Game-theoretic Patrol Strategies for Transit Systems: The TRUSTS System and its Mobile App (Demonstration)

Samantha Luber
Microsoft, WA, USA
samil@microsoft.com

Zhengyu Yin, Francesco Delle Fave, Albert Xin Jiang, Milind Tambe
University Of Southern California
Los Angeles, CA, USA
{zhengyuy,dellefav,jiangx,tambe}@usc.edu

John P. Sullivan
LA County Sheriff's Department
Los Angeles, CA, USA
jpsulliv@lasd.org

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *multiagent systems.*

Keywords: Agent-based software systems, Game theory for security, Real-world deployment

1. INTRODUCTION

Fare evasion costs proof-of-payment transit systems significant losses in revenue. In 2007 alone, the Los Angeles Metro system, using proof-of-payment, suffered an estimated revenue loss of $5.6 million due to fare evasion [2]. In addition, resource limitations prevent officers from verifying all passengers. Thus, such officers periodically inspect a subset of the passengers based on a patrol strategy. Effective patrol strategies are then needed to deter fare evasion and maximize revenue in transit systems. In addition, since potential fare evaders can exploit knowledge about the patrol strategy to avoid inspection, an unpredictable patrol strategy is needed for effectiveness. Furthermore, due to transit system complexity, human schedulers cannot manually produce randomized patrol strategies, while taking into account all of the system's scheduling constraints [3].

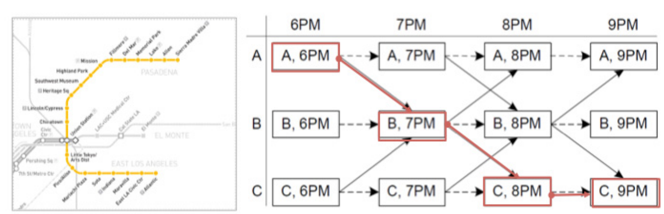
In previous work on computing game-theoretic patrol strategies, Bayesian Stackelberg games have been successfully used to model the patrolling problem. In this model, the security officer commits to a patrol strategy and the fare evaders observe this patrol strategy and select a counter strategy accordingly [4]. This approach has also been successfully deployed in real-world applications, including by the L.A. International Airport police, the U.S. Coast Guard at the Port of Boston, and the Federal Air Marshal Service [5]. However, this approach cannot be used within our setting due to the increased complexity of having more potential followers and scheduling constraints [6]. In addition, transit systems face the challenge of execution uncertainty, in which unexpected events cause patrol officers to fall off schedule and exist in unknown states in the model [1].

Addressing the increased complexity challenge, TRUSTS (Tactical Randomizations for Urban Security in Transit Systems) reduces the temporal and spatial scheduling constraints imposed by the transit system into a single transition graph, a compact representation of all possible movement throughout the transit system as flows from each station node [1]. In addition, TRUSTS remedies the execution uncertainty challenge by modeling the execution of patrol units as Markov Decision Processes (MDPs) [1]. In simulation and trial testing, the TRUSTS approach has generated effective patrol strategies for L.A. Metro System [1, 6].

In order to implement the TRUSTS approach in real-world transit systems, the METRO mobile app presented in this paper is being developed to work with TRUSTS to (i) provide officers with real-time TRUSTS-generated patrol schedules, (ii) provide recovery from schedule interruptions, and (iii) collect patrol data. An innovation in transit system patrol scheduling technology, the app works as an online agent that provides officers with the best set of patrol actions for maximizing fare evasion deterrence based on the current time and officer location. In this paper, we propose a demonstration of the TRUSTS system, composed of the TRUSTS and METRO app components, which showcases how the system works with emphasis on the mobile app for user interaction. To establish sufficient background context for the demonstration, this paper also presents a brief overview of the TRUSTS system, including the TRUSTS approach to patrol strategy generation in Section 2.1 and discussion of the METRO app's features and user interface design in Section 2.2, and the expected benefits from deployment in the L.A. Metro System.

2. THE TRUSTS SYSTEM

The TRUSTS system is composed of the TRUSTS and interface components, which will be discussed in detail in this section.



(a) Gold Metro Rail Line **(b) Officer Patrol Schedule**

Figure 1: TRUSTS-generated schedule for L.A. Metro System

2.1 TRUSTS Patrol Strategy Generation

The TRUSTS component generates transit system patrol strategies that maximize revenue, including ticket sales and violation fines, thereby deterring fare evasion. Described in detail in [1], TRUSTS models the L.A. Metro System patrolling domain, shown in Figure 1(a), as a Bayesian Stackelberg game, representing the patrol officer as the leader and the potential fare evaders as the followers [3]. As previously discussed, TRUSTS represents execution uncertainty in patrol schedules using Markov Decision Processes in order to provide contingency plans when the officer's patrol schedule is interrupted [1]. Robust against missed patrol

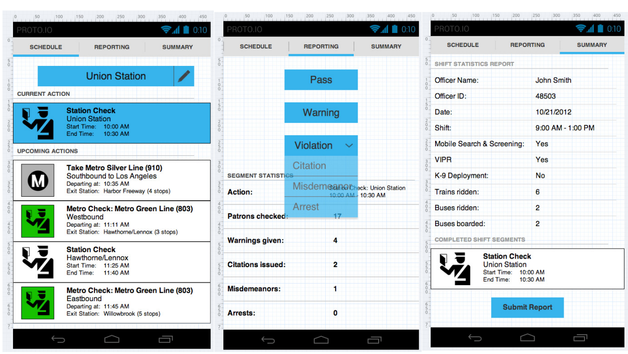
Appears in*: Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013),* Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.
Copyright © 2013, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

actions, the TRUSTS-computed solution is a Markov strategy, a stochastic policy that maps each patrol officer state to a probability distribution over the next patrol action to perform [1]. Based on this pre-computed strategy, the METRO app uses the officer's current state to determine the next approximately optimal patrol action for the officer to take.

The TRUSTS component runs on a server machine and uses the specified transit system parameters, including route schedules, shift start times, and number of patrol officers, to compute patrol strategies for each patrol officer's shift. These patrol strategies are stored in the TRUSTS database for retrieval by the METRO app. The TRUSTS database also stores patrol data collected by the METRO app, which will be discussed in detail in Section 2.2.

2.2 The METRO App

The METRO app is a software agent carried by each patrol officer that provides an interface for interaction between the officer and TRUSTS. Shown in Figure 2, the METRO app provides three principal features: a TRUSTS-generated patrol schedule for the current shift, a system for reporting passenger violations, and a shift statistics summary report. At the beginning of the shift, the METRO app queries the TRUSTS database for the officer's patrol strategy for the current shift. From the patrol strategy, the METRO app displays a schedule of the current and upcoming patrol actions in Schedule View, shown in Figure 2(a). Implementing recovery from real-world unexpected events that interrupt the officer's schedule, Schedule View also allows the officer to manually set their current location, triggering the app to compute a new patrol schedule based on the officer's current location and time. The app does so by sampling the TRUSTS-generated Markov strategy (described in Section 2.2) and calculating a new patrol schedule to display in the view [1].



(a) Schedule View (b) Reporting View (c) Summary View

Figure 2: METRO app User Interface.

The app also allows patrol officers to record passenger violations, such as fare evasion, for the current patrol action using Reporting View, shown in Figure 2(b). Officers can also view and edit the passenger violations reported for past actions in Summary View, shown in Figure 2(c). Upon shift completion, the officer can also use Summary View to submit the app-generated shift statistics summary report, including all unexpected events and violations reported throughout the shift, to the TRUSTS database.

Through analysis on this collected patrol data, we expect to gain valuable insight on the L.A. Metro patrolling domain, such as passenger behavior patterns, and the ability to evaluate the effectiveness of TRUSTS system deployment in the system. In addition, this app-collected data could also benefit transit system security departments that manually record violations data or conduct their own analysis on patrol strategy performance.

3. DEMONSTRATION

[1]Demo: http://www.youtube.com/watch?v=Qa_mZDVzjts

Our demonstration consists of three parts: an introductory video outlining the TRUSTS system, an interactive simulation of the system running on a simulated patrol shift, and a closing video on deployment in the L.A. Metro System. Engaging participants with an overview of the TRUSTS system, the introductory video presents the problem scenario, TRUSTS, and the METRO app. The interactive simulation, the focus of the demonstration, showcases the TRUSTS system running over the course of a shortened shift of a patrol officer in the L.A. Metro System. The participants experience the real-world deployment of the TRUSTS system first-hand from the perspective of the patrol officer using the METRO app. This portion of the demonstration uses a laptop to simulate the TRUSTS server, a mobile phone running the METRO app for user interaction, and a monitor (connected to the TRUSTS server) displaying what the TRUSTS system is doing at the present state in the shift simulation. The laptop runs the TRUSTS on an L.A. Metro Gold Line-based dataset, modified to support a two-minute patrol officer shift for the demonstration, to generate a patrol strategy for the simulated shift. The METRO app deployed on the mobile phone has also been modified for the demonstration to continuously communicate the METRO app's reporting data to the TRUSTS server for display on the monitor.

Throughout the shift simulation, participants are prompted by various shift event occurrences to interact with the METRO app as the patrol officer. For example, one scenario informs the user that a fare evader was caught and gives instructions on recording this violation in the Reporting View. Another scenario prompts the user with a missed action to exit at the scheduled station and to manually set their current location to trigger a re-schedule. Participants watch as the METRO app interacts with the TRUSTS and visually responds to their input. To ensure participant understanding, the simulation pauses at significant points to explain what the system is doing. Following the interactive simulation, the demonstration concludes with an overview of the TRUSTS system's success in simulation testing and the expected benefits from deploying TRUSTS in the L.A. Metro system.

4. ACKNOWLEDGMENTS

We thank the Los Angeles Sheriff's Department for their exceptional collaboration. This research is supported by TSA grant HSHQDC-10-A-BOA19.

5. REFERENCES

[1] A. Jiang, Z. Yin, C. Zhang, S. Kraus, M. Tambe Game-theoretic Randomization for Security Patrolling with Dynamic Execution Uncertainty. In *AAMAS*, 2013.

[2] Booz Allen Hamilton. Faregating analysis. Report commissioned by the LA Metro. 2007.

[3] Jiang, A., Yin, Z., Johnson, M., Tambe, M., Kiekintveld, C., Leyton-Brown, K., and Sandholm, T. Towards Optimal Patrol Strategies for Fare Inspection in Transit Systems. In *AAAI*, 2012.

[4] Paruchuri, P.; Pearce, J. P.; Marecki, J.; Tambe, M.; Ordóñez, F.; and Kraus, S. Playing games with security: An efficient exact algorithm for Bayesian Stackelberg games. In *AAMAS*, 2008.

[5] Tambe, M. Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned. 2011.

[6] Yin, Z., Jiang, A. X., Johnson, M. P., Tambe, M., Kiekintveld, C., Leyton-Brown, K., Sandholm, T., and Sullivan, J. 2012. TRUSTS: Scheduling Randomized Patrols for Fare Inspection in Transit Systems. In *AAAI*, 2012.