Computational Economics (CS 396) Progress Report

Brian Gauch
brian.gauch@vanderbilt.edu

ABSTRACT

Game theoretic approaches, particularly Stackelberg games, have been widely deployed in recent years to allocate security resources. Most existing security game models assume that a security resource assigned to a target can only protect that target. However, in many important real-world security scenarios, when a resource is assigned to a target it exhibits protection externalities; that is, it simultaneously provides protection for nearby targets. We build off of the formulation in Gan et al. [Gan] that incorporates protection externalities within a Stackelberg game. They made no assumptions about the topology of the space which the targets and defense resources are in. However, this required that resources were constrained so that they could only be located on a target. We note that in many cases security problems are fundamentally planar and develop a new model that incorporates this assumption. By constraining the problem to a planar space, we are able to consider a more flexible set of locations for resources, i.e, we remove the restriction that resources must be located at targets. We then evaluate the effectiveness of our model in comparison to Gan's model in terms of overall defender solution quality.

1. INTRODUCTION

1.1 Motivation and Previous Work

Game theory is currently being applied to model security for airports, ports, shipping and other vulnerable infrastructure. Much of the research has focused on Stackelberg games, games in which the attacker knows the defender's strategy when choosing their own through *a priori* surveillance [Korzhyk 11]. The goal of the game, from the defender's perspective, is to pick a strategy, i.e., a probability distribution over resource allocations, that maximizes their utility. An assumption in most existing security game models is that a security resource assigned to a target protects only that target. However, in many real-world security scenarios, when a resource is assigned to a target, it exhibits protection externalities; that is, it also protects other "neighboring" targets.

One formulation that includes protection externalities is described in Gan et al. [Gan]. Their model makes no assumptions about the topology of the space in which the targets and resources are placed, however they do include the restriction that a resource must be allocated "at" exactly one target in some sense. This restriction is undesirable because there are many scenarios in which one might want to place resources at locations that are not co-incident with a target. Consider the situation in which there are more targets than resources; one might want to place the resources between the targets so that, if the targets are close together, a single resource could cover more than one target.

To do this, we need to specify the metric space in which the resources and targets lie. In many of the applications where externalities exist in the allocation of defense resources, one can imagine modeling the problem with targets on a plane and defense resources defending a disc on the plane. When a defense resource is a camera, radar, or mobile unit, the region that it defends is roughly a disc. Accordingly, we choose for our metric space a plane. Admittedly, this model is less suited to urban areas where line of sight is important. This model is also unsuited to cyber security or the defense of multi-story buildings, because the planar assumption is likely to be violated.

1.2 Accomplishments

We have a working prototype of a simple security model that does not incorporate protection externalities when allocating defender resources to protect targets. We have initial versions of all of our modules and are able to calculate the expected defender utility for this naïve version. By making use of an external Nash equilibrium solver, we are able to calculate the expected defender utility for a security game.

2. MODEL

Gan et al. describe a model for allocating resources to targets that incorporates protection externalities. We use this model as our inspiration and it is the model to which our approach will be compared. It is described more fully in Section 2.1. In Section 2.2, we describe our modifications to their model and discuss the pros and cons of our approach.

2.1 Gan's Model of Externalities

In [Gan], Gan et al. describe a Stackelberg game-based security model in which the defender allocates resources to a set of targets given that the resource has the externality of defending other neighboring targets.

The defender has *d* resources and a set of *t* targets to defend. The attacker's available actions (who can only attack one target) are exactly the set *t*. The set of actions available to the defender varies based on the experiment, but always corresponds to points on the plane so that the set of defended targets can be determined.

These actions are used to generate a game where the payoff for each pair of actions depends on the value of the target and whether or not the attacker got caught (the defender defended the target that the attacker attacked). More generally, the defender could have different values for targets that the attacker (the game could not be zero-sum).

Their algorithm assumes the availability of an adjacency matrix A indicating target adjacencies. A pure strategy for the defender corresponded to an allocation of integer resources/defenders to targets, such that each target was either defended or not defended (1 or 0). The authors acknowledged the limitation of only locating resources at targets and not between targets, but noted that a dummy target with no value to the attacker or defender could be placed at such locations. The set of actions available to the defender is the set of subsets of *t* of size *d*.

2.2 Allowing Off-target Resource Location

Previously, the set of actions available to the defender was the set of subsets of *t* of size *d*. Now, we would like to somehow allow resources to be placed not on targets. So, instead of taking in as input an adjacency matrix for targets, we need to define a metric space so that we can determine distances, and therefore adjacencies, ourselves. Although it would be natural to consider defending a three-dimensional space, for simplicity, we choose for our metric space a plane. In summary, we adding a new feature to Gan's model, off-target resource placement, but at the cost of introducing a new restriction, i.e., requiring that all defenders and resources inhabit a two-dimensional world.

When a defense resource is a camera, radar, or mobile unit, the region that it defends is roughly a disc. Therefore, the most natural model is for a defense resource to defend a unit disk.

Consider the case where each resource defends a unit disk. Now, instead of considering defense resources as disks and targets as points, consider targets as disks and resources as points. In both problems, consider a target to be defended by a resource if the point lies within the disk. These problems map directly from one to other, in the sense that a resource defends a target in the first problem if and only if it defends that target in the second problem, as long as the disks in both problems have the same radius. In this new formulation, our goal is to find points (resource locations) that are not strictly dominated by another point. That is, point p1, which lies in a set of disks S1, is considered to be a good resource location if ¬∃ p2 such that S1 is a subset of S2. We also want to avoid adding multiple actions which defend exactly the same set of targets – that is, we want to avoid adding p1 if ∃ p2 such that S1 is exactly S2 and p2 was already added as an action to consider during the game proper.

Now consider the intersection graph of disks on a plane. If a point on the plane lies in a set of S disks, then there is a clique in the intersection graph containing exactly the members of S. Out first requirement from above corresponds to requiring that the clique be maximal. Our second requirement is that no two chosen points correspond to the same clique. So, to ensure that we satisfy both requirements, we will not directly choose exact point locations but instead choose among maximal cliques, and afterwards choose arbitrary points from within the corresponding region in the plane.

Now, the set of actions available to the defender corresponds to a subset of the set of maximal cliques in the unit disk graph with disks centered on targets. We consider only a subset because there could be exponentially many maximal cliques [Gupta].

It is worth reconsidering our model of resources defending targets within a unit disk at this point. If we instead model a defense resource as defending an axis-aligned rectangle, finding un-dominated actions would instead correspond with finding maximal cliques in a boxicity 2 graph, and boxicity 2 graphs have a polynomial number of maximal cliques. We will continue modeling with disks nonetheless.

There is a fundamental tradeoff between the number of actions considered and the time to solve the resulting game. Because the solution to the SPE corresponds to a set covering, each target should be defended by at least one action to allow for a good solution.

What makes a selection of maximal cliques good? That is, what actions should be added first when attempting to allow for good solutions to the game? There are a number of natural heuristics one might consider. One could greedily add the largest maximal cliques to the set of actions. One could greedily add the action that protects the most as-yet-unprotectable targets. One could imagine more sophisticated algorithms searching for a set of actions with backtracking. We will investigate the effectiveness of various heuristics in one of our experiments.

3. IMPLEMENTATION

Our model is implemented in C++ on a linux platform. It is comprised of three major modules, namely, the *world builder*, the *resource location selector*, the *game initializer*, and the *game solver*. Each of these is described in more detail below.

3.1 World Builder

The *world builder* is given the size of the world and the number of targets. It produces a mapping of targets to locations. Targets are placed uniformly at random locations.

Status: This is completed.

3.2 Resource Location Selector

The *resource location selector* is given the target placements, the size of the world, the number of targets, and which model to use. It produces a set of resource locations to be considered during the game proper. It implements Gan's resource placement model and our planar, maximal clique algorithm. The resource location algorithm used is controlled by a flag for experimental purposes.

Status: The implementation for Gan's version is completed.

3.3 Game Initializer

The *game initializer* is given as input the target placements, the possible resource placements, and the target values. It produces a normal form game utility matrix for use by the game solver. It has three main tasks:

1) It enumerates attacker actions that are simply the set of targets

2) It enumerates defender actions by selecting subsets of selected resource placements;

3) It then calculates the utility of each action pair by determining whether or not the attacker attacked a defended target, and using the value of the attacked target.

Status: The game initializer is completed, without target values (all targets have value 1).

3.4 Game Solver

The *game solver* is given a game in normal form utility matrix form, and outputs the expected utility of the defender.

Status: We downloaded a Nash equilibrium solver from http://cgm.cs.mcgill.ca/~avis/C/lrslib/USERGUIDE.html#nash.

4. RESULTS

4.1 Experimental Design

We will compare three models: 1) a simple security game with no protection externalities; 2) Gan's model; 3) our planar, maximal-clique model. We will run each model on worlds of varying sizes and compare the results based on the expected defender utilities each produces.

4.2 Results

We ran Gan's model on several test cases to validate our implementation and report the utility. In each run, we placed 5 targets within a 1x1 unit square world, uniformly at random. We varied the number of resources available to the defender from 1 to 4 and the resources' effective range from 0.0 to 0.4. For each data point below, we ran 4 test cases and report the average. As we expected, we observe that as the number of resources increases and/or the resource's range increases, the expected utility increases. We also note that when the range is 0, the defender's best strategy is to place resources uniformly at random.

Table 1. Preliminary Results

No. of Resources	Resource Range		
	0.00	**0.20**	**0.40**
1	-0.80	-0.78	-0.46
2	-0.60	-0.58	-0.25
3	-0.40	-0.17	-0.08
4	-0.20	-0.05	0.00

5. RELATED WORK

Game theory in its present form was originally presented by von Neumann in his 1928 paper. Although most closely tied to economics, almost since its inception game theory has been applied to security problems, most famously during the 1950s to model global nuclear strategy. Also during the 1950s, Nash famously developed a criterion for mutual consistency of players' strategies in non-cooperative games, known as the Nash equilibrium [Wiki]. This sparked great interest in the field as many research groups extended the models, defining for example repeated and extensive form games, and applied them to new areas.

More recently, thanks to increased computational power and new, efficient algorithms, it has become practical to use Stackelberg games to model real-world security problems. Because it seems natural to assume that the attacker conducts surveillance, Stackelberg games became more popular for such problems than models in which both players act simultaneously. One of the key revitalizing papers was Brown et al., 2006 [Brown 06] in which they argue that critical infrastructure defense must become more sophisticated to be adequately protect against terrorist attacks. They advocate the use of optimization models, in particular, the Stackelberg model, as a way to model attackers and defenders to develop a robust protection system. Real world applications include the ARMOR security system deployed at the Los Angeles International Airport [Pita, 2008] and the IRIS system deployed by the Federal Air Marshals Service to allocate marshals to tours of duty to protect commercial flights [Jain, 2010].

Gan et al. [Gan] term games in which one defense resource can defend multiple nearby targets Security Games with Protection Externalities (SPEs). They use the term externality because each resource is assigned to one primary target, and may defend other nearby targets, as a bonus. They show that finding Stackelberg equilibria in such games is NP-hard. Nevertheless, they show that polynomial approximations perform better than ignorant solutions when proximity-based protection externalities exist.

Fang et al. [Fang 2013] and Xu et al. [Xu 2014] did not call the effect an externality, but studied patrol paths on the real number line with moving targets, where a defense resource had an effective range. Because a defense resource was not bound to a particular target, but was instead constrained to a particular metric space, this work is in some senses more similar to our own than Gan's.

6. CONCLUSIONS

6.1 Summary

This report presents our planar, maximal clique model for allocating defender resources to protect targets. Our version extends the protection externalities described by Gan to allow for resource locations other than at target locations. To solve this problem we have defined a metric space for the game world, in our case, a plane. We have completed initial versions of each of our four modules.

6.2 Future Work

We plan to extend several of the existing modules. Most future development will focus on improving the resource location selector. We have Gan's model implemented but we have not yet implemented our planar, maximal clique model. We will also extend our utility calculation in the game initializer to incorporate target values. Finally, we will replace the downloaded game solver with our own implementation from homework 3.

In the longer range, outside of the scope of this project, we would ideally like to use an algorithm such as CLASPE to approximately solve the game as a Stackelberg game. We could also extend the resource location selector to take target values into account when choosing among maximal cliques.

7. REFERENCES

[1] Jiarui Gan, Bo An, and Yevgeniy Vorobeychik. Security games with protection externalities. (AAAI, 2015, to appear).

[2] Gupta, Rajarshi, Jean Walrand, and Olivier Goldschmidt. "Maximal cliques in unit disk graphs: Polynomial approximation." Proceedings INOC. Vol. 2005. 2005.

[3] Fang, F.; Jiang, A. X.; and Tambe, M. 2013. Optimal patrol strategy for protecting moving targets with multiple mobile resources. In *Proceedings of the 12th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS'13)*, 957–964.

[4] Brown, Gerald, et al. "Defending critical infrastructure." Interfaces 36.6 (2006): 530-544.

[5] Xu, H.; Fang, F.; Jiang, A. X.; Conitzer, V.; Dughmi, S.; and Tambe, M. 2014. Solving zero-sum security games in discretized spatio-temporal domains. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI'14)*.

[6] Pita, James, et al. "Deployed ARMOR protection: the application of a game theoretic model for security at the Los Angeles International Airport." Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[7] Jain, Manish, et al. "Security Games with Arbitrary Schedules: A Branch and Price Approach." AAAI. 2010.

[8] Korzhyk, Dmytro, Vincent Conitzer, and Ronald Parr. "Complexity of Computing Optimal Stackelberg Strategies in Security Resource Allocation Games." AAAI. 2010.

[9] Wikipedia contributors. "Game theory." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 1 Mar. 2015. Web. 4 Mar. 2015.