TALOS: A Tool for Designing Security Applications with Mobile Patrolling Robots (Demonstration)

Nicola Basilico
DEI, Politecnico di Milano, Milano, Italy
basilico@elet.polimi.it

Nicola Gatti
DEI, Politecnico di Milano, Milano, Italy
ngatti@elet.polimi.it

Pietro Testa
DEI, Politecnico di Milano, Milano, Italy
pietro.testa@mail.polimi.it

ABSTRACT

TALOS is a software tool for supporting the development of security applications with mobile patrolling robots. Exploiting TALOS's functionalities, a user can easily compose a patrolling setting and apply recent algorithms presented in the multi–agent literature to find optimal patrolling strategies. Results can be evaluated and compared with intuitive graphical representations and an interacting game can be played by the user in a simulated patrolling scenario.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents*

General Terms

Algorithms, Economics, Security

Keywords

Game theory, security, mobile robot patrolling

1. INTRODUCTION

The employment of multi–agent techniques, especially *algorithmic game theory*, for security applications has recently received a lot of attention in the scientific community. The main works deal with the security of physical locations. The most known result is [5], which focuses on the problem of protecting several locations against an attacker whose preferences are uncertain by placing static checkpoints. The setting is modeled as a two–player (a defender and an attacker) game problem. The goal is the computation of a randomized optimal strategy for the placement of the checkpoints. This result was applied to secure the Los Angeles Airport [5]. To achieve a higher level of security, the use of mobile patrolling robots has been explored in the artificial intelligence and robotic literature. The most recent theoretical results are [1] and [3]. The work in [1] deals with perimeter settings, whereas the work in [3] can be applied to settings with arbitrary topologies and with several sources of uncertainty. However, no application is currently available to support the employment of these techniques for practical settings.

In this demo we present a software tool, named TALOS (available at HTTP://HOME.DEI.POLIMI.IT/NGATTI/TALOS) that supports a user in developing effective security applications with mobile patrolling robots. More precisely, TALOS allows a user to easily define models of the environment to secure and to exploit state–of–the–art [3, 4] algorithms to compute the optimal patrolling strategy. Moreover, TALOS provides methods to evaluate the performance of optimal strategies and to conduct comparisons between different variants of a single setting. To simulate the real interaction with a human (possibly non–rational) intruder the user can play an interactive game against the optimal patroller.

Cite as: TALOS: A Tool for Designing Security Applications with Mobile Patrolling Robots (Demonstration), N. Basilico and N. Gatti and P. Testa, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1317-1318.
Copyright © 2011, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

2. MAIN FEATURES

TALOS is a web application that interacts with the user via a web browser. Web application technologies can be easily accessed by every user. The user can register to the web site obtaining an account to manage and share with other users the composed settings, results and logs. TALOS provides four main functionalities. They are described in the following.

2.1 Composing and editing patrolling settings

A patrolling setting is the set of features describing the environment to be patrolled and the robot capabilities. When dealing with realistic patrolling settings, building models that can be efficiently processed by algorithms can be a cumbersome task. TALOS provides the user with a set of graphical tools to easily compose and edit patrolling settings, hiding the low–level representations and exposing the patrolling settings in an intuitive graphical format. Following the definition of patrolling setting of [3], the user can:

- draw the environment's topology over a grid map by specifying free cells and obstacles;
- label some cells as *targets*, i.e., those locations subject to an intrusion risk and for each one of them specify a pair of *values* (one for the patroller and one for the intruder) and a *penetration time* (the time, or its probability distribution, needed by the intruder to complete an intrusion in a target);
- label some cells as *entry points*, i.e., locations from which the intruder can gain an initial access to the environment;
- specify the *range* of the detection sensor mounted on the patrolling robot (e.g., a sensor with an high range

could detect an intruder with a probability monotonically decreasing with the distance from the robot's current cell);

- specify the *game type*, i.e., if the game is strictly competitive or not; in the strictly competitive case the patroller and the intruder must share the same ordering over targets' values (this parameter influences the resolution process to be performed for the optimal patrolling strategy's computation).

Once the patrolling setting is composed, TALOS automatically checks for its consistency and warns the user in case of a non–well–formulated setting. For example, if the environment topology is not connected (and consequently the patroller cannot reach some cells) or always–winning situations for the intruder are present the user is requested to (eventually) edit the setting and remove inconsistencies. Once a well–formulated setting is completed, a low–level representation is generated to enable an efficient processing in the optimal strategy computation phase.

2.2 Optimal strategy computation

When the user submits a request to TALOS for solving a well–formulated patrolling setting, the optimal patrolling strategy is computed according to two steps. In the first one, TALOS searches for a *deterministic* patrolling strategy. This strategy is defined as a cyclical sequence of cell visits such that, when it is indefinitely repeated, every target is always patrolled within a number of turns smaller than its penetration time (if penetration times are described by probabilities distributions, lower bounds are considered). If the patroller follows this strategy, the optimal intruder's action is not to intrude any target. A deterministic strategy is therefore the optimal patrolling strategy. This problem is treated according to the techniques discussed in [2] with the addition of a temporal deadline over the execution of the algorithm (results in [2] showed that 30 s is suitable).

If a deterministic strategy does not exist, TALOS executes the second step where the optimal *non–deterministic* patrolling strategy is computed. This strategy is defined as a Markovian randomization over the next cell to patrol. The algorithms applied in this phase build a game model from the composed patrolling setting and resort to bilinear mathematical programming to determine its equilibria (see [3] for more details). Moreover, reduction techniques based on the removal of dominated actions (as shown in [2]) and game theoretical abstractions are exploited to reduce the computational burden (producing approximate solutions).

During the computation of an optimal strategy the user can continue to use the other functionalities of TALOS, e.g., designing new settings. An alert (also sent by email) will notify the user of the availability of the solution.

2.3 Strategy evaluation and comparison

Once the optimal patrolling strategy is obtained, analyses of the results can be conducted. A graphical representation of the strategy can be superimposed to the environment's grid map where colors and arrows are exploited to depict transition probabilities. An animation of the patrolling strategy can also be displayed to give some insights about its actual realization. Moreover, TALOS allows the user to assess the effectiveness of the optimal strategy, namely to obtain a quantitative evaluation of how well it will protect the setting it was computed for. To achieve this, the user can examine a number of numerical indexes. Among these, there is a table reporting the intruder's expected utilities for each possible attack action. Inspecting these values, the user can get a global assessment of the strategy's performance. For example, large values would mean that the corresponding setting is difficult to protect effectively. Conversely, small values would demonstrate a high protection level.

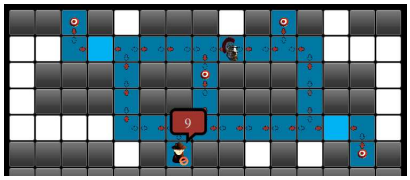

Figure 1: Interactive play screen shot.

TALOS allows a user to compare the results obtained for different variants of the same setting. In this way, the user can decide whether or not to change the setting, e.g., moving targets or changing their values if possible, spending money to strengthen targets (to extend the corresponding penetration times), or equipping the robot with better sensors. Variants can be easily composed by editing existing settings. Given two variants, a number of indexes can be compared. Among these, there is a table in which the utilities for each intruder's attack action in both settings are reported. The red color denotes (for each intruder's attack action) the setting in which the intruder's utility is the largest (which corresponds to the setting with the worst protection level). Thus, the user can graphically compare the performance of the optimal strategy in different settings, understanding how it can improve the security level by changing the setting.

2.4 Interactive play

Finally, TALOS provides an interactive scenario in which the user can play a patrolling game acting in the role of the intruder (see Fig. 1). The game is composed of a number of runs where the human player can observe the patroller executing its strategy and select a target to attack. Playing this game, the user can assess the performance of the optimal strategy against non–rational intrusion strategies (e.g., a human intruder that selects targets without considering the observed patroller's movements) and compare it with the results in the case of a rational intruder. The user can exploit this information to change the patrolling setting.

3. REFERENCES

[1] N. Agmon, S. Kraus, and G. Kaminka. Multi-robot perimeter patrol in adversarial settings. In *ICRA*, pages 2339–2345, 2008.

[2] N. Basilico, N. Gatti, and F. Amigoni. Developing a deterministic patrolling strategy for security agents. In *IAT*, pages 557–564, 2009.

[3] N. Basilico, N. Gatti, and F. Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *AAMAS*, pages 57–64, 2009.

[4] N. Basilico, N. Gatti, and F. Villa. Asynchronous multi-robot patrolling against intrusion in arbitrary topologies. In *AAAI*, pages 1224–1229, 2010.

[5] J. Pita, M. Jain, J. Marecki, F. Ordonez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus. Deployed armor protection: the application of a game theoretic model for security at the Los Angeles International Airport. In *AAMAS*, pages 125–132, 2008.