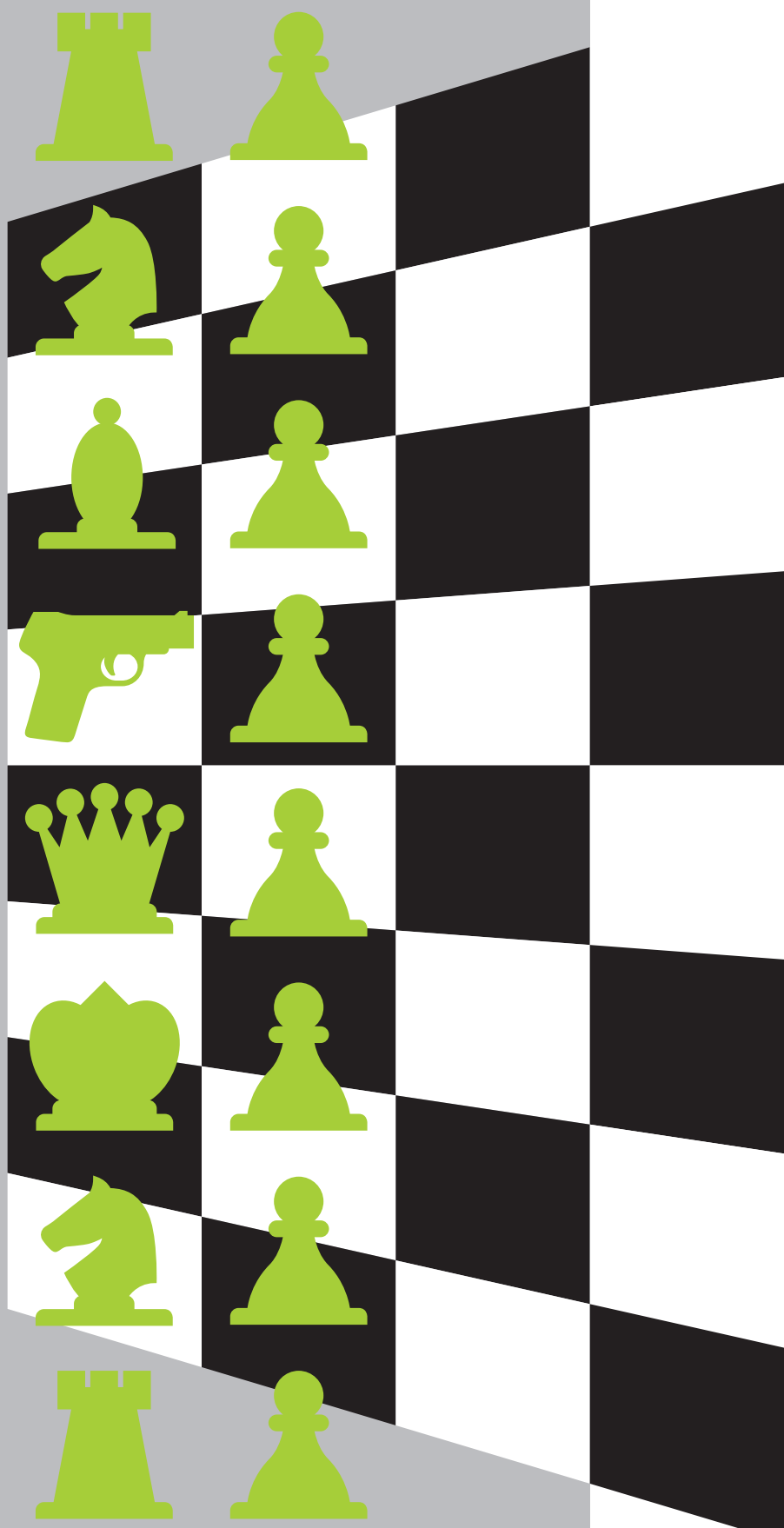
Serious games

Thanh Nguyen explains how game theory and algorithms are being used to optimise security and patrol schedules to thwart terrorist attacks



In October 1984, the British Conservative Party gathered in the seaside town of Brighton for its annual conference. Party leader and prime minister Margaret Thatcher was in attendance, as was her cabinet of government ministers – all of whom were staying in the town's aptly-named Grand Hotel.

Four weeks earlier, the hotel had welcomed Patrick Magee, a member of the Irish Republican Army (IRA), as a guest – though staff did not know it at the time. Magee was staying under the false name of Roy Walsh, and during his stay, he planted a bomb on a long-delay timer in the bathroom wall of room number 629.

At 2:54am on 12 October 1984, the timer ran out. Thatcher – the target of the bomb – was unhurt in the ensuing blast. Thirty-six others were not so fortunate: five died, and 31 were injured. In taking responsibility for the bombing, the IRA acknowledged the part luck, both good and bad, had played in the outcome of this devastating event. "Today we were unlucky," it said, "but remember we only have to be lucky once. You will have to be lucky always."

This idea – that attackers have many opportunities to achieve their goals, and need succeed only once – has echoed down the years and remains relevant to our present-day concerns regarding security. Whether it is a question of protecting public transportation and other critical national infrastructure from terrorists, or curtailing the illegal flow of weapons, drugs, and money across international borders, there are many more potential targets (or weak points) than there are resources to defend them. Defences must therefore be deployed intelligently. But how?

Game theory, which models interactions among multiple self-interested agents, is well suited to the adversarial reasoning required to solve this type of resource allocation and scheduling problem. Non-cooperative game theory, in particular, "deals largely with how intelligent individuals interact with one another in an effort to achieve their own goals", says UCLA economist David Levine (bit.ly/2cQojuR). The emergence of game theory as a field of research is often dated to 1944, with the publication of John von Neumann and Oskar Morgenstern's text, *Theory of Games and Economic Behaviour*. However, in looking to solve our modern security challenges, a team of researchers at the University of Southern California (USC) have taken inspiration from work first published in 1934 by the German economist, Heinrich von Stackelberg.

Leaders and followers

The late economist Peter Senn described Stackelberg as "one of the seminal thinkers in economics of the middle twentieth century", someone whose ideas have become "indispensable for some of mathematical economics and game theory". Yet, according to Senn, "Stackelberg's contributions were relatively unusual in that he did not develop, and could not have developed, all of his ideas to their full potential." Indeed, writes Senn, "practically all of game theory were not yet invented" during Stackelberg's time (1905–1946).

In 1934, Stackelberg published *Market Forms and Equilibrium*, which introduced the "leadership game model" in which there are *leaders*, who move first, and *followers*, who move next. Although Stackelberg's idea was formulated to address market behaviours, such as price-setting and sales strategies, the leader and follower model neatly captures the strategic interaction between security agencies and human adversaries. The security agency, as leader, develops a strategy and implements it, while the adversary, as follower, observes the implementation of the strategy before taking action.

In casting the security problem as a Stackelberg game, the Teamcore group at USC has developed new algorithms for solving such games and devising randomised patrolling or inspection strategies. These algorithms have led to successes and advances in security scheduling and allocation by addressing a key weakness in human-designed approaches: that of predictability. In what follows, we will first introduce the general security games model, before giving an example of how it is applied in a real-world setting, and how the performance of these algorithms can be assessed.

Thanh Nguyen is a postdoctoral researcher at the University of Michigan. She completed her PhD in Computer Science at the University of Southern California (USC) in 2016. While at USC, she was a member of the Teamcore Research Group

Security games

In Stackelberg security games, a defender must perpetually defend a set of targets using a limited number of resources, whereas the attacker is able to surveil and learn the defender's strategy and attack after careful planning. The goal for both defender and attacker is to maximise their *utility* – a concept used in economics and artificial intelligence

The leader and follower model neatly captures the strategic interaction between security agencies and human adversaries

to measure the importance of, or values assigned to, objects or outcomes. In this scenario, a defender maximises their utility by safeguarding their most valuable assets; an attacker maximises their utility by scoring a successful hit on a valuable target. For each target, there is a set of payoff values that define the utilities for both the defender and the attacker in case of a successful attack or failed attack.

To maximise their utility, both defenders and attackers take actions. An action, or *pure strategy*, for the defender represents deploying a set of resources on patrols or checkpoints, such as scheduling checkpoints at an airport or assigning federal air marshals to protect flight tours. The *pure strategy* for an attacker represents an attack on a target – a flight, for example. A defender can also adopt a *mixed strategy*, which is a probability distribution over the *pure strategies*.

Table 1 illustrates a simple two-target security game. In each cell, the first number is the defender's payoff while the second number is the attacker's payoff. If target 2 is successfully attacked while the defender is stationed at target 1, the defender receives a penalty of –2 while the attacker obtains a reward of 2. Conversely, if target 2 is successfully defended and the attacker is detained, the defender achieves a reward of 3 and the attacker receives a penalty of –3.

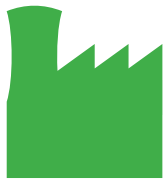


TABLE 1 An example of a two-target Stackelberg security game, showing payoff values for defender and attacker. In each cell, the first number is the defender's payoff while the second number is the attacker's payoff

		Attacker	
		Target 1	Target 2
Defender	Target 1	4, –4	–2, 2
	Target 2	–1, 1	3, –3

FIGURE 1 USS *Cole* after suicide attack (credit: US Navy, via Wikimedia Commons)

Since the attacker can observe the defender's patrolling strategy, it is critical that the defender randomise their patrols so that the attacker can no longer predict which target the defender is going to protect. The solution to a security game is, therefore, a mixed strategy for the defender that maximises the expected utility of the defender, given that the attacker learns the mixed strategy of the defender and chooses a best response. This solution concept is known as a "Stackelberg equilibrium", which can be computed by solving an optimisation problem. For example, in the two-target game in Table 1, if the attacker goes for target 1, then the defender and the attacker's expected utility, denoted by $EU^d(1)$ and $EU^a(1)$, at target 1 is as follows:

- $EU^d(1)$ = (probability of protecting target 1) × (reward for successfully catching the attacker) + (probability of not protecting target 1) × (penalty for not catching the attacker)
- $EU^a(1)$ = (probability of protecting target 1) × (penalty for being caught by the defender) + (probability of not protecting target 1) × (reward for successfully attacking the target)

Similarly, we can also compute the defender and attacker's expected utility at target 2 if the attacker chooses to attack target 2.

Now, assuming that the defender has only one guard to deploy, the optimal mixed strategy for protecting both targets is computed by searching over the solution space of all possible strategies to find the one that maximises the defender's utility. In the example in Table 1, the optimal strategy is to assign the guard to randomly protect target 1 and target 2 40% and 60% of the time, respectively. In this scenario, whichever target the attacker chooses, its expected utility is -1 and the defender's expected utility is 1. We calculate the defender's expected utility at target 1 as $EU^d(1) = 0.4 \times 4 + 0.6 \times (-1) = 1$. Any other strategy of the defender will lead to a lower expected utility for one or both of the targets. For example, if the defender protects target 1 and 2 90% and 10% of the time, respectively, the best target for the attacker is now target 2, since it will obtain the highest expected utility at this target. In this case, the defender's utility will be –1.5 (calculated as $EU^d(2) = 0.9 \times (-2) + 0.1 \times 3 = -1.5$).

Playing for real

Security games and calculations of this sort have been used by the United States Coast Guard (USCG) since 2011, to protect both passenger ferries and ports by randomising patrol strategies. For ferries, the risk is that these vessels – transporting millions of passengers each year in waterside cities such as Seattle, New York, Boston and San Francisco – are an attractive target for an attacker. For example, the attacker may ram a suicide boat packed with explosives into a ferry, mirroring the attacks carried out on the US Navy destroyer USS *Cole* (see Figure 1) and French supertanker *Limburg*.[1]

Small, fast, and well-armed patrol boats (see Figure 2) can provide protection to such ferries by detecting the attacker within a certain distance and stopping them from attacking. However, the numbers of patrol boats are often limited, so the

defender cannot protect all ferries at all times and locations. Teamcore thus developed a game-theoretic system for scheduling escort boat patrols to protect ferries, and this has been deployed on the Staten Island Ferry since 2013.

The key research challenge relates to the fact that ferries are continuously moving over a wide area, and the attacker could attack at any moment in time and location. This type of problem leads to game-theoretic models with continuous strategy spaces, which presents computational challenges. Therefore, the PROTECT-FERRY algorithm that was developed uses a compact representation of the defender's mixed strategy space while being able to exactly model the attacker's continuous strategy space.

Overall, the algorithm casts the ferry protection problem as a zero-sum security game in which targets move along a *one-dimensional* domain – i.e., a straight-line segment connecting two terminal points. This one-dimensional assumption is valid as, in ferry protection, ferries normally move back and forth in a straight line between two terminals (or ports). Although the ferries' locations will vary with respect to time changes, they do have a fixed daily schedule, meaning that, for an attacker, determining the locations of potential targets at a certain time is straightforward.

Now, the defender has patrol boats moving between two terminals to protect the ferries. While the defender is trying to protect these vessels, the attacker will decide to attack a certain target at a certain time. The probability that the attacker successfully attacks depends on the positions of the patroller at that time. Specifically, each patroller possesses a protective circle of radius within which they can detect and try to intercept any attack, whereas they are incapable of detecting the attacker outside that radius.

FIGURE 2 An escort boat protects a passenger ferry in a video shot by Teamcore researchers (see youtu.be/Zc5fp_L-gm4)

The goal for both defender and attacker is to maximise their *utility* – which measures the importance of, or values assigned to, objects or outcomes

Figure 3 (page 18) shows an example of a ferry transition graph in which each node of the graph indicates a particular location and time step for the target. (Note that while we describe PROTECT-FERRY in the discretised case, our game-theoretic solution is generalised to the continuous problem setting in reality.) Here, there are three location points, A, B, and C, on a straight line, where B lies between A and C. Initially, the target is at one of these location points at the 5-minute time step. Then the target moves to the next location point, which is determined based on the connectivity between these points at the 10-minute time step and so on. For example, if the target is at location point A at the 5-minute time step, denoted by (A, 5 min) in the transition graph, it can move to the location point B or stay at location point A at the 10-minute time step. The defender follows this transition graph to protect the target.

Real-world deployments

Teamcore's algorithms have been used in a number of different scenarios.

- **ARMOR:** the first application, deployed at the Los Angeles International Airport in 2007 to randomise checkpoints on the roadways entering the airport, and canine patrol routes within the airport terminals.
- **IRIS:** a game-theoretic scheduler for randomised deployment of the US federal air marshals has been in use since 2009.
- **PROTECT-PORT:** which schedules the US Coast Guard's randomised patrolling of ports, has been deployed in the port of Boston since April 2011 and has been in use at the port of New York since February 2012. It has spread to other ports, such as Los Angeles/Long Beach and Houston.
- **PROTECT-FERRY:** an application for deploying escort boats to protect ferries has been deployed by the US Coast Guard since April 2013.
- **TRUSTS:** has been evaluated in field trials by the Los Angeles Sheriff's Department in the LA Metro system.
- **PAWS:** another game-theoretic application, was initially tested by rangers in Uganda for protecting wildlife in Queen Elizabeth National Park in April 2014. The application was then extensively tested by rangers for protecting wildlife in a conservation area in Malaysia in July 2015.
- **MIDAS:** was tested by the US Coast Guard for protecting fisheries.

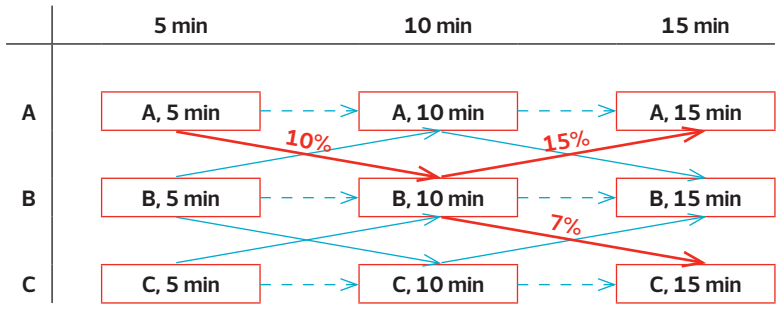


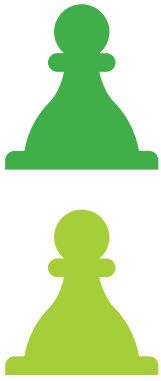
FIGURE 3 An example of a ferry transition graph

A pure strategy for the defender is defined as a trajectory of this graph; for example, the trajectory including (A, 5 min), (B, 10 min), and (C, 15 min) indicates a pure strategy for the defender, moving from point to point sequentially. The defender's mixed strategy assigns a probability to each of the patrol routes, or pure strategies, that can be executed.

One key challenge of this representation for the defender's pure strategies is that the transition graph consists of an exponential number of trajectories, N^T, where N is the number of location points and T is the number of time steps. To address this challenge, PROTECT-FERRY proposes a compact representation of the defender's mixed strategy. Instead of directly computing a probability distribution over complete trajectories of the graph for the defender, PROTECT-FERRY attempts to compute the marginal probability that the defender will follow a certain edge of the transition graph, for example the probability of being at the node (A, 5 min) and moving to the node (B, 10 min). This compact representation allows PROTECT-FERRY to reformulate the resource-allocation problem as computing the optimal *marginal* coverage of the defender over a small number of the edges of the transition graph.

Given the values of ferries and the distances they travel, PROTECT-FERRY then provides an optimal mixed strategy for the defender which can be represented as a transition probability distribution. For example, in Figure 3, the probability that the defender is at (A, 5 min) and moves to (B, 10 min) can be set to 10%, while the probability of being at (B, 10 min) and moving to (C, 15 min) can be 7%. The other transition edges will be assigned a probability in a similar fashion.

This optimal transition probability distribution is computed by searching over the solution space of all possible transition distributions to find an optimal solution that maximises the defender's utility. PROTECT-FERRY then casts these transition probabilities into detailed patrol schedules for the defender to follow.



How well does this work?

Demonstrating the effectiveness of these algorithmic solutions is important – lives are at stake, after all. However, no evidence can be provided that these algorithms provide 100% security – there is no such thing. The question to ask, then, is whether these game-theoretic algorithms are better at allocating security resources than other methods, which typically rely on human schedulers or a simple dice roll.

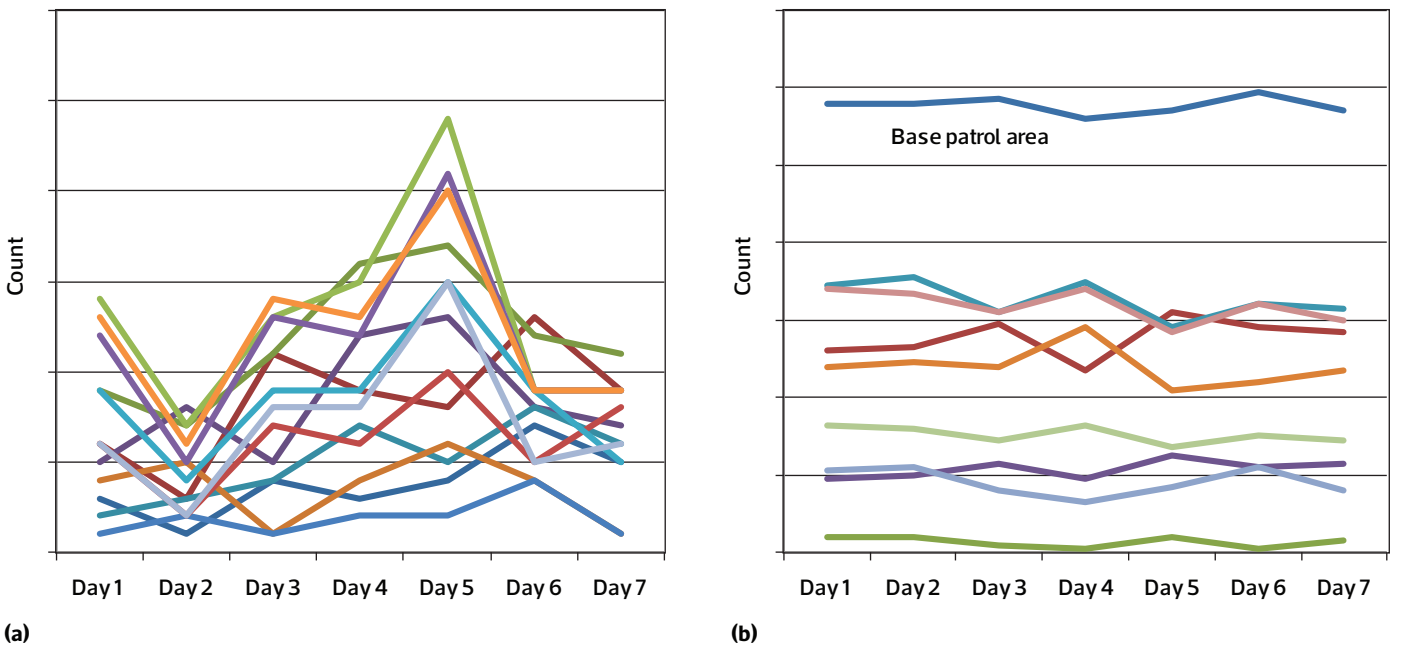


FIGURE 4 PROTECT-PORT evaluation results: (a) pre-deployment and (b) post-deployment patrols. Coloured lines represent different patrol areas, or likely targets

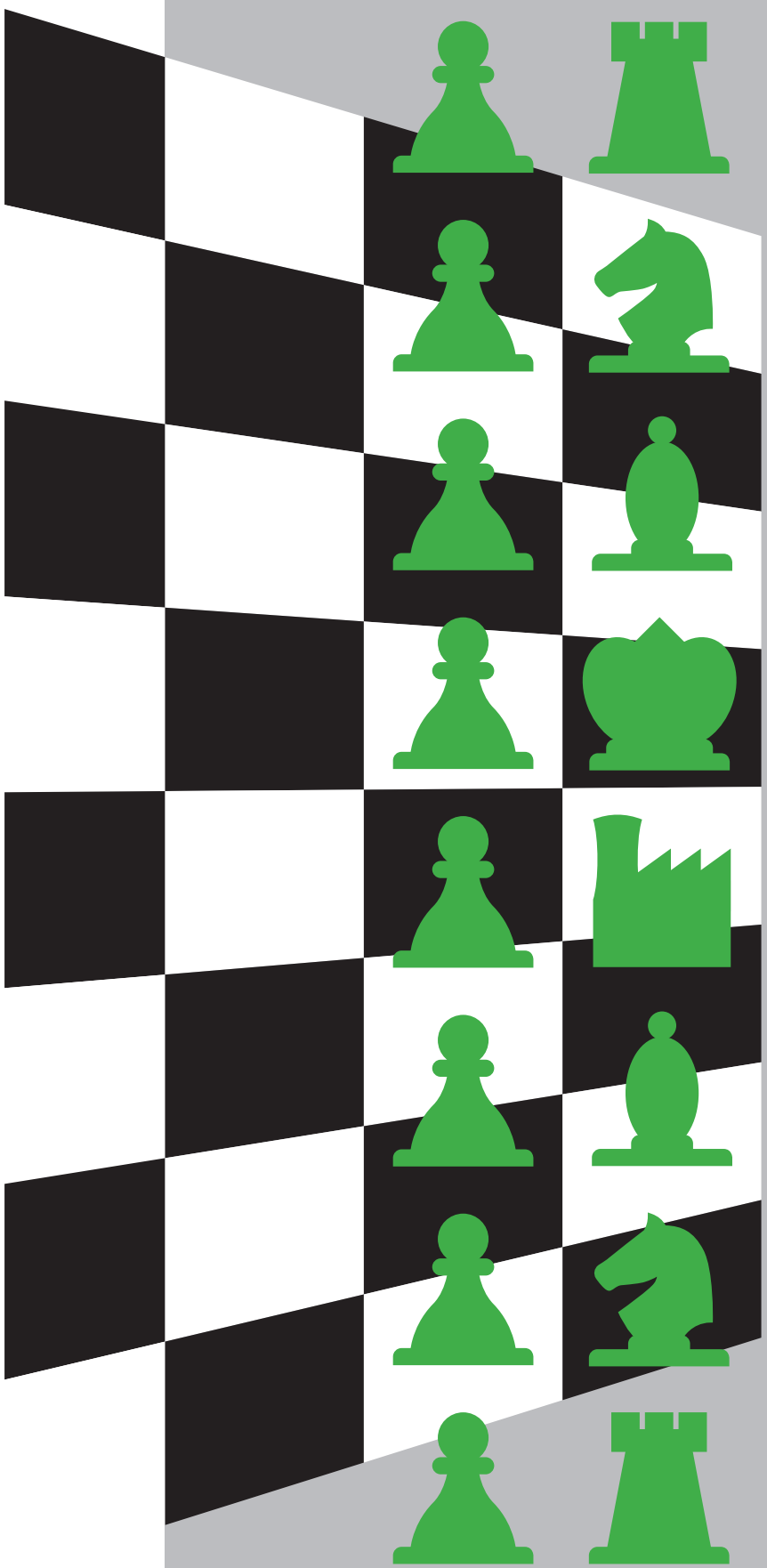
Demonstrating the effectiveness of these algorithmic solutions is important – lives are at stake, after all

For measures of interest to security agencies, such as predictability in patrol schedules, it is possible to compare the actual human-generated schedules to those designed by algorithm. Consider the following evaluation of PROTECT-PORT, another algorithm designed for the USCG, which schedules the randomised patrolling of ports. Figure 4 shows the frequency of visits by USCG to different patrol areas (represented by the different coloured lines) over a number of weeks. The *x*-axis represents the day of the week and the *y*-axis is the number of times a patrol area is visited for a given day of the week. The patrols before PROTECT, in Figure 4(a), show a definite pattern, with a spike on day 5, and a dearth of patrols on day 2 – a pattern that terrorists might easily exploit. Besides this, the lines in Figure 4(a) intersect frequently, indicating that on some days a higher-value target was visited more often while on other days it was visited less often, even though the value of a target does not change from day to day. This means that there was not a consistently high frequency of coverage of higher-value targets before PROTECT. In Figure 4(b), game-theoretic schedulers are seen to perform significantly better by avoiding predictability and ensuring that more important targets are covered with a higher frequency of patrols. The pattern of low patrols on day 2 disappears. Furthermore, lines no longer intersect frequently, meaning that higher-value targets are visited consistently throughout the week.

Conclusion

Security is recognised as a global challenge and game theory is an increasingly important paradigm for reasoning about complex security resource allocation. As described, the general model of security games is applicable (with appropriate variations) to varied security scenarios and there are applications deployed in the real world (see box, page 17) that have led to measurable improvements in security.

But while the deployed game-theoretic applications have shown a promising start, a significant amount of research remains to be done. These are large-scale interdisciplinary research challenges that call upon multi-agent researchers to work with researchers in other disciplines, as well as being "on the ground" with domain experts examining real-world constraints and challenges that cannot be abstracted away. ■



Reference

1. Greenberg, M., Chalk, P., Willis, H., Khilko, I. and Ortiz, D. S. (2006). *Maritime Terrorism: Risk and Liability*. Santa Monica, CA: RAND Center for Terrorism Risk Management Policy.