NECTAR: Game-Theoretic Factory Inspection Scheduling and Explanation for Toxic Wastewater Abatement *

Benjamin Ford*, Amulya Yadav*, Amandeep Singh+, Matthew Brown*,
Arunesh Sinha*, Biplav Srivastava†, Christopher Kiekintveld‡, Nicole Sintov, Milind Tambe*
*University of Southern California, Los Angeles, CA, 90089, USA
+Columbia University, New York, NY 10027, USA
†IBM Research, New Delhi, Delhi 110070, India
‡University of Texas at El Paso, El Paso, TX 79968, USA
*{benjamif,amulyaya,mattheab,aruneshs,sintov,tambe}@usc.edu,+as4330@columbia.edu,
†sbiplav@in.ibm.com,‡cdkiekintveld@utep.edu

ABSTRACT

Leather is an integral part of the world economy and a substantial income source for developing countries. Despite government regulations on leather tannery waste emissions, inspection agencies lack adequate enforcement resources, and tanneries' toxic wastewaters wreak havoc on surrounding ecosystems and communities. Previous works in this domain stop short of generating executable solutions for inspection agencies. We introduce NECTAR - the first security game application to generate environmental compliance inspection schedules. NECTAR's game model addresses many important real-world constraints: a lack of defender resources is alleviated via a secondary inspection type; imperfect inspectors are modeled via a heterogeneous failure rate; and uncertainty, in traveling through a road network and in conducting inspections, is addressed via a Markov Decision Process.

Previously unexplored in security game literature, NECTAR features a novel explanation system to improve user understanding of inspection schedules; understandability is a critical component to build trust and facilitate user adoption. This explanation system generalizes to any security game type, and we demonstrate its application to NECTAR. To evaluate our model, we conduct a series of simulations and analyze their policy implications. We also conduct a preliminary survey to assess explanation systems' potential impact on understandability.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms,Human Factors,Security

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), John Thangarajah, Karl Tuyls, Stacy Marsella, Catholijn Jonker (eds.), May 9–13, 2016, Singapore.*

Keywords

Game Theory, Inspection, Security Games, Explanation, Human-robot/agent interaction

1. INTRODUCTION

The leather industry is a multi-billion dollar industry [15], and in many developing countries such as India and Bangladesh, the tanning industry is a large source of revenue. Unfortunately, the chemical byproducts of the tanning process are highly toxic, and the wastewater produced by tanneries is sent to nearby rivers and waterways. As a result, the Ganga River (along with many others) has become extremely contaminated, leading to substantial health problems for the large populations that rely on its water for basic needs (e.g., drinking, bathing, crops, livestock) [12]. Tanneries are required by law to run wastewater through sewage treatment plants (STPs) prior to discharge into the Ganga. In many cases, however, the tanneries either do not own or run this equipment, and it is up to regulatory bodies to enforce compliance. However, inspection agencies have a severe lack of resources; the combination of the tanneries' unchecked pollution and the inspection agencies' failure to conduct inspections forced India's national environment monitoring agency to ban the operation of 98 tanneries near Kanpur, India with a further threat of closure for approximately 600 remaining tanneries [14]. It is our goal to provide agencies with randomized inspection plans so tanneries reduce harmful effluents and an important facet of India's economy can operate in a sustainable fashion. However, we recognize that the intended users of these plans (inspectors with backgrounds in Hydrology and the physical sciences) have not used randomized schemes in the past and may not be familiar with game theory or optimization techniques. [20] observed that user perceptions on ease of use and solution quality have a significant impact on user adoption of information technology; if the randomized solution cannot be understood by users (that are not experts in the randomization process), the solution risks not being adopted.

In this paper, we introduce a new game-theoretic application, NECTAR (**N**irikshana for **E**nforcing **C**ompliance for **T**oxic wastewater **A**batement and **R**eduction)[1], that in-

[*] The majority of this work has been previously published, but this paper introduces a novel explanation component applied to this domain.

[1] Nirikshana, the Hindi word for inspect. As many mythological stories and even popular Bollywood songs attest, Ganga

corporates new models and algorithms to support India's inspection agencies by intelligently randomizing inspection schedules. While we build on previous deployed solutions based on Stackelberg Security Games (SSG) for counterterrorism [18] and traffic enforcement [6], NECTAR represents the first security game application to directly address user adoption concerns by introducing a novel solution explanation component. Our SSG models are also the first to focus on the problem of pollution prevention by modeling the interaction between an inspection agency (the leader) and leather tanneries (many followers) - an interaction which poses a unique set of challenges. (i) Because there is a large disparity between the number of inspection teams and the number of tanneries, inspection plans must be efficient. (ii) We cannot assume that inspectors can catch 100% of violations. (iii) Inspectors must travel to the tanneries via a road network so solutions must be robust to delays (e.g., traffic). Finally, current fine policies may not be sufficient to induce compliance, and (iv) it is important to investigate alternative fine structures.

NECTAR addresses these new challenges of tannery inspections. (i) Our SSG model captures the inspection process and accounts for two types of inspections: thorough inspections and simple (i.e., quick) inspections. While thorough inspections take longer to conduct (and thus less of them can be conducted), they are more likely to detect violations than simple, surface-level inspections which may only be able to check for obvious violations. To model the imperfect nature of these inspections, we (ii) introduce two failure rates: one for thorough inspections and one for simple inspections, with simple inspections failing at a higher rate. (iii) We also address the uncertainty involved with road networks by using a Markov Decision Process (MDP) that will represent and ultimately generate the game solution. In addition, (iv) we also investigate how tannery compliance is affected by two fine structures: fixed fines and variable fines, where the latter will result in larger tanneries receiving larger fines. Finally, (v) we introduce the explanation component framework and demonstrate how it can be applied to explaining NECTAR's solutions. For the evaluation of our model, we apply NECTAR to a real-world network of tanneries in Kanpur, India, and we evaluate the quality of NECTAR's generated solutions. We also piloted a survey among the study team and affiliates in order to receive initial feedback on the explanation component such that we can further refine our explanations and conduct full-scale human subject experiments. We also demonstrate how NECTAR's solutions can be visualized via a Google Earth overlay that we anticipate will improve ease of use and, ultimately, odds of user adoption.

2. RELATED WORK

Several theoretical papers have used game theory to model the impact of environmental policies. Environmental games [19] use Stackelberg Games to model interactions between a regulator and a polluting firm, while [7] used game theory to study the effect of environmental policies in the Chinese electroplating industry. *Inspection games* consider the general problem of scheduling inspections, and have been extensively studied in the literature. For example, [8] models cases where an inspector must travel to multiple sites and determine violations as a stochastic game. A general theory of inspection games for problems such as arms control and environmental policy enforcement has been studied in [2], including analysis of whether inspectors can benefit from acting first. [21] also considered inspection games with sequential inspections, including compact recursive descriptions of these games. However, most of these works do not focus on concrete applications and thus, unlike our work, do not provide executable inspection schedules to inspectors.

Other areas of research have considered various models of patrolling strategies and scheduling constraints. These include patrolling games [1, 5, 3] and security games with varying forms of scheduling constraints on resources [23, 13, 6]. There have also been recent work on utilizing MDPs to represent strategies in security games [17, 4]. However, none of these efforts have focused on environmental inspections and have not investigated topics important in this domain, such as the impact of fine structures on adversary behavior, i.e., compliance, or the explanation of solutions to users that are non-experts in game theory and optimization.

As previously noted, it is important for users to be able to explore the plans and understand the system's rationale such that they gain trust in the system and adopt its solutions. The CoastWatch system helps the user define and solve a dynamic search and rescue problem and includes a visualization tool which creates an animation of the planning and scheduling problem in Google Earth [10]. [22] captures a human-agent interaction scenario in an agent-based simulator, PsychSim, where humans and agents work together to cooperatively solve a problem. The work studies how agents' communications of their current belief state can improve the team's performance and build trust between humans and agents. Similarly, we recognize the importance of system communication in building trust and fostering user adoption. As such, for the first time in security games, we explore this problem and present a novel framework to explain game theoretic solutions in an accessible manner.

3. MOTIVATING DOMAIN

The pollution of India's rivers is a major environmental concern. The waters of India's largest river, the Ganga (or Ganges) River, are used by over 400 million people – roughly one-third of India's population. Unfortunately, the Ganga is ranked the fifth dirtiest river in the world [16]. Pollution, including untreated sewage and industrial effluents, inflicts serious health consequences on all life that depends on the river. In Kanpur, villagers suffer from conditions including cholera and miscarriages, while livestock yield less milk and occasionally die suddenly [9].

Situated around the city of Kanpur, the various leather tanneries are a major source of pollution in the Ganga river [9]. While there are a few sewage treatment plants (STPs) in Kanpur, they can neither treat the full volume nor the full range of produced pollutants [11]. In particular, treating heavy metals like chromium, mercury, arsenic, and nickel is costly and needs specialized personnel (in addition to the personnel required to operate the STPs). The government has put in regulations requiring the tanneries to own and operate effluent plants to remove the pollutants before discharging sewage. However, the tanneries have not been willing to undertake the additional cost of installing and operat-

water is supposed to be NECTAR (or Amrit, the Hindi antonym of poison) which has inspired our project. The project name is intentionally chosen to fit this international and inter-cultural theme.

ing the treatment units. Even when tanneries have installed the units, they avoid operating them whenever possible.

To address non-compliance issues, the government sends inspection teams to visit the tanneries. Inspecting the tanneries is a time-consuming, quasi-legal activity where the "as-is" situation is carefully recorded and samples are collected that can later be subjected to judicial scrutiny. It is also costly because, apart from the inspectors themselves, help from local police is requisitioned for safety, lab work is done for sample testing, and movement logistics are carefully planned; a full inspection is costly to conduct. Due to these costs, the number of inspectors that can be sent out on a patrol is very limited. Our application seeks to help with this difficulty by (1) generating randomized inspection patrols that maximize the effectiveness of available inspectors, and (2) introducing limited inspection teams which conduct simple inspections, a low-cost alternative to full inspection teams which conduct thorough inspections. While limited inspection teams cannot replace the needed capabilities of a full inspection team, they can still inspect tanneries and issue a fine for obvious violations (e.g., the site not owning an STP). Henceforth, we will refer to full inspection teams and limited inspection teams as thorough inspection resources and simple inspection resources, respectively.

4. MODEL

In this section, we model this pollution prevention problem as a defender-attacker Stackelberg Security Game (SSG). The task of the defender is to send resources to different tannery sites (i.e., the multiple adversaries) on a road network. The defender must devise a patrol strategy to maximize compliance among a number of sites (each site denoted by l), where each site has a number of factories f_l and each site's compliance cost increases with the number of factories. In addition, the defender must take into account the time it takes to travel to and inspect each site. We model the road network as a graph where the nodes represent sites and the edges represent the roads connecting each site. Each edge also has a cost, e_{ab}, associated with it that represents the travel time from a site a to another site b. Using publicly available data regarding tannery locations in Kanpur, we constructed a graph consisting of 50 sites.

The defender has two types of resources: r_1 number of thorough inspection resources and r_2 simple inspection resources. For thorough inspection resources, the inspector conducts a detailed inspection that takes i time units. We model imperfect inspections such that even if a violation exists, the inspectors will fail to detect it with a low probability γ_1. For simple inspection resources, the inspector will conduct a superficial inspection of the site that takes d time units. Since it is not a detailed inspection, simple inspection resources will not be able to detect anything but obvious violations. Thus, such resources have a higher probability of failure γ_2. Each of the defender's resources (thorough and simple) have a maximum time budget, t_1 and t_2 respectively, to conduct inspections and travel to sites.

In the SSG framework, the defender will commit to a randomized patrol strategy (a mixed strategy) which is a probability distribution over the executable daily inspection patrols (the pure strategies for all resources). The adversaries (the sites) can fully observe the defender's mixed strategy and know the probability of being inspected by a thorough inspection team or a simple inspection team on a given day. Formulating the mixed strategy requires enumerating all feasible pure strategies for the defender. However, this approach is impractical for two main reasons: (1) for any realistically-sized patrolling problem, the defender pure strategy space is so large that it cannot fit into memory. For example, with our Kanpur graph which includes 50 tanneries, only one defender resource, and a time horizon of 10 hours, the pure strategy space size would be too large to enumerate (approximately 50 choose 10). Therefore, we adopt a compact representation (a transition graph) that will allow our approach to scale to large problem sizes. (2) Inspectors must travel to tanneries sites via a road network (with potential delays), and the corresponding uncertainty cannot be handled by a standard SSG formulation. Rather than reasoning about mixed strategies, we instead use the compact representation to reason about spatio-temporal flow through a transition graph. To account for stochasticity and uncertainty in the outcome of actions, we use a Markov Decision Process (MDP) to represent the defender's inspection patrolling problem. We can solve the corresponding linear program (LP) to compute the optimal inspection strategy, i.e., the optimal MDP policy.

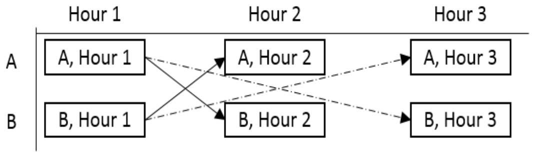


Figure 1: Illustrative MDP Example

4.1 Compact Game Representation: Transition Graph

[6] also faced the challenge of large state spaces for a traffic enforcement domain. Since their game also takes place on a road network, there are sufficient similarities between our approach and theirs to apply their techniques, based on transition graphs, to improve the scalability of our model.

Instead of enumerating an exponential number of pure strategies, we need only enumerate a polynomial number of states and edges in the transition graph. We then compute the optimal probability flow (as seen in the next section), also called a marginal coverage vector, and sample from the vector to create inspection schedules. As the defender resource types (thorough inspection resources and simple inspection resources) have different time constraints, each has its own transition graph.

We discretize time into a granularity of h hours. In the thorough inspection resource transition graph, a vertex is added for each site l every h hours until the thorough inspection resource time budget t_1 has been expended. Similarly for the simple inspection resource transition graph, a vertex is added until the time budget t_2 has been expended.

4.2 MDP Formulation

We present an MDP ($\langle S, A, T, R\rangle$) to incorporate uncertainty into the transition graph. An example MDP is shown in Figure 1 to illustrate the following definitions.

- S: Finite set of states. Each state $s \in S$ is a tuple (l, τ), where l is the site that the resource is located, and τ is the current time step. For example, an inspector at site A at hour 1 is represented as $s_{A,1}$. Each vertex in the transition graph corresponds to a state s.

- A: Finite set of actions. $A(s)$ corresponds to the set of actions available from state s, i.e., the set of sites reachable from l, that the resource can travel to and inspect. For example, at site A at hour 1, the only available action is to move to site B (i.e., the solid arrow from A to B in Figure 1).

- $T_1(s, a, s')$: Probability of an inspector ending up in state s' after performing action a while in state s. Travel time and inspection time are both represented here. As a simple example, there could be probability 0.7 for transition $T_1(s_{A,1}, a_B, s_{B,2})$: a transition from site A at hour 1 to move to and inspect site B will, with a probability of 0.7, finish at hour 2 (a travel + inspection time of 1 hour). The dashed lines in Figure 1 represent the remaining probability (0.3) that the same action will instead finish at hour 3 (due to a delay). Note that the two resource types have separate transition functions due to the difference in action times (i for thorough inspection resources and d for simple inspection resources).

- $R(s, a, s')$: The reward function for ending in state s' after performing action a while in state s. As we are interested in the game-theoretic reward, we define the reward in the LP and define R $= 0\ \forall s, a, s'$.

5. INSPECTION PATROL GENERATION

Following the transition graph and MDP, we provide a linear program (LP) to compute the optimal flow through the transition graph. By normalizing the outgoing flow from each state in the MDP, we obtain the optimal MDP policy from which we can sample to generate dynamic patrol schedules.

5.1 LP Formulation

In the following LP formulation, we make use of the following notation. A site l has a number of factories f_l, and if a site is caught violating by an inspection, they are penalized with a fine, α_l. On the other hand, if a site wants to remain in compliance, they will need to pay a compliance cost β for each factory (total cost $= \beta f_l$). We represent the expected cost for each site l as v_l. As defined in the following LP, the expected cost corresponds to the lowest of either the expected fine a site will pay or the full cost of compliance; given that we are dealing with rational adversaries, each site will choose to pay the lowest of those two expected costs (either the expected fine or the cost of compliance). Finally, we denote as S_l the set of all states that correspond to site l (i.e., all time steps associated with site l).

As discussed in the transition graph definition, the optimal flow through the graph corresponds to the optimal defender strategy, and that flow is represented by a marginal coverage vector. We denote the marginal probability of a resource type i (either thorough or simple inspection team) reaching state s and executing action a as $w_i(s, a)$. We also denote, as $x_i(s, a, s')$, the marginal probability of a resource type i reaching state s, executing action a, and ending in state s'.

$$\max_{w,x} \sum_l v_l \tag{1}$$

$$s.t. x_i(s, a, s') = w_i(s, a) T_i(s, a, s'), \forall s, a, s', i \tag{2}$$

$$\sum_{s',a',i} x_i(s', a', s) = \sum_{a,i} w_i(s, a), \forall s, i \tag{3}$$

$$\sum_{a,i} w_i(s_i^+, a) = r_i \tag{4}$$

$$\sum_{s,a,i} x_i(s, a, s_i^-) = r_i \tag{5}$$

$$w_i(s, a) \geq 0 \tag{6}$$

$$v_l \leq \alpha_l(p_{l1} + p_{l2}) \tag{7}$$

$$p_{l1} = (1 - \gamma_1) \sum_{s \in S_l, a} w_1(s, a) \tag{8}$$

$$p_{l2} = (1 - \gamma_2) \sum_{s \in S_l, a} w_2(s, a) \tag{9}$$

$$p_{l1} + p_{l2} \leq 1 \tag{10}$$

$$0 \leq v_l \leq \beta f_l \tag{11}$$

The objective function in Equation 1 maximizes the total expected cost over all sites. Constraints 2-5 detail the transition graph flow constraints (for thorough inspections and simple inspections). Constraint 2 defines that x is equal to the probability of reaching a state s and performing action a multiplied by the probability of successfully transitioning to state s'. Constraint 3 ensures that the flow into a state s is equal to the flow out of the state. Constraints 4-5 enforce that the total flow in the transition graph, corresponding to the number of defender resources r_i, is held constant for both the flow out of the dummy source nodes s_i^+ and into the dummy sink nodes s_i^-.

Constraint 7 constrains the expected cost for site l. Constraints 8-9 define the probability of successfully inspecting a given site l and is the summation of probabilities of reaching any of l's corresponding states (thus triggering an inspection) and taking any action a. Note that the failure probability γ means that even if a violating site is inspected, there may not be a fine issued. Constraint 10 limits the overall probability of a site being inspected. If a site is visited by both thorough and simple inspection resources, the site will only have to pay a fine, at most, once. Constraint 11 defines the bounds for the adversary's expected cost; if the adversary's expected cost is at the upper bound ($v_l = \beta f_l$), we assume that the adversary would prefer to have a positive public perception and choose to comply rather than pay an equivalent amount in expected fines.

6. EXPLAINING NECTAR SOLUTIONS

For NECTAR to be adopted as an inspection planning tool, the end users must have a high degree of confidence that the solutions computed by the system are feasible and efficient patrolling strategies. In work on the Technology Acceptance Model (TAM), [20] found that user perceptions of the solution quality and ease of use significantly influenced user acceptance of four different information technology systems. In our context, the end users will likely be inspectors and managers with degrees in the physical sciences. However, it is unlikely that they will also be experts in game theory and optimization; the NECTAR system may seem to function as a black box that generates strategies for opaque reasons. To address this key challenge for adoption, we have developed an explanation module for NECTAR that is designed to make the solutions more transparent to the users, ultimately building trust in the system.

6.1 Simplifying Explanations

The main challenge in explaining the solutions to users is that the optimal policy is very complex: it is the solution to an MDP that specifies inspection probabilities for multiple locations and time steps. In addition, the optimal policy may be the result of complex tradeoffs between many different priorities and constraints. We have designed our explanations to focus on the most important aspect of the solution: *how frequently each site will be inspected.* This allows for simpler explanations that abstract away many of the details of time and real-world uncertainties that are captured in the complete NECTAR model in Section 5.1.

Our simplified model for explanation focuses on the aggregate probability that each site will be inspected: $\hat{x_l}$, which is the sum of incoming flow into the site, $\sum_{s \in S_l, a} w(s, a)$. The defender's expected utility in this case is the sum of the expected fines ($\alpha_l \hat{x_l}$) over all sites, and the optimal solution maximizes this quantity. An additional advantage of this approach is that representing the solution in terms of the coverage probabilities for a set of targets is common to many of the Stackelberg Security Games that have been presented in the literature, even though the details of the resources and scheduling constraints vary depending on the specific domain. As such, our method for generating explanations can be applied with very little modification to other existing decision support systems based on security games.

6.2 Explanation Overview

Our explanations are based on the paradigm of "what-if" analysis. We allow users to ask specific questions about potential modifications to the solution calculated by the system, such as increasing or decreasing the probability of visiting a specific location. The system generates a series of statements that describe the implications of this change and show how it leads to a worse solution overall. We show example output from NECTAR's explanation module in Figure 2 in response to a user query: "Why isn't there 10% more coverage on site L3?" The explanation component analyzes this hypothetical scenario, and at key points in its internal evaluation, outputs explanatory statements to the user.

The key ideas that must be explained to the user include (1) there are tradeoffs due to the overall resource limitations, and adding coverage in one location means removing it from another location, (2) even if we assume the best case scenario for the modification (e.g., removing coverage from the least important location), the overall solution quality does not improve, so (3) NECTAR has already generated a solution that optimally balances these tradeoffs within the limitations of the resources.

There are a limited number of different ways for the user to modify the solution, and a few general types of arguments can be used to explain why the modification does not improve solution quality. For each of these possible "what-if" scenarios, we have developed an *explanation template* that has the basic text and structure of the argument. However, the details of the argument are problem-specific so they must be generated by the system each time a user asks for an explanation. The explanation shown in Figure 2 is an example of a template that has been instantiated with these details.

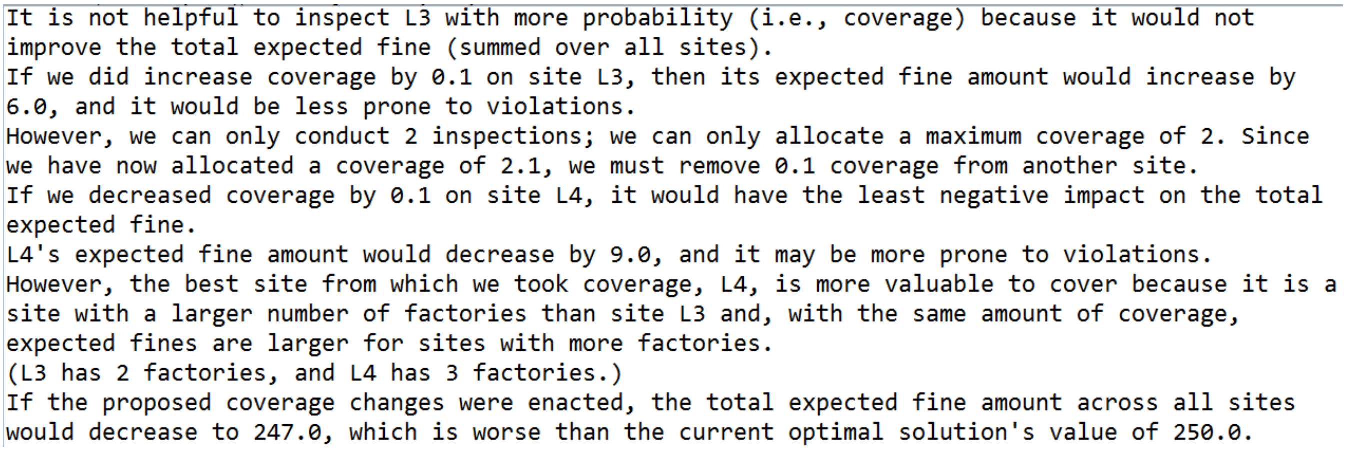
6.3 Automating Explanations

We now describe how the system automatically generates explanations for questions of the form "Why is target l covered with $\hat{x_l}$ probability?" There are two versions of this question for increasing or decreasing the probability, but they are very similar so we focus on the case of increasing the coverage on l. Consider the scenario of allocating Δ more coverage to some $l \in L$, which is currently assigned coverage $\hat{x_l}$. The system makes this change to generate the modified coverage distribution $\hat{x'}$. However, this coverage change may violate the constraint that the system cannot change the overall number of resources; the sum of the coverage in $\hat{x'}$ should be the same as in $\hat{x}$. The system checks for any violations of these constraints and then attempts to "repair" the solution in the way that is best for the defender. Based on the outcome of this repair operation, the system presents a final explanation comparing the outcomes of the original solution and the modified one to demonstrate that the modification does not result in an improvement for the defender.

The details for how the explanation system repairs violations in coverage overallocation are shown in Algorithm 1. Note that the notation *explain* refers to filling in a template explanation with specific details as needed. Here the system needs to both repair the solution and explain to the user why the violations is resolved in this way. It is important that the repaired solution represents the best case for the defender in order for it to be convincing to the user. For example, if the modified solution requires too many resources (e.g., as a result of adding coverage to a location), the way to resolve this is to remove coverage from another target ($l' \in L, l' \neq l$). Logically, this coverage should be removed from the least-harmful target and not a more valuable target. Our system systematically considers each target and picks the one where reducing the coverage is least harmful to the defender. This rationale is communicated to the user in the explanation.

Algorithm 1 Explanation System: Resolve Target Coverage Overallocation

```
function RESOLVE-OVERALLOCATION($l, \hat{x'}, L$)
    $\Delta' \leftarrow$ ComputeOverallocation;
    $EU_d^* \leftarrow -\infty$;
    $l^* \leftarrow null$;
    for each $l' \in L, l' \neq l$ do
        Reduce coverage on $l'$ by $\Delta'$;
        Compute adversary best response;
        Compute $EU_d^{\hat{x''}}$ given adversary best response;
        if $EU_d^{\hat{x''}} > EU_d^*$ then
            $EU_d^* \leftarrow EU_d^{\hat{x''}}$;
            $l^* \leftarrow l'$;
        end if
        Revert coverage on $l'$;
    end for
    explain Coverage on $l^*$ could be reduced with the least harm;
    Reduce coverage on $l^*$ by $\Delta'$;
    explain State changes in attacker response;
```



```
It is not helpful to inspect L3 with more probability (i.e., coverage) because it would not
improve the total expected fine (summed over all sites).
If we did increase coverage by 0.1 on site L3, then its expected fine amount would increase by
6.0, and it would be less prone to violations.
However, we can only conduct 2 inspections; we can only allocate a maximum coverage of 2. Since
we have now allocated a coverage of 2.1, we must remove 0.1 coverage from another site.
If we decreased coverage by 0.1 on site L4, it would have the least negative impact on the total
expected fine.
L4's expected fine amount would decrease by 9.0, and it may be more prone to violations.
However, the best site from which we took coverage, L4, is more valuable to cover because it is a
site with a larger number of factories than site L3 and, with the same amount of coverage,
expected fines are larger for sites with more factories.
(L3 has 2 factories, and L4 has 3 factories.)
If the proposed coverage changes were enacted, the total expected fine amount across all sites
would decrease to 247.0, which is worse than the current optimal solution's value of 250.0.
```

Figure 2: Example output from NECTAR's explanation component.

The system first computes the amount of coverage that is overallocated, Δ', that must be removed from another target $l'(l' \neq l)$. The impact on the defender's expected utility for removing this amount of coverage is assessed for each target. This is done by temporarily reducing the coverage, generating the new coverage distribution $\hat{x}''$, computing the adversary's best response to this coverage, and calculating the expected utility for the defender in this case. In our domain, this corresponds to computing the change in expected fine ($\alpha_{l'}\hat{x_{l'}}$) for each target l'. Once the best case target is found, the explanation is given to the user for why decreasing coverage on l^* is the best case. Finally, the system explains how the attacker's best response changes in this best-case scenario.

6.4 NECTAR Visualization

Since our goal is to assist inspection agencies with patrol planning, in addition to solution explanations, it is useful to visualize the proposed inspection patrols. In Figure 3a, we show a simple graph and strategy visualization in Google Earth (a visualization for the Kanpur area is shown in Figure 3b). The lines represent edges on the graph (i.e., straight line connections between sites). Each line also has a time step and a coverage probability associated with it, where the probability represents the value of the MDP's transition function, $T(s, a, s')$. In other words, this answers the question: "If the defender resource starts at the site at the beginning of this edge at this time step (i.e., state s), what is the probability that the defender resource will take action a and arrive at the site at the end of this edge in a following time step (i.e., state s')?" By clicking on an edge, the user can call up the information shown in Figure 3a.

Future work will integrate this Google Earth visualization component with the explanation component, allowing the user to access both of these subsystems from one application. In addition, more complex visualizations will aim to explain more details of the model, such as the temporal component of inspections, in an accessible manner. For example, a colored coverage heatmap may give the user a quick summary of the coverage distribution and animations could show how this coverage changes over time. The goal of these features will be to improve ease of use and solution understandability and thus further encourage user adoption.

7. NECTAR EVALUATION

In order to explore the strategic tradeoffs that exist in our model of the tannery domain, we ran a series of experiments on our Kanpur tannery graph. For each experiment, we generated 3 distinct patrolling strategy types. 1. a NECTAR strategy, 2. a Uniform Random (UR) strategy: at each time step, every site has an equal probability of being chosen, and 3. an Ad-Hoc (AH) strategy: a deterministic strategy where sites are visited in numerical order (by ID number).

In order to analyze how the different resource types affect performance, for each experiment we generated a total of six defender strategies: the first three (NECTAR, UR, AH) correspond to when the defender has twice as many simple inspection resources as thorough inspection resources, and the last three (again NECTAR, UR, AH) correspond to when the defender had no simple inspection resources.

In addition to running experiments where each site l has the same fine (α), we also ran an additional set of experiments where each site's fine α_l was: $\alpha_l = \alpha f_l$ or, in other words, the fine amount is a constant α multiplied by the number of factories f_l at that site. This ensures that violations at sites with more factories will be penalized more harshly than violations at small sites (with fewer polluting factories). As this type of analysis requires heterogeneous sites, we randomize the number of factories at each site.

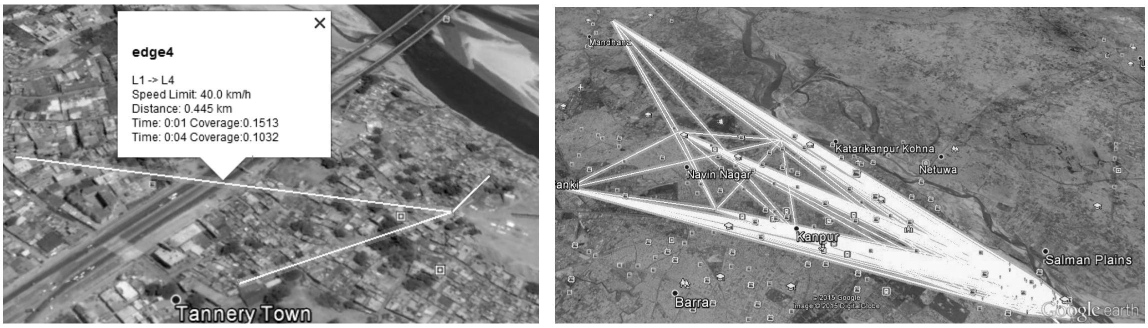
Ultimately, we are interested in inducing compliance of the sites, and for our performance metric, we compute the number of sites that would be in full compliance given the defender strategy (i.e., how many sites have an expected cost $v_l = \beta f_l$). The maximum number of sites in compliance for each experiment is 50 (i.e., the number of sites on our graph). The default parameter values for each experiment (unless otherwise specified) are listed in Table 1.

7.1 Analysis of Compliance Trade-offs

7.1.1 *Fixed Fine Amount*

In Figure 4, we analyze the effects of the fixed fine amount α on the number of complying sites. The x-axis shows the fixed fine amount, and the y-axis shows the number of sites that are in full compliance (i.e., $v_l = \beta f_l$).

From the figure, we observe the following trends: (1) the NECTAR strategy does not achieve any compliance until



(a) Visualization example

(b) A Kanpur inspection patrol plan

Figure 3: Google Earth Visualizations of NECTAR Output

Table 1: Default Experiment Values

Variable	Value
Compliance Cost β	10
Fixed Fine Amount α	100
Number of Factories at Each Site f_l	2-5
Number of Simple Inspection Resources r_2	2
Number of Thorough Inspection Resources r_1	1
Number of Sites	50
Patrol duration (hours) t_1, t_2	6
Simple Inspection Failure Rate γ_2	0.6
Thorough Inspection Failure Rate γ_1	0.1
Time granularity (hours) h	1
Time steps to complete simple inspection	1
Time steps to complete thorough inspection	2
Variable Fine Amount α_l	30

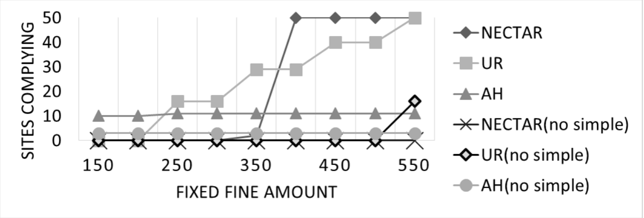


Figure 4: Fixed Fine: Number of Sites in Compliance

the fine amount is 350, with all sites in compliance at 400. This is due to the objective function attempting to maximize expected cost over all sites simultaneously with a homogeneous fine. (2) While the UR and AH strategies achieve compliance from some of the sites for smaller fine amounts, they do not achieve compliance for all of the sites as quickly as the NECTAR strategy. (3) The inclusion of simple inspection resources improve performance for every strategy as expected.

7.1.2 Variable Fine Amount

In Figure 5, we analyze the effects of the variable fine amount α_l on the number of complying sites. The x-axis shows the variable fine amount, and the y-axis shows the number of sites that are complying.

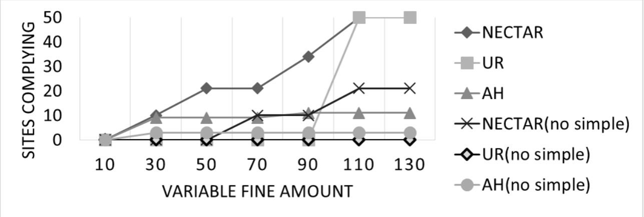


Figure 5: Variable Fine: Number of Sites in Compliance

From the figure, we observe the following trends: (1) both the NECTAR and UR strategies achieve compliance from all sites for the same variable fine amount; (2) as the fines are not homogeneous for all sites, it is beneficial for NECTAR to try to maximize expected cost in sites with many factories first (unlike with the fixed fine, there is no "water filling" effect); the NECTAR approach achieves faster compliance from larger sites, and (3) the NECTAR achieves compliance from the most sites at every point.

7.1.3 Number of Resources: Variable Fine

In Figure 6, we analyze the effect of the number of resources when there is a variable fine amount α_l on the number of complying sites. The x-axis shows the number of thorough inspection resources, r_1 (for the strategies with simple inspection resources, the number of simple inspection resources is $r_2 = 2 \times r_1$), and the y-axis shows the number of sites that are complying (i.e., $v_l = \beta f_l$).

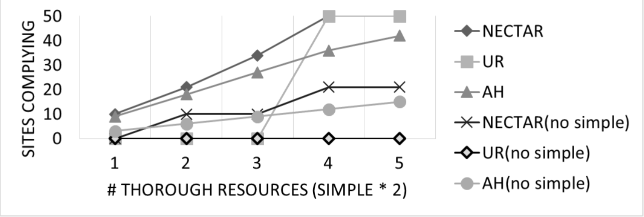


Figure 6: Number of Resources: Variable Fine: Number of Sites in Compliance

From the figure, we observe the following trends: (1) the NECTAR and AH strategies achieve compliance from some sites even with few thorough inspection resources, but NECTAR achieves compliance from the most sites at every point, (2) both the NECTAR and UR strategies achieve compliance from all sites for the same number of thorough inspection resources, and (3) even when there are many resources, the AH strategy does not achieve compliance from all sites.

7.1.4 Patrol Duration: Variable Fine

In Figure 7, we analyze the effects of the patrol duration on compliance when there is a variable fine amount α_l. The x-axis shows the patrol duration, and the y-axis shows the number of sites that are complying (i.e., $v_l = \beta f_l$).

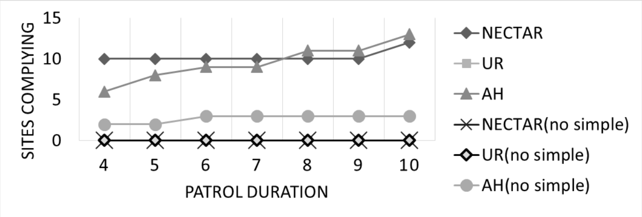


Figure 7: Patrol Duration: Variable Fine: Number of Sites in Compliance

From the figure, we observe the following trends: (1) while the NECTAR strategy performs the best for lower values of patrol duration, it is eventually outpaced by the AH strategy, (2) regardless of the strategy, there is not much change in the number of sites in compliance as a function of patrol duration. For this experiment, the default values for the other parameters result in a low compliance rate regardless of the value of the variable of interest, and (3) having simple inspection resources is helpful for the NECTAR and AH strategies, but it is not very helpful for the UR strategy.

7.2 Experiment Discussion

Based on these simulations, we make the following conclusions: (1) when the number of resources or variable fine amount is the experiment variable, NECTAR makes the most efficient use of its resources, regardless of whether it is using only thorough inspections or a combination of simple and thorough inspections; (2) having more resources (more manpower) is more useful than increasing the duration of patrols (longer work hours). This is intuitive when considering that each resource must spend time traveling to each site; two resources can each cover a separate sub-section of the graph whereas one resource will be forced to spend more time traveling. Finally, (3) using a variable fine (in which sites are fined according to their number of factories) leads to better compliance rates. This observation makes sense when put in the context of our LP's objective function: maximize the sum of the expected costs v_l over all sites.

8. EXPLANATION PILOT SURVEY

With the comparative explanation component still in its infancy, we piloted a survey among our affiliates. The goal was to acquire a baseline measurement of how explanations could increase trust in security game decision aids such as NECTAR. In order to refine our methodology for future, full-scale human subject experiments, we also wanted to receive feedback on explanations of varying verbosity and on the survey itself. Pilot respondents were randomly assigned to complete one of three different survey versions, where each version contained explanations of a single verbosity (i.e., level of detail) type: low, medium, or high.

In the survey, we presented a simplified NECTAR scenario consisting of the simplified model (as presented in section 6.1), a sample problem, and an optimal coverage strategy generated by the NECTAR decision aid. Before any sample explanations were presented, a baseline questionnaire assessed the respondent's level of trust, perceived ease of use, and understanding of the solution. Next, we presented two sets of sample question (e.g., "Why isn't there more coverage on site L4'?"), explanation (e.g., Figure 2), and post-explanation questionnaire. Responses were provided on a 5-point likert scale ranging from 1="Strongly disagree" to 5="Strongly agree". As a result of the ordering of these measurements, we would expect increases in the respondent's level of trust to be a result of the explanations. At the end of the survey, we also presented a set of open-ended questions to elicit more detailed feedback.

For this analysis, we evaluated changes in trust as a function of explanation via the following pair of questions: "I trust the decision aid to make the best decisions." and "In the future, if there were explanations provided, I would trust the decision aid to make the best decisions." Out of the 12 respondents, 7 (2 in the low verbosity group, 2 in the medium verbosity group, and 3 in the high verbosity group) expressed an increase in trust in the decision aid, 4 (1 in the low verbosity group, 2 in the medium verbosity group, and 1 in the high verbosity group) already trusted the decision aid and did not express an increase in trust, and only 1 (in the low verbosity group) expressed neither trust nor distrust in the decision aid before and after the explanations.

In the open-ended question section, 75% of respondents in the low verbosity group and 50% in the medium group indicated that more quantitative information would be even more convincing of the solution's optimality. As such, future experiments, focusing on improving user understanding and acceptance, will test explanations containing more quantitative information in an effort to identify the optimal balance between verbosity and cognitive load.

9. CONCLUSION

In this paper, we introduced a new game-theoretic application, NECTAR, which aims to aid inspection agencies in scheduling inspections of tanneries along vital rivers and waterways. NECTAR provides (1) randomized inspection policies and schedules that incorporate various real-world uncertainties and constraints, and (2) explanations and visualizations in the hopes of improving users' perceptions of solution quality and ease of use to support user adoption.

NECTAR has been proposed to decision makers in governments, pollution control boards, and funding agencies that cover cleaning of large river basins. While field inspectors have not used randomized inspection schemes in the past, they have given positive feedback on this approach, and we anticipate that by allowing them to ask "what-if" questions via the explanation component and by visualizing patrols, they will be more likely to understand NECTAR's solutions and will thus be more likely to adopt the NECTAR approach. These proposals are still in a preliminary state, and experience from literature suggests that the success of such initiatives, potentially lasting years, will greatly depend on the collaboration of multiple stakeholders so that the tannery industry and economy can continue to grow while the urgent need to protect the environment is also satisfied.

Acknowledgments: This research was supported by MURI Grant W911NF-11-1-0332.

REFERENCES

[1] S. Alpern, A. Morton, and K. Papadaki. Patrolling games. *Operations research*, 59(5):1246–1257, 2011.

[2] R. Avenhaus, B. von Stengel, and S. Zamir. *Inspection Games*, volume 3, book section 51. 2002.

[3] N. Basilico, N. Gatti, and F. Amigoni. Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder. *Artificial Intelligence Journal*, 184–185:78–123, 2012.

[4] B. Bosansky, A. Jiang, M. Tambe, and C. Kiekintveld. Combining compact representation and incremental generation in large games with sequential strategies. In *AAAI*, 2015.

[5] B. Bošanskỳ, V. Lisỳ, M. Jakob, and M. Pěchouček. Computing time-dependent policies for patrolling games with mobile targets. In *AAMAS*, pages 989–996, 2011.

[6] M. Brown, S. Saisubramanian, P. R. Varakantham, and M. Tambe. Streets: game-theoretic traffic patrolling with exploration and exploitation. In *IAAI*, 2014.

[7] X. Dong, C. Li, J. Li, J. Wang, and W. Huang. A game-theoretic analysis of implementation of cleaner production policies in the chinese electroplating industry. *Resources, Conservation and Recycling*, 54(12):1442–1448, 2010.

[8] J. Filar et al. Player aggregation in the traveling inspector model. *Automatic Control, IEEE Transactions on*, 30(8):723–729, 1985.

[9] S. Gupta, R. Gupta, and R. Tamra. Challenges faced by leather industry in kanpur. Report, 2007.

[10] W. Haas and W. S. Havens. Generating random dynamic resource scheduling problems. In *ICAPS 2008 Workshop on Knowledge Engineering for Planning and Scheduling*. Citeseer, 2008.

[11] P. C. o. India. Evaluation study on the function of state pollution control boards. Report, 2013.

[12] B. Institute. Top ten toxic pollution problems: Tannery operations. Report, 2011.

[13] M. Jain, V. Conitzer, and M. Tambe. Security scheduling for real-world networks. In *AAMAS*, 2013.

[14] D. Jainani. Kanpur leather industry in danger as ngt cracks whip on pollution, 2015.

[15] M. Mwinyihija. Emerging world leather trends and continental shifts on leather and leathergoods production. In *World leather congress proceedings, Rio de-janeiro, Brazil*, 2011.

[16] S. Queen. 11 most polluted rivers in the world, 2012.

[17] E. Shieh, A. X. Jiang, A. Yadav, P. Varakantham, and M. Tambe. Unleashing dec-mdps in security games: Enabling effective defender teamwork. In *ECAI*, 2014.

[18] M. Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, New York, NY, 2011.

[19] C. S. Tapiero. Environmental quality control and environmental games. *Environmental Modeling & Assessment*, 9(4):201–206, 2005.

[20] V. Venkatesh and F. D. Davis. A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management science*, 46(2):186–204, 2000.

[21] B. von Stengel. Recursive inspection games. *arXiv preprint arXiv:1412.0129*, 2014.

[22] N. Wang, D. V. Pynadath, K. Unnikrishnan, S. Shankar, and C. Merchant. Intelligent agents for virtual simulation of human-robot interaction. In *Virtual, Augmented and Mixed Reality*, pages 228–239. Springer, 2015.

[23] Z. Yin, A. Jiang, M. Johnson, M. Tambe, C. Kiekintveld, K. Leyton-Brown, T. Sandholm, and J. Sullivan. Trusts: Scheduling randomized patrols for fare inspection in transit systems. In *IAAI*, 2012.