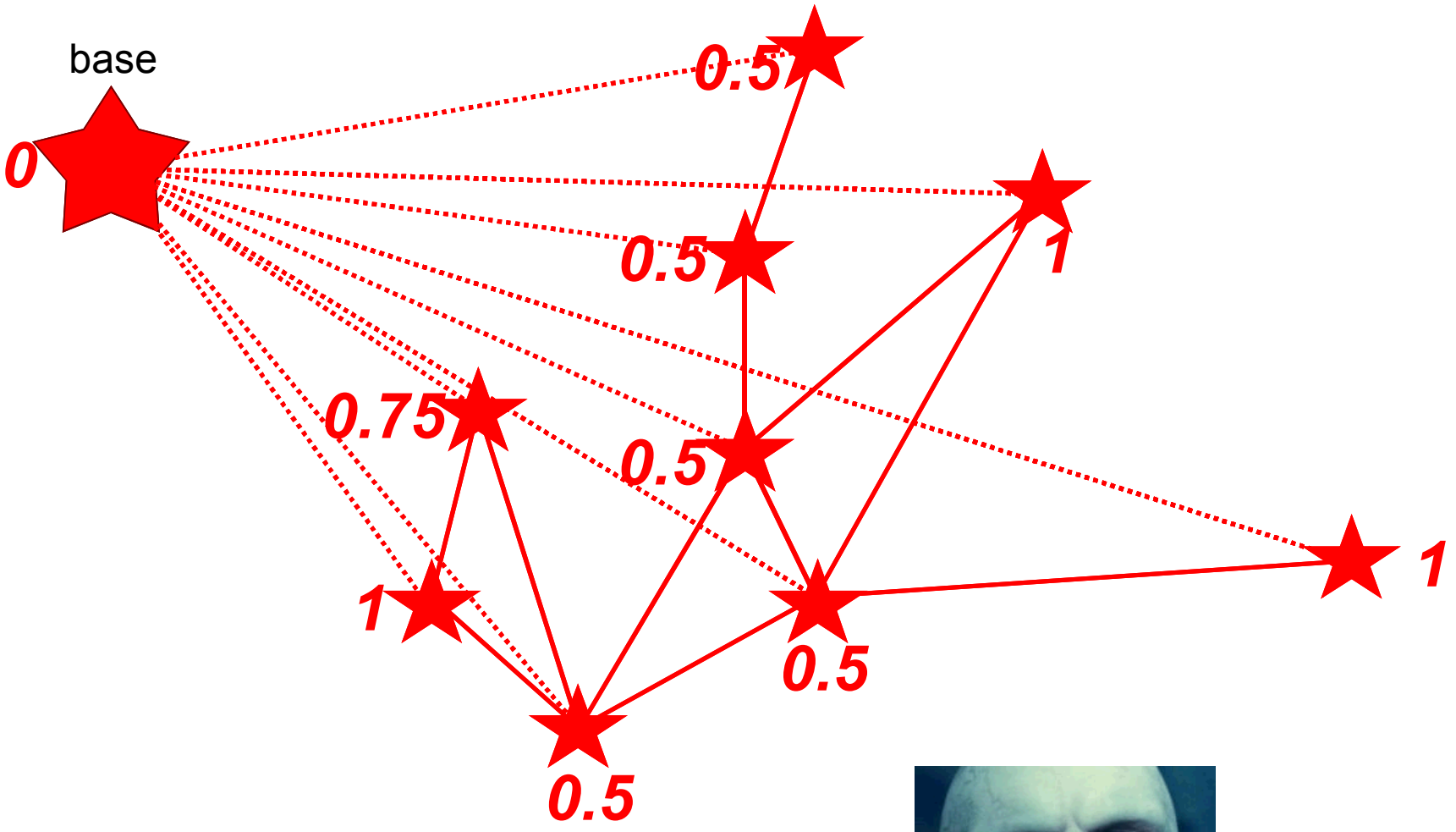


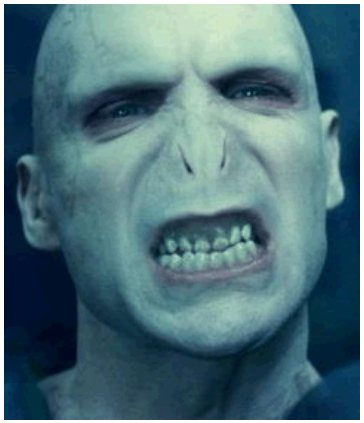
Sandia National Laboratories



Sandia National Laboratories


base
0
0.5
0.5
1
0.75
0.5
1
1
0.5
0.5





Sandia National Laboratories

APG Formally (2 players)

- APG = {T,u,δ,G}
  - T: set of targets
  - u: vector of attacker values (assume zero-sum)
  - δ: discount factor
  - G = (T,E) a graph, T = nodes, E = edges; defender can only move from i to j if (i,j) is in E (A_{ij} = 1 iff edge from i to j)
- Defender always starts at target 0
- π: defender policy (choose next target as function of history)
- Attacker observes current location i of defender, and knows π
- a: attacker policy (choose whether to wait/attack; if attack, choose which target to attack; decisions a function of observed defender position)
  - if attacker chooses to attack a target, attack happens simultaneously with the next defender move

Goal: Compute Stackelberg Equilibrium

- Stackelberg equilibrium
  - For every defender policy, there is an optimal attacker policy ("best response")
  - Goal: compute optimal defender policy, accounting for attacker's best response behavior
  - We allow defender's policies to be stochastic (can randomly move between targets)

Stepping Back:
Stackelberg Equilibria in Stochastic Games

Sandia National Laboratories

APGs and Stochastic Stackelberg Games

- *APGs can be viewed as a special case of stochastic Stackelberg games*
- Stochastic Stackelberg game (SSG), formally:
  - 2 players: leader (L; think: defender) and follower (F; think: attacker)
  - S : a set of states
  - $A = \{A_L \times A_F\}$: joint action space of players
  - $P : S \times A \rightarrow S$: transition function ($\Pr\{s' \mid s, a_l, a_f\}$)
  - $R_L/R_A : S \times A \rightarrow R$: payoff functions
  - Infinite horizon: game goes on "forever"
  - discounted: payoffs discounted by δ at each step
  - $h_t = \{s(1)a_l(1)a_f(1)\ldots s(t)a_l(t)a_f(t)\}$: history at time t (of states and decisions up till now)
  - H : set of all possible histories

Sandia National Laboratories

Policies in SSGs

- $\pi : H \rightarrow A_L$: leader's policy, given an (arbitrary) history, return an action (or, in general, a probability distribution over actions in A_L)
- Same for the follower
- If the game is infinite horizon, can't even represent these!
- Hope: perhaps we can just focus on Markov stationary policies?
  - stationary: doesn't depend on time
  - Markov: depends only on previous state
  - Can be finitely represented and computed, *but is it always optimal*?

Restriction to Markov stationary policies

- ***Proposition: stationary Markov policies do not suffice even in adversarial patrolling games.***
- ***Proof sketch: if the defender is very impatient and the attacker is very patient, the defender can have a policy which is very attractive for the attacker if he only waits a few rounds.***
- ***In practice, even though Markov stationary policies can be suboptimal, they are very natural and non-stationary policies are difficult to implement. We assume that the defender is restricted to such policies.***

Mixed-Integer Non-Linear Program to Compute Markov Stationary SSE

$$\max_{\pi,\phi,V_L,V_F} \sum_{s\in S} \beta(s) V_L(s)$$

subject to:

$$\pi(a_l|s) \geq 0 \qquad \forall s, a_l$$

$$\sum_{a_l} \pi(a_l|s) = 1 \qquad \forall s$$

leader policy is a valid probability distribution

$$\phi(a_f|s) \in \{0,1\} \qquad \forall s, a_f$$

$$\sum_{a_f} \phi(a_f|s) = 1 \qquad \forall s$$

follower policy is deterministic (can only choose one action)

$$0 \leq V_F(s) - \tilde{R}_F(s,\pi,a_f) \leq (1-\phi(a_f|s))Z \;\forall s, a_f$$

$$V_L(s) - \tilde{R}_L(s,\pi,a_f) \leq (1-\phi(a_f|s))Z \;\forall s, a_f$$

follower plays a best response to the leader

Sandia National Laboratories

Approximating SSE through Discretization

- MINLP too hard to solve; better: approximate optimal policy by discretizing the probabilities
- Bilinear constraints now have integer variables, and we can use McCormick inequalities to linearize these
- *End result: MILP for approximating SSE in general Stochastic games*

Impact of discretization

- **Theorem**: Can bound the impact of discretization in general **finite-action** Stackelberg games.
- *Proof uses the multiple LP algorithmic approach for computing SSE in general finite Stackelberg games.*
- Corollary: if we restrict the defender to Markov stationary policies, discretization will converge.

The value of discretization

	Exp Utility	Running Time (s)
MINLP (5 states)	9.83	375.26
MILP (5 states)	10.16	5.28
MINLP (6 states)	9.64	1963.53
MILP (6 states)	11.26	24.85

MILP approximation (using CPLEX) much faster,
and better solutions than MINLP (using KNITRO + restarts)

Computing a Stackelberg Equilibrium in APGs

- In zero-sum APGs, we can actually get rid of integer variables
- What remains is a non-linear non-convex program

Application: APGs

- **Zero-sum game: defender wants to minimize attacker values**

$$\min_{v,\pi} \sum_i v_i$$

defender tries to make constraints bind at the lowest possible values

subject to

Compute attacker value

$$v_i \geq (1 - \pi_{ij}) u_j$$

$$v_i \geq \delta \sum_j \pi_{ij} v_j$$

Valid probability distribution

$$\pi_{ij} \geq 0$$

$$\sum_j \pi_{ij} = 1$$

graph constraint → $\pi_{ij} \leq A_{ij}$

APG Extensions

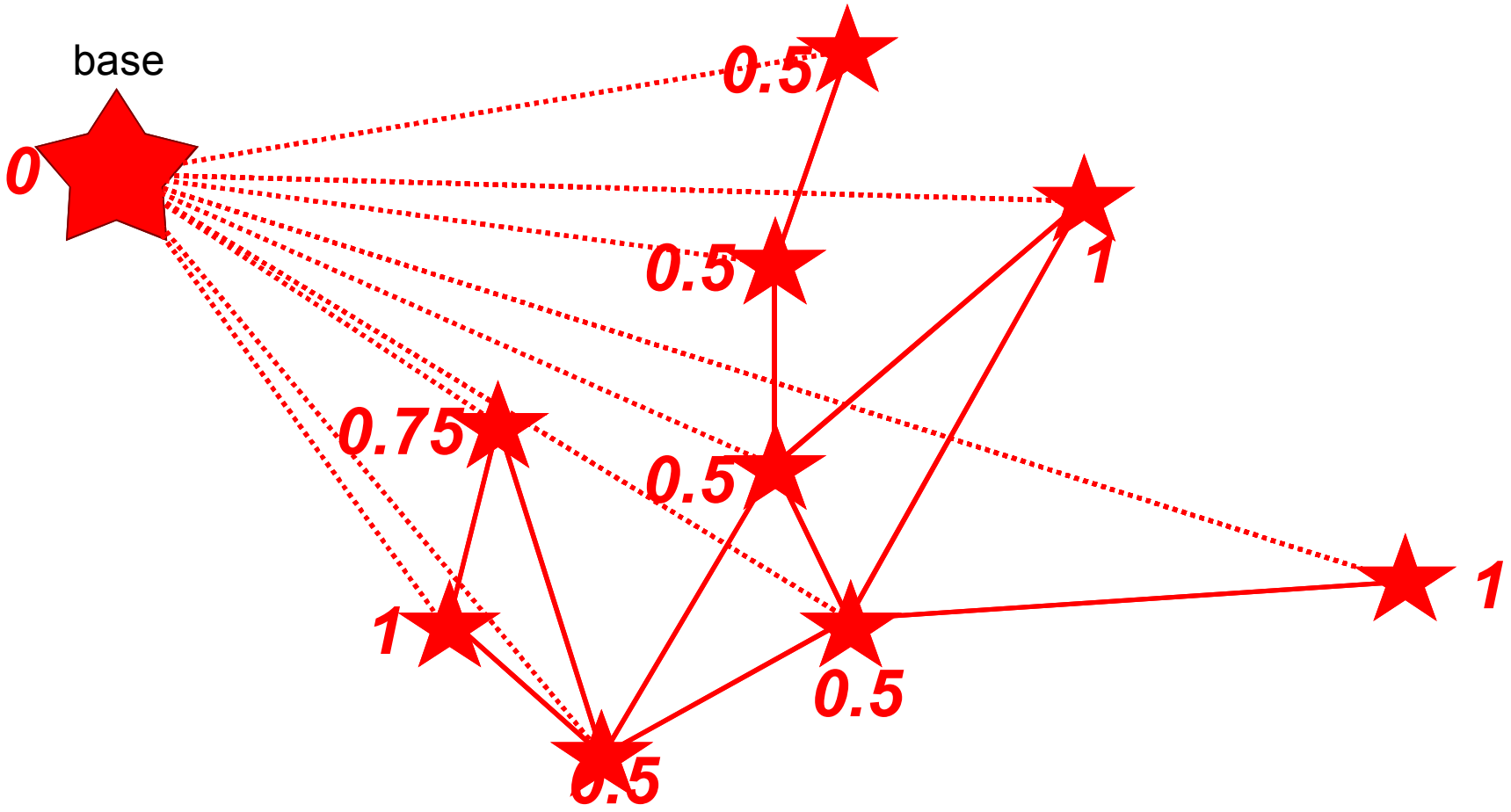
- Can allow one to have multiple defense resources (e.g., patrol boats/cars/etc)
- Defender chooses coverage vectors
  - for each target, 1 if it is covered, 0 otherwise
- State = coverage vector (observed by attacker)
- Graph constraints imply constraints on moves between coverage vectors
  - Consider a move from s to s'
  - Construct a bipartite graph with links between covered targets in s and those in s' induced by the constraint graph; call this graph G
  - Theorem: a move from s to s' is feasible iff G has a perfect matching

APG Extensions

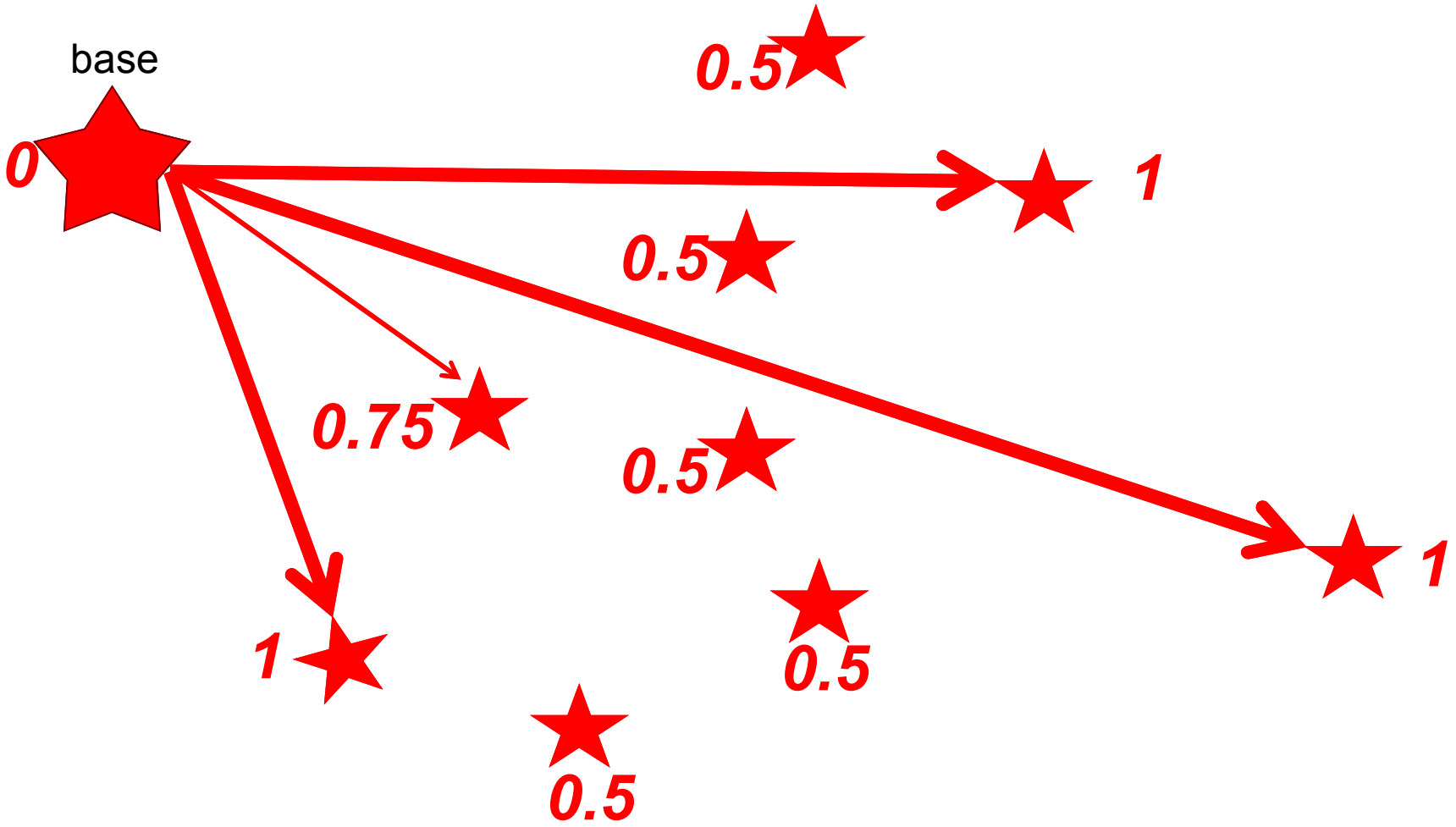
- Can also consider settings in which attacks take more than one time step to deploy
- State s is now a sequence of defender moves

USCG Illustration

Sandia National Laboratories

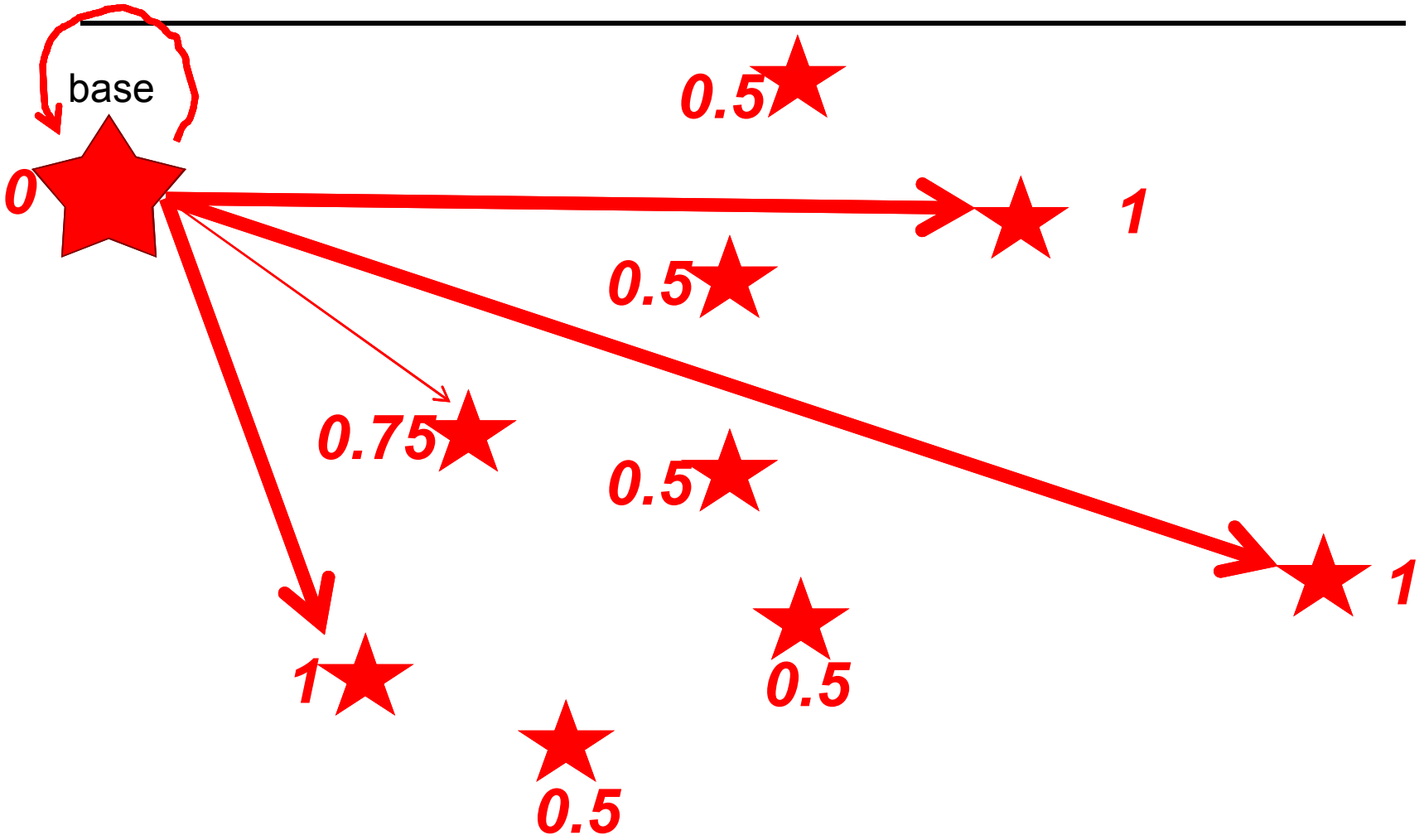


δ = 0.5 (impatient attacker)



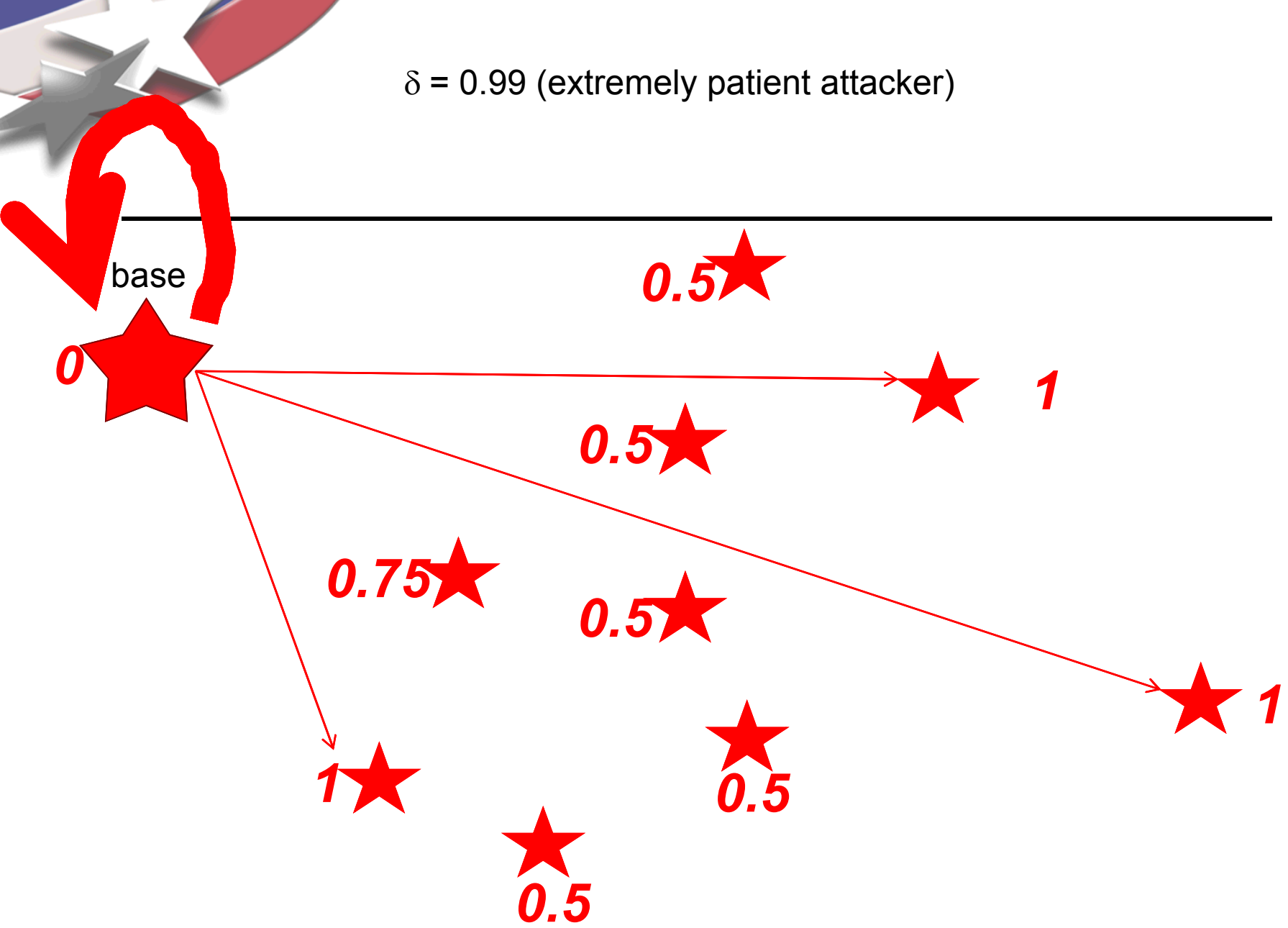
(return to base from every target with positive probability)

δ = 0.75 (moderately patient attacker)



Sandia National Laboratories

$\delta = 0.99$ (extremely patient attacker)



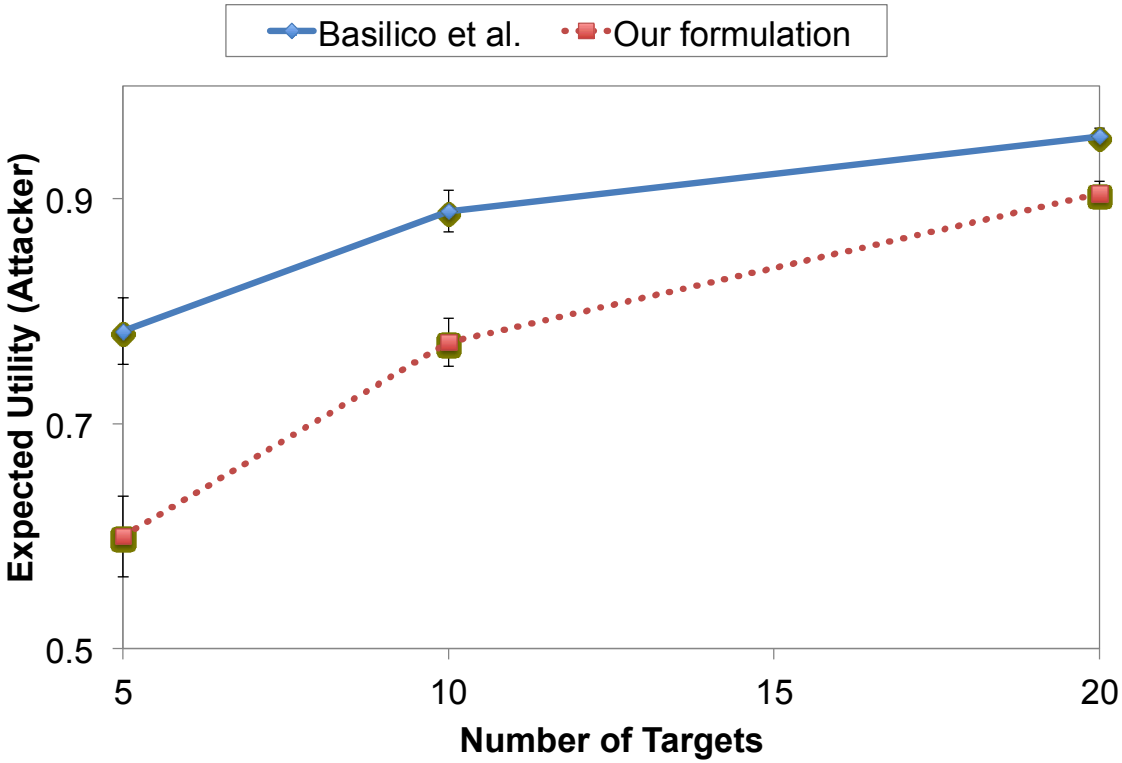
Experiments: Adversarial Patrolling on Exogenous Graphs

Sandia National Laboratories

Related Work

- Basilico et al. 2009-2011: math programming formulations
  - No discounting
  - General-sum
  - An attack can take more than one time step
  - Substantially different formulations from ours

Comparison to Basilico et al.



Basilico et al. clearly suboptimal, ***even when discount factor = 1***!

Sandia National Laboratories

Summary

- Model patrolling problem with an intelligent adversary as an APG, a special case of Stochastic Stackelberg games (SSGs)
- SSGs always have equilibria in Markov stationary policies
- Can solve exactly in finite time, and approximate arbitrarily well by discretizing the probabilities
- Discretization yields a MILP which is much faster and yields better solutions using state-of-the-art optimizers
- APGs can be solved much faster if they are zero-sum, and solutions are much better than state-of-the-art