ALMA: Active Law-enforcement Mixed-strategy Allocator

Atharva Naik (annaik2) and Apurv Singhdeo (apurvas3)

University of Illinois Urbana-Champaign, Champaign IL 61820, USA
https://www.cs.illinois.edu

1 Introduction and Background

The Active Law-enforcement Mixed-strategy Allocator (ALMA) is a game-theoretic system to determine the optimal police patrol routes for the University of Illinois. The aim of this project is to use real-world data to explore possibilities to make our campus safer. Using a combination of existing Stackelberg Security Games (SSG) research and information from the police department, we uniquely create a program that creates patrol schedules that outperform random simulations.

A *Stackelberg Security Game* is a model for resource allocation in adversarial environments. In general, this is a two-player game where one player is a leader (*defender*) and the other player is a follower (*attacker*). The defender commits to a strategy first, and the attacker optimizes her reward based on the action chosen by the defender. The objective is to find the optimal mixed strategy for the defender.

Stackelberg Security Games have been widely implemented in practice to allow defenders like state protect against various attackers like terrorists, poachers, etc. For example, IRIS [6] is an air marshall patrol scheduler that outputs a randomized schedule of flight assignments for marshalls to maximize air travel security. PROTECT [5] outputs maritime patrol schedules that newly models bounded-rational attackers. And TRUSTS [4] uses a time-expanded network for policing trains and train stations for the LA Metro system. Based on this background and our status as hardcore Illini, we pose the natural question: how might we find the optimal patrol routes for the University of Illinois Police Department (UIPD) to make our campus safer?

1.1 Context in Champaign

The City of Champaign can be modeled as a graph. We are in a unique position where we have three different police departments: the Champaign Police Department (CPD), the Urbana Police Department (UPD), and the University of Illinois Police Department (UIPD). Each department has it's own jurisdiction with their own resources and separate priorities. For example, the CPD will be more interested in defending Downtown Champaign than the UIPD, but the UIPD will have a vested interest in protecting the dorms and/or research facilities. Each of these also patrol a subset of the graph of Champaign, and this patrol schedule may or may not overlap.

It is also worth discussing the profile of crime that is seen on the UIUC campus. We use the model that the crime is opportunistic and non-adaptive. Opportunistic criminals only commit crimes if there is a good opportunity, which means that police presence will change if and where crime is attempted. [1] It is non-adaptive in the sense that attackers are less likely to observe and learn from patrol routes.

1.2 Our Model

Based on the above context, the ideal model for a security game for Champaign is a multi-defender security game with a bounded-rational (opportunistic) attackers. [1] Also, time and date may play a role in the attacker's decisions, so a time-expanded graph would be appropriate. [3] Being a University campus, it also makes sense that certain nodes in the graph are designated as important, or given a higher reward if covered while being attacked.

Due to data and time constraints, we had to settle for a much simpler model. Since we only recieved data from UIPD, we chose to specifically focus on the campus area in Champaign. This closely resembles the existing patrol zone/jurisdiction of the UIPD. We have a single-defender single-attacker model with perfectly rational attackers. There is also no further importance attached to each node. The sections below detail the data and methods used.

2 Data

The University of Illinois Police Department is required by the Clery Act, a federal law, to collect and report crime statistics and campus safety policies. The university of mandated to collect, classify, and count crime reports and statistics related to crime and maintain a daily crime log of alleged criminal incidents which is open to public inspection. Through this act, we are able to get the UIUC Crime Log starting on January 1, 2021 until October 31, 2025. This gives us the following important infromation regarding crimes on campus: description, location (address), time of crime, and disposition. We also use OpenStreetMaps to obtain a campus map with information about roads and intersections. Finally, we collected data about UIPD resources such as patrol units, cars, and officers to identify resource constraints for the defender.

3 Methods

3.1 Data Processing

In order to assign meaningful weights to the graph, we have to measure the severity of crime at each node of the graph. The data we have only provides a description of the crime, and it becomes a challenge to quantify these. For example, crimes like assault are much more severe than a noise complaint. In order to identify the severity of the crime, we use `gpt-5-nano` with the following system prompt:

```
You are a strict crime severity classifier. Output only a single digit 1-5.
Scale:
1 = minor/administrative (e.g., underage drinking, noise),
2 = low harm (trespass, vandalism),
3 = property crimes (most theft),
4 = threat/force without severe injury (robbery, burglary),
5 = violent/sexual/weapons or severe harm (aggravated battery, sexual assault).
```

This system prompt allows us to pass in a description of the crime and obtain a severity rating between $[1, 5] \in Z$ for the crime.

We also want the location of the crime based off the address. We use the Google Maps Geocoding API, which enables the conversion of an address to it's latitude and longitude coordinates. With these steps, we have successfully processed our crime data.

3.2 Graph Design

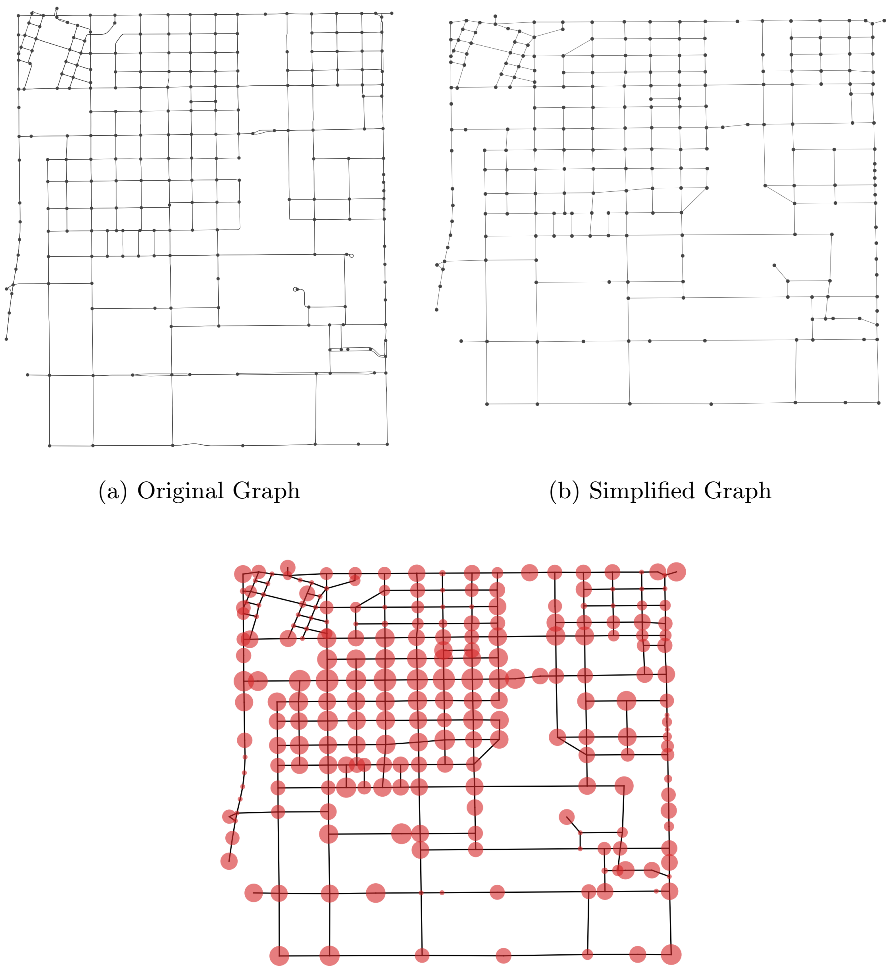
Using data from OpenStreetMaps, we can model the UIUC campus as a graph in NetworkX. This creates a graph $G = (V, E)$ as follows:

- $v \in V$ is v is an intersection of roads
- Each node v has the following attributes:
  - Latitude and longitude
  - Crimes assigned to the node based on nearest latitude and longitude
  - Risk factor as a function of the severity and number of crimes that take place at that node. This is given by the formula

    $$R_v = S_{avg} \times (1 + \log(1 + N_{crimes}))$$

    Where S_{avg} is the average severity of crimes at that node, and N_{crimes} is the total number of crimes at the node
- $(u, v) \in E$ if there is a road connecting u and v

Using this, we get the graph of Champaign as shown in Fig. 1(a). We note that there are nodes in this graph that are too close together (eg: Lincoln Ave, and College Ct). We simplify the graph by merging such nodes, as seen in Fig. 1(b). Intuitively, this helps us consolidate the complexity of our route and should have no effect on the optimal patrol, as police presence in the area is already a deterrent. Fig 1(c) shows the graph with the risk factor of each node.



(a) Original Graph

(b) Simplified Graph

(c) Risk-factor Graph

Fig. 1: Graph representation of the map of Champaign

4 Game Design

We model patrol allocation as a Stackelberg Security Game (SSG) on the simplified campus graph $G = (V, E)$ where each node $v \in V$ is a potential target. The defender (UIPD) commits to a mixed strategy that is coverage probability $x_v \in [0, 1]$ for each node v, where x_v represents the probability that node v is covered by a patrol unit at an arbitrary time.

4.1 Payoffs from risk factors

Each node has a risk factor R_v (Section 3.2), which we use as a measures for how valuable that node is to the attacker and how costly it is to the defender if attacked. We assume each target has four payoffs: attacker reward/penalty and defender reward/penalty. Our payoff design follows two monotonicity assumptions:

- it is always better for the defender to cover a target than to leave it uncovered
- it is always better for the attacker to attack an uncovered target.

Accordingly For each target $v \in V$, we define four payoffs, two for the attacker and two for the defender:

- $R_v^a = \alpha \cdot R_v$: attacker reward if v is not covered.
- $P_v^a = -\beta \cdot R_v$ attacker penalty if v is covered.
- $R_v^d = \gamma \cdot R_v$: defender reward if v is covered.
- $P_v^d = -\delta \cdot R_v$: defender penalty if v is not covered.

The scalar multipliers $\alpha, \beta, \gamma, \delta > 0$ control payoff magnitudes:

- α scales how valuable a successful attack is to the attacker.
- β scales how costly it is to the attacker to attack a covered target.
- γ scales the defender's benefit from covering a target.
- δ scales the defender's cost when a target is attacked while uncovered.

so that higher-risk nodes have proportionally larger payoffs for both players, and the signs encode reward and penalty of covering a node.

4.2 Solving the SSG (linear program)

We solve for an optimal defender mixed strategy using the standard SSG linear-program formulation[2].

$$\boxed{\begin{array}{ll} \text{maximize} & U_d(t \mid x) \\ \text{s.t} & 0 \le x_i \le 1 \quad \forall i \\ & \sum_i w_i x_i \le K \\ & U_a(t \mid x) \ge U_a(j \mid x) \quad \forall j \end{array}}$$

Here $U_a(i \mid x) = x_i P_i^a + (1-x_i) R_i^a$ is the attacker's expected utility for attacking node i, and $U_d(t \mid x) = x_t R_t^d + (1-x_t) P_t^d$ is the defender's expected utility when the attacker attacks t.

Because the attacked target t is not known in advance, we solve this LP once for each candidate target $t \in V$ by enforcing the best-response constraints $U_a(t \mid x) \geq U_a(j \mid x)$ for all $j \in V$. This yields a coverage vector $x^{(t)}$ for each t. We then select the solution with the largest defender objective value (equivalently, the best $U_d(t \mid x^{(t)})$) as the defender's final coverage vector.

4.3 Patrol Generation Protocol

The SSG solver returns a marginal coverage vector $x \in [0,1]^{|V|}$, where x_v is the coverage probability assigned to node v. To generate feasible patrol routes on the graph, we convert x into a Markov transition matrix P over adjacent nodes. For each node u and neighbor $v \in N(u)$, we define an transition weight

$$w(u \to v) = x_v.$$

We then normalize these weights over the neighbors of u:

$$P_{uv} = \frac{w(u \to v)}{\sum_{v' \in N(u)} w(u \to v')} = \frac{x_v}{\sum_{v' \in N(u)} x_{v'}}.$$

If $\sum_{v' \in N(u)} x_{v'} = 0$ (e.g., all neighboring nodes have zero assigned coverage), we fall back to an unbiased random walk. This produces a randomized yet feasible patrol movement policy that preferentially moves toward nearby nodes that the SSG assigns higher coverage.

4.4 Simulation

Given K patrol units, we initialize each unit at a chosen starting node and simulate movement for a fixed number of time steps T. At each step, every unit at node u moves to a neighbor v sampled from $P_{u\cdot}$. The resulting sequence of visited nodes for each unit constitutes a patrol schedule. At each time step, a crime event occurs with probability p_{event}. A crime is counted as detected if at least one patrol unit is located at the crime node at the same time step. We measure performance using the interdiction rate:

$$\text{Efficiency} = \frac{\#\text{detected crimes}}{\#\text{total crimes}}.$$

To benchmark the benefit of using the SSG output, we compare against a uniform baseline patrol. In the baseline, patrol units follow an unbiased random walk on G, while keeping K, T, and p_{event} fixed. This isolates the effect of risk/coverage-aware scheduling from overall simulation settings.

5 Results

For each configuration (choice of K, T, p_{event}, and payoff multipliers $\alpha, \beta, \gamma, \delta$), we run multiple trials with different random seeds and report the mean efficiency. generation.

Overall, the SSG-derived patrol policy achieves higher interdiction efficiency than the uniform baseline across the settings we tested. Intuitively the improvement is caused by the SSG assigning higher marginal coverage to nodes with larger risk factor R_v, and the transition matrix conversion then biases patrol movement toward these high priority regions. Figure 2 shows the graph with our initial simulation. The graphs from varying p_{event} can be found in the appendix.

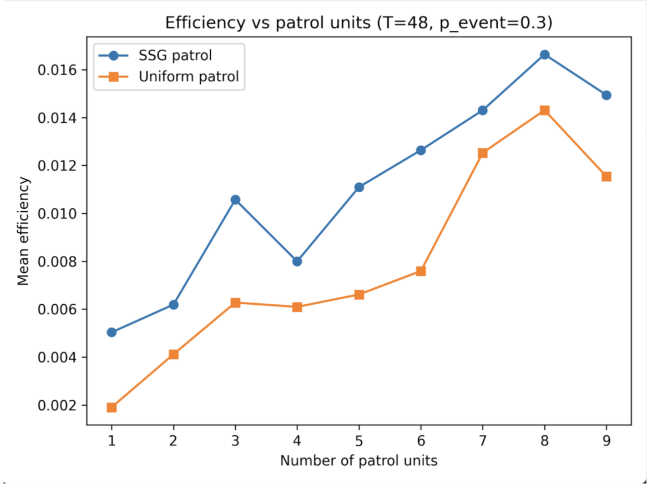


Fig. 2: Efficiency vs Patrol Units (T=48, p-event=0.3)

We also vary the payoff multipliers $(\alpha, \beta, \gamma, \delta)$, which control how aggressively the LP prioritizes high-risk targets. Across moderate changes in these parameters, the SSG-based policy continues to outperform the uniform baseline, suggesting that performance is not an of a single payoff configuration. The graphs from these results may be found in the Appendix.

5.1 Conclusions

We find that the ALMA system outperforms random simulations in all the cases that we simulated. In particular, we see a 20-50% increase in efficiency over random simulations. This shows promising results for a preliminary proof of concept, and we hope to develop this further in collaboration with the University and the UIPD. An immediate next step for improvement is to compare our results with the current patrol routes by the UIPD.

5.2 Next Steps

Given the success of our preliminary experiments, we are very interested in approaching the relevant stakeholders (UIPD, etc.) with our findings to develop this into implementation. The immediate next steps to improve the simulation and modeling can be broken down into two main parts: the graph, and the algorithm. The graph can have better alignment with police patrol zones, temporal routing, and node-prioritization. During our presentation, there was some discussion about improving the risk factor calculation for each node, which is a good direction for improvement. The algorithm itself can be extended to multi-defender, have better rewards/weights, and model bounded-rational (opportunistic) attackers. We hope to work with the UIPD in order to design some of these improvements and find actionable routing changes.

References

1. An, B., Tambe, M.: Stackelberg Security Games (SSG) Basics and Application Overview, pp. 485–507 (11 2017). https://doi.org/10.1017/9781316676714.021
2. Conitzer, V., Sandholm, T.: Computing the optimal strategy to commit to. In: Proceedings of the 7th ACM Conference on Electronic Commerce. pp. 82–90 (2006). https://doi.org/10.1145/1134707.1134717, https://doi.org/10.1145/1134707.1134717
3. Jiang, A.X., Yin, Z., Johnson, M., Tambe, M., Kiekintveld, C., Leyton-Brown, K., Sandholm, T.: Towards optimal patrol strategies for fare inspection in transit systems. AAAI Spring Symposium - Technical Report (01 2012)
4. Luber, S., Yin, Z., Fave, F., Jiang, A.X., Tambe, M., Sullivan, J.: Game-theoretic patrol strategies for transit systems: the trusts system and its mobile app. pp. 1377–1378 (05 2013)
5. Shieh, E., An, B., Yang, R., Tambe, M., Baldwin, C., DiRenzo, J., Maule, B., Meyer, G.: Protect: An application of computational game theory for the security of the ports of the united states. Proceedings of the National Conference on Artificial Intelligence **3**, 2173–2179 (01 2012). https://doi.org/10.1609/aaai.v26i1.8436
6. Tsai, J., Rathi, S., Kiekintveld, C., Ordóñez, F., Tambe, M., Networks, T.: Iris – a tool for strategic security allocation in transportation networks. Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS **2** (12 2011). https://doi.org/10.1017/CBO9780511973031.005

Appendix

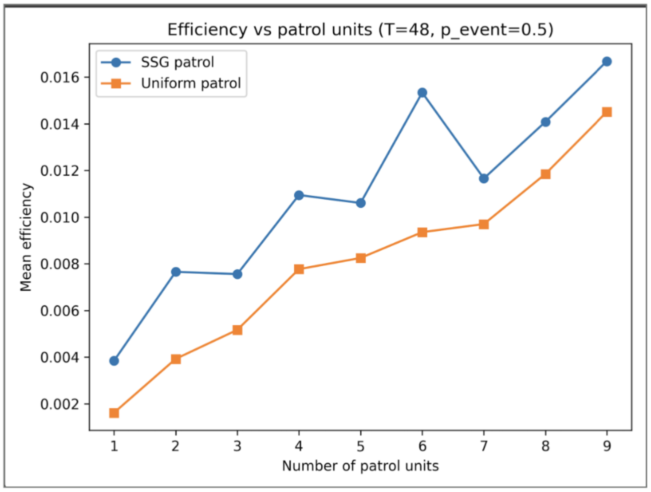


Fig. 3: Efficiency vs Patrol Units (T=48, $p_event = 0.5$)

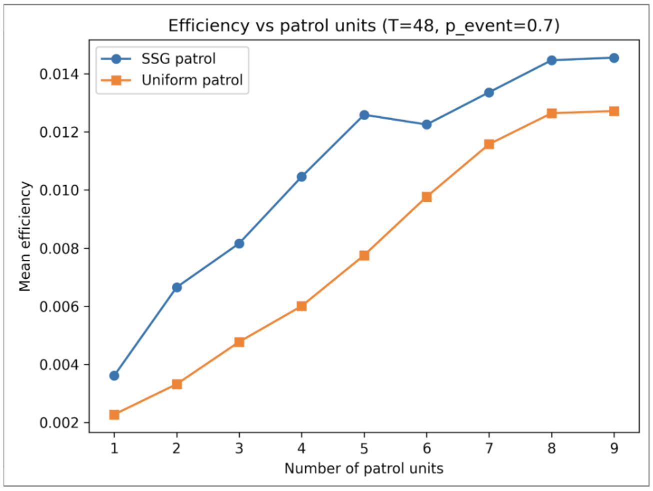


Fig. 4: Efficiency vs Patrol Units (T=48, $p_event = 0.7$)

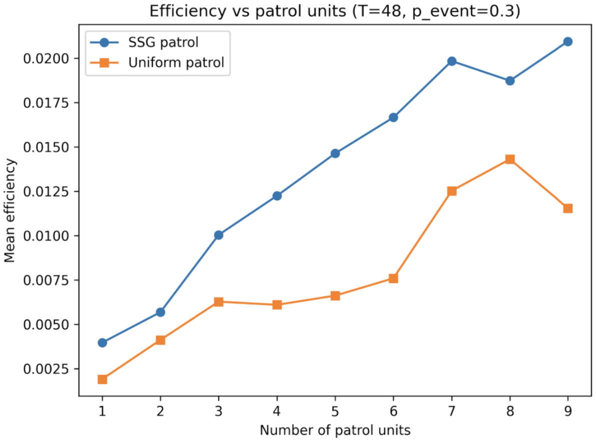


Fig. 5: Efficiency vs Patrol Units $(\alpha = 1, \beta = 0.9, \gamma = 0.9, \delta = 0.9)$

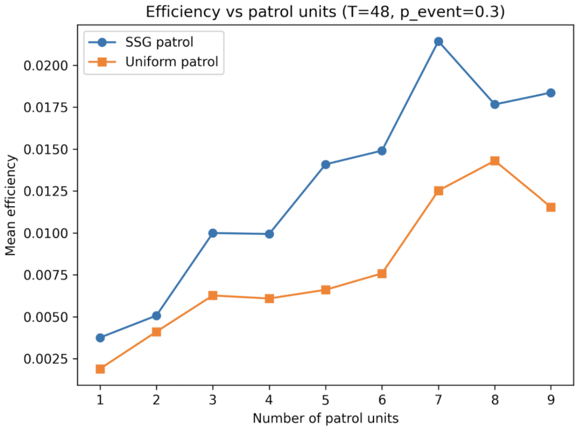


Fig. 6: Efficiency vs Patrol Units ($\alpha = 0.9, \beta = 1, \gamma = 0.9, \delta = 0.9$)

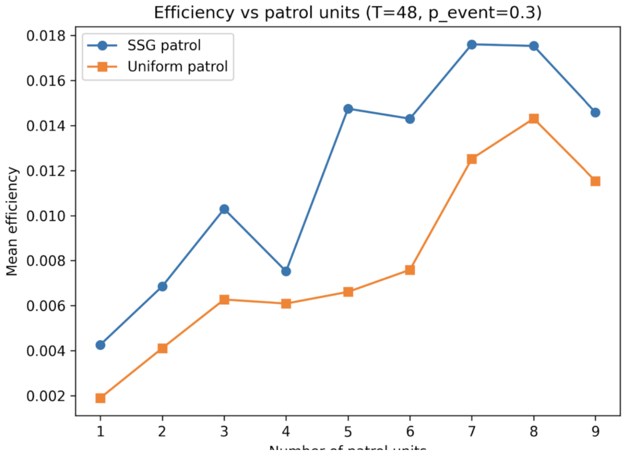


Fig. 7: Efficiency vs Patrol Units ($\alpha = 0.9, \beta = 0.9, \gamma = 1, \delta = 0.9$)

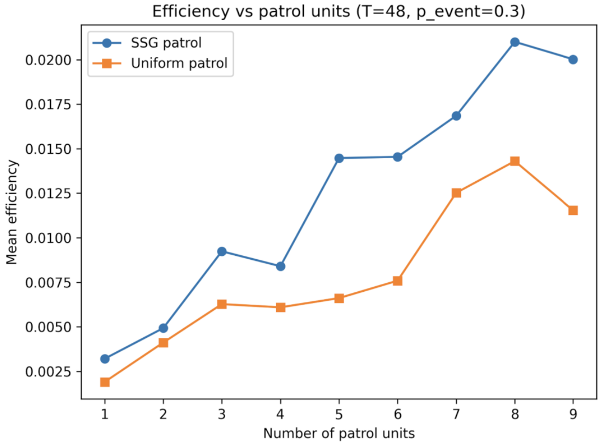


Fig. 8: Efficiency vs Patrol Units ($\alpha = 0.9, \beta = 0.9, \gamma = 0.9, \delta = 1$)