AI Technology for Quickly Solving Urban Security Positioning Problems

● Hiroaki Iwashita ● Kotaro Ohori ● Hirokazu Anai

Security games are used for mathematically optimizing security measures aimed at minimizing the effects of criminal activity. Their use has been attracting attention in the fields of artificial intelligence (AI) and multi-agent systems, and they are now being put to practical use by several U.S. public agencies. However, the use of urban network security games to analyze the problem of catching criminals at road checkpoints is difficult, which has hindered their application to city-scale networks. To overcome this difficulty, we have developed min-cut arrangement and graph contraction algorithms. The min-cut arrangement algorithm identifies candidate checkpoint locations that maximize security. The graph contraction algorithm reduces the problem, leading to a dramatic reduction in computational time for urban network security games, for which the computational cost increases exponentially with the size of the problem. In this paper, we introduce these algorithms and present results for a 200,000-node problem centered on the 23 wards of Tokyo.

1. Introduction

Ideal security measures would entail simultaneous surveillance of all places in cities and airports where it is likely for a crime to be committed. However, performing such surveillance on a daily basis would require an enormous amount of money. Therefore, an important challenge in actual surveillance planning is how to realize effective surveillance with limited surveillance resources. In addition, sophisticated analysis such as quantification of damage risk considering the behavior and psychological characteristics of criminals is required to evaluate the effects of surveillance. Currently, surveillance planning mainly relies on the experience and hunches of experts. However, as the sense of crisis over increasingly sophisticated organized crimes and other fraud is increasing, it is expected that artificial intelligence (AI) technology will be applied to surveillance planning.

In the AI field of study, the technology used to analyze complicated systems consisting of multiple agents is called multi-agent technology. In recent years, security game technology especially is being actively developed by applying game theory to the surveillance plan decision problem.[1)–3)] Security game technology includes problem-solving methods that depend on the individual surveillance target such as mathematical modeling and addressing the problem size (scalability improvement). Some target-related technologies have reached the practical level and are already used by organizations such as the Los Angeles International Airport,[4)] the Federal Air Marshals Service,[5)] the U.S. Coast Guard,[6),7)] and the Los Angeles County Sheriff's Department.[8)] However, practical use of such technology for many targets requires addressing various challenges.

Fujitsu Laboratories is jointly studying AI mathematical technologies related to security games with the University of Electro-Communications. This paper provides an overview of new findings related to technology for improving the scalability of urban network security games, one of our study themes.[9)]

2. Background

This section provides an overview of security games and describes the difficulties of urban network security games.

2.1 Security games

A security game is a mathematical model in which a defender (such as the police or a security company) and an attacker (a criminal or criminal organization) predict each other's behavior and make rational decisions about how to act. Each player has an individual collection of pure strategies (actionable behaviors). The gain each player gets is determined for all combinations of their pure strategies. Both players aim to maximize their gain. A mixed strategy, that is, a strategy in which pure strategies are stochastically selected, can be adopted for surveillance planning. Random selection of a surveillance plan is an important element in preparing for skillful criminals who research or steal the surveillance plan in advance.

For example, assume that only one security officer can be deployed even though Target A with a value of 60 and Target B with a value of 30 are located in separate places. The defender and attacker select and surveil or attack one of the targets. If the same target is selected, no damage occurs. If different targets are selected, the attack succeeds, and damage with the same value as the target occurs. A decisive surveillance plan to select only A or B or to alternately select A or B is completely powerless against criminals who have the ability to predict selection rules. In contrast, a surveillance plan to randomly select A or B with equal probability is not powerless: the expected damage is 30 if the criminal attacks A and 15 if the criminal attacks B. If the attacker is very skillful, a surveillance plan in which the selection probability ratio of A to B is 2:1 is optimal. In this case, the expected damage is 20.

As many security games resolve to a linear programming problem or integer programming problem, small problems can be solved using a general-purpose optimization solver. However, since real problems contain an enormous number of strategies and complicated gain structure, a solution in line with the problem is often required.

2.2 Urban network security games

In an urban network security game, the defender attempts to catch the criminal en route to a target. As shown in **Figure 1**, the city network is expressed as a graph structure consisting of nodes and edges, with some nodes connected to edges. Several nodes in the graph are labeled "source" or "target." A source represents a point at which the criminal may enter the network while a target represents a point that may be attacked. A "value" is assigned to each target to represent the damage generated if the target is attacked. The defender deploys a security officer to each

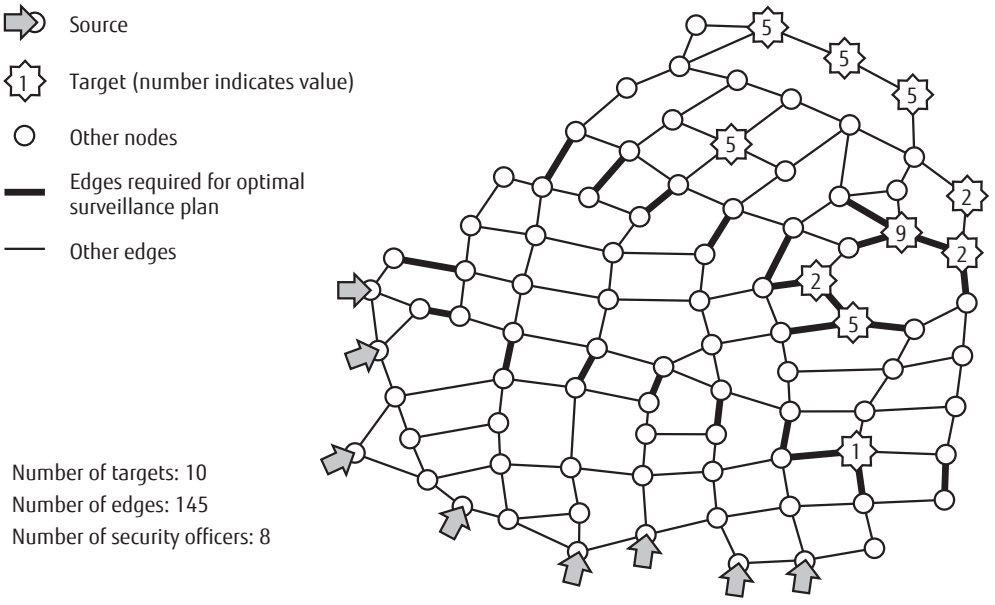

Figure 1
Urban network surveillance problem.

of k edges arbitrarily selected. The criminal selects one route from an arbitrary source to an arbitrary target. Unless the route selected by the criminal contains edges with security officers, the damage corresponding to the value of the attacked target occurs.

Figure 1 shows a sample small problem. Because the defender deploys security officers to edges, an urban network security game needs many more pure strategies than a basic security game in which the defender deploys security officers to targets. The number of combinations is 45 if 8 security officers are deployed to 10 targets. However, if 8 security officers are deployed to 145 edges, the number of combinations is as many as 3,981,762,826,470. The number of pure strategies increases much more for the attacker. The number of routes to be considered, which is 10 when a target is selected, is 9,806,370,645,605,329 for the entire game. In the basic formulation of a linear programming problem, the number of pure strategies for both players is directly proportional to the number of variables or restriction conditions. Pure strategies in a real urban network that consists of several tens of thousands to hundreds of thousands of nodes and edges are uncountable. Therefore, it is critical to develop a highly scalable computation technique for practical applications.

Although multiple techniques have been proposed at international conferences at the forefront of the AI field,[10)–12)] solutions that can stably handle problems with several tens of thousands or hundreds of thousands of nodes have not been found. The best conventional technique (hereafter, the conventional technique)[12)] enables application to real problems by improving the double-oracle method[13)] and combining other technologies without changing the formulation to seek game balance amid an enormous number of pure strategies. With the double-oracle method, a balance is established by expanding and repeatedly computing subsets of pure strategies used by both players because it is not necessarily required to use all the pure strategies in a good mixed strategy (allocate a probability other than 0).

A follow-up experiment that we conducted using the conventional technique demonstrated that the number of pure strategies required for computation to converge on a solution varies greatly depending on the problem. While a problem with several tens of thousands of edges can be solved in several seconds in some cases, it might not be able to be solved in a day in others. The mathematical model has to be improved to radically reduce the number of distinct pure strategies to obtain a solution more quickly and stably.

3. Developed technology

Observation of the surveillance plan obtained with the conventional technique showed that the defender uses only a part of their set of pure strategies and that security officers can be deployed to a part of the set of edges. For example, one of the optimal surveillance plans for the problem in Figure 1 can be made using only the edges shown in bold in the figure. We used this characteristic to significantly reduce the computation cost. The developed algorithm computes the solution by expanding the set of edges to which security officers are deployed (candidate edges) and repeating the computation.

3.1 Minimum cut deployment algorithm

When separating the source and target by removing one or more edges from the graph, the edge set from which the minimum number of edges is removed is called the minimum cut. In the optimal surveillance plan for a single target, security officers are deployed to the minimum cut.[14)] If the number of edges that constitute the minimum cut w is equal to or less than the number of security officers k, the target is perfectly surveilled, and the expected damage is 0. If the number of edges w is larger than k, the optimal plan is to uniformly and randomly select and surveil k of w edges, and the expected damage to a target with value $U(t)$ is $U(t) \cdot k/w$.

It is also desirable to select candidate edges from the minimum cut for an individual target or a combination of multiple targets if multiple targets have different values. However, it is not obvious which one of the various minimum cuts should be selected. It is not practical to consider all combinations ($2^{|T|}$-1) if the number of targets $|T|$ is large. If too few minimum cuts are selected, surveillance quality degrades. If too many minimum cuts are selected, the computation cost increases.

We solved this problem by improving the candidate edge set and the surveillance plan at the same time. An overview of this algorithm is provided below:

1) The initial value of candidate edge set C is an empty set. No targets are surveilled in the first surveillance plan.
2) Select the set of targets with the highest expected damage in the current surveillance plan T'.
3) Add the minimum cut that separates T' from the source set to C. End processing if C does not become larger.
4) Calculate the optimal surveillance plan regarding the new C as the candidate edge set and go back to 2).

Figure 2 shows how this algorithm selects minimum cuts. No targets are surveilled in the first surveillance plan. Because the target that has the highest expected damage is the target with a value of 9, C_1 is selected as the first minimum cut. In the second surveillance plan in which C_1 is the candidate edge set, security officers are deployed to the four edges in that set. This completely protects the target with a value of 9. Now, five targets with a value of 5 have the highest expected damage. Therefore, the minimum cut selected second is C_2, and this cut separates these five targets from the source. In the third surveillance plan, the probability of deploying eight security officers to the total of C_1 and C_2, which is 13 candidate edges, is optimized. As a result, a balance is established that makes the expected damage of the six targets with a value of 5 or more 2.228, and C_3, which separates them from the source is selected third. In the fourth surveillance plan in which C_3 is added to the candidate edge set, the expected damage of all nine targets separated with C_3 is less than 1 (at most 0.824). Then, C_4, which enhances the surveillance of the remaining targets with a value of 1 (expected damage: 1), is added to the candidate edge set. Finally, minimum cut C_5, which protects all the targets, is added, completing a surveillance plan in which the expected damage is 0.900 at worst.

3.2 Simplification by contracting graph

With the above algorithm, the problem of calculating the optimal surveillance plan with surveillance deployment restricted to candidate edges must be repeatedly computed. This section introduces a technique to dramatically streamline this computation.

Under the candidate edge restriction, differences in the movement of the attacker in relation to non-candidate edges do not affect the gain of either player. Therefore, we introduced a new simplified model that ignores these edges. This model considers connected components that consist of non-candidate edges and nodes adjacent to them and a graph that contracts each of them to a node. For example, if C_1 to C_5 are all considered candidate edge restrictions in Figure 2, the

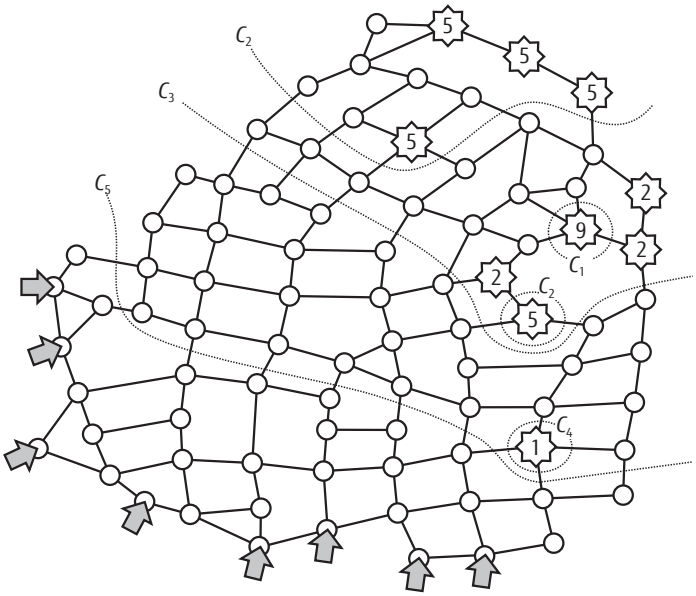

Figure 2
Group of minimum cuts that realize effective surveillance.

graph can be contracted as shown in **Figure 3**. In addition, because it is not necessary to distinguish multiple edges that connect the same node pairs, this problem can be simplified to a weighted graph, as shown in **Figure 4**. Here, the weight indicates the edge capacity. Security officers can be deployed to each edge up to its capacity. The probability of a criminal passing through each edge encountering a security officer is equal to the ratio of security officers deployed to the capacity. The optimal surveillance plan calculated on the simplified urban network is theoretically guaranteed to remain optimal even if it is associated with the original urban network (under the candidate edge restriction).[9]

With the simplified graph, the number of pure strategies for both the defender and the attacker is significantly reduced. In the model in Figure 4, the deployment patterns of eight security officers are reduced to 2,690, and the routes from the source to the target are reduced to only 18.

4. Trial results

We evaluated the performance of this technique by using a real road network to determine whether it can be applied to large problems. Computation results using data for the extracted area (the 23 wards of Tokyo from latitude 35.5322 to 35.8189 north and longitude 139.583 to 139.920 east from OpenStreetMap[15]) are shown in **Table 1**. This area has 202,547 nodes and 329,609 edges. Twenty nodes were randomly selected as sources and 20 were randomly selected as targets, and an integer value from 1 to 10 was randomly assigned to each target as its value. The CPU in the computer used was Intel Xeon Processor E3-1275 (3.60 GHz).

This technique was able to stably solve the problem within a realistic time (<5 minutes) whereas the conventional technique[12] could not solve it. Comparison of the computation speed for problems that could be solved with the conventional technique without difficulty showed that the speed was improved by 20 times on average for problems with 100 nodes and 500 times on average for problems with 200 nodes.

5. Conclusion

This paper provides an overview of findings for significantly improving the scalability of urban network security games developed by Fujitsu Laboratories.

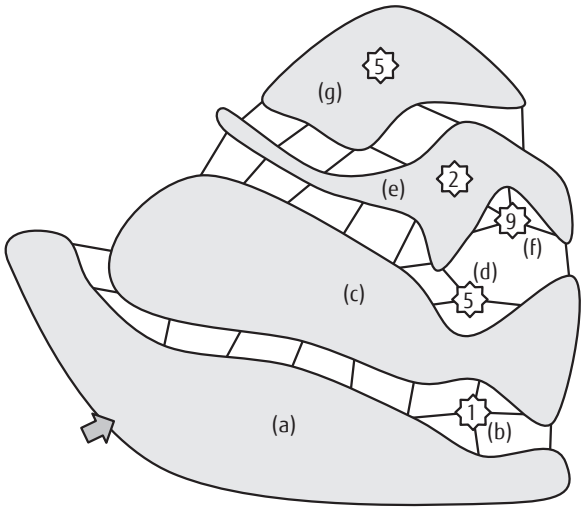

Figure 3
Contraction of non-candidate edges.

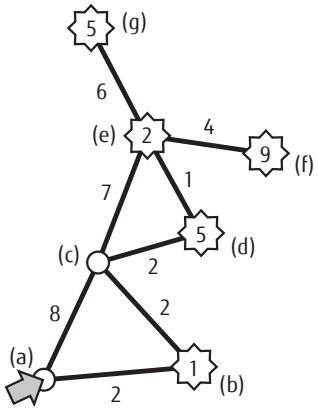

Figure 4
Simplified urban network.

Table 1
Relationship between number of security officers and computation time.

Number of security officers	10	20	30	40	50	60	70
Expected damage	4.71	2.91	1.88	1.25	0.71	0.24	0.00
Computation time (s)	149	222	254	257	291	291	101

Details on the optimization computation technique using mathematical programming and the evaluation results including comparison with the conventional method are available elsewhere[9].

Future work includes expanding the application areas of these findings to contribute to urban security technology.

References

1) M. Tambe: Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned. Cambridge University Press, 2011.
2) M. Jain et al.: An Overview of Recent Application Trends at the AAMAS Conference: Security, Sustainability, and Safety. AI Magazine, Vol. 33, No. 3, pp. 14–28, 2012.
3) A. Iwasaki et al.: An Overview of Studies on Game Theory and Mechanism Design. Journal of the Japanese Society for Artificial Intelligence, Vol. 28, No. 3, 389–396, 2013.
4) J. Pita et al.: Deployed Armor Protection: The Application of a Game Theoretic Model for Security at the Los Angeles International Airport. International Conference on Autonomous Agents and Multi-agent systems (AAMAS), pp. 125–132, 2008.
5) J. Tsai et al.: Iris–A Tool for Strategic Security Allocation in Transportation Networks. International Conference on Autonomous Agents and Multi-agent Systems (AAMAS), 2009.
6) E. Shieh et al.: Protect: A Deployed Game Theoretic System to Protect the Ports of the United States. International Conference on Autonomous Agents and Multi-agent Systems (AAMAS), pp. 13–20, 2012.
7) F. Fang et al.: Optimal Patrol Strategy for Protecting Moving Targets with Multiple Mobile Resources. International Conference on Autonomous Agents and Multi-agent Systems (AAMAS), pp. 957–964, 2013.
8) Z. Yin et al.: Trusts: Scheduling Randomized Patrols for Fare Inspection in Transit Systems. Innovative Applications of Artificial Intelligence Conference (IAAI), 2012.
9) H. Iwashita et al.: Simplifying Urban Network Security Games with Cut-Based Graph Contraction. International Conference on Autonomous Agents and Multi-agent Systems (AAMAS), pp. 205–213, 2016.
10) J. Tsai et al.: Urban Security: Game-Theoretic Resource Allocation in Networked Physical Domains. Conference on Artificial Intelligence (AAAI), pp. 881–886, 2010.
11) M. Jain et al.: A Double Oracle Algorithm for Zero-Sum Security Games on Graphs. International Conference on Autonomous Agents and Multi-agent Systems (AAMAS), pp. 327–334, 2011.
12) M. Jain et al.: Security Scheduling for Real-World Networks. International Conference on Autonomous Agents and Multi-agent Systems (AAMAS), pp. 215–222, 2013.
13) H. B. McMahan et al.: Planning in the Presence of Cost Functions Controlled by an Adversary. International Conference on Machine Learning, pp. 536–543, 2003.
14) A. Washburn et al.: Two-Person Zero-Sum Games for Network Interdiction. Operations Research, Vol. 43, No. 2, pp. 243–251, 1995.
15) OpenStreetMap. *https://www.openstreetmap.org/*

Hiroaki Iwashita
Fujitsu Laboratories Ltd.
Dr. Iwashita is currently engaged in research and development on AI technologies for solving social issues.

Kotaro Ohori
Fujitsu Laboratories Ltd.
Dr. Ohori is currently engaged in research and development on AI technologies for solving social issues.

Hirokazu Anai
Fujitsu Laboratories Ltd.
Dr. Anai is currently engaged in research and development on theory and application of mathematical analysis, optimization, and artificial intelligence.

