CCP game : a game theoretical model for improving the scheduling of chemical cluster patrolling

Abstract

Chemical clusters can be attractive targets for terrorism, due to the extremely importance of them as well as due to the existence of dangerous materials. Patrolling is scheduled for better securing chemical clusters. However, the current patrolling strategies fail on competing with intelligent attackers and therefore can be non-optimal. The so-called Chemical Cluster Patrolling (CCP) game is proposed in this paper. The CCP game employs game theory as a methodology, aiming at randomly but strategically scheduling security patrols in chemical clusters. The patroller and the attacker are modelled as the two rational players in the CCP game. The patroller's strategy is defined as probabilistically traveling within the cluster or patrolling some plants while the attacker's strategy is formulated as a combination of an attack target, the start time of the attack, and the attack scenario to be used. The Stackelberg Equilibrium and a robust solution which takes into consideration of the patroller's distribution-free uncertainties on the attacker's parameters are defined for predicting the outcome of the CCP game. Results of the case study indicates that the patrolling strategy suggested by the CCP game outperforms both the fixed patrolling route strategy and the purely randomized patrolling strategy.

Highlights

1) A Chemical Cluster Patrolling game for optimizing chemical cluster patrol is proposed;

2) Patroller's distribution-free uncertainties on attackers' parameters are modelled;

3) Algorithms are proposed for calculating the Stackelberg Equilibrium and the robust solution;

4) A case study is conducted to verify the advantages of the CCP game.

Key words

Patrolling Game; Chemical Cluster Security; Game Theory

1. Introduction

Due to economies of scale and all kinds of collaboration benefits, chemical plants are usually geographically clustered, forming chemical industrial parks or so-called 'chemical clusters'. Some examples of such clusters are the Antwerp port chemical cluster in Belgium, the Rotterdam port chemical cluster in the Netherlands, the Houston chemical cluster in the US, or the Tianjin chemical cluster in China. Alongside with the benefits, being geographically clustered also interconnects the risks of each plants. For instance, due to the existence of domino effects, a major explosion caused by malicious attackers in one plant may cause failures (e.g., explosion, fire, leakage) of facilities in neighbour plants, worsening the consequences.

The importance of protecting chemical facilities from intentional attacks (e.g., terrorists, criminal acts, sabotages, etc.) has been emphasized frequently in Baybutt's publications [1-4]. Not only the physical security perspective is important, but also the cyber perspective should be taken into consideration [5]. Gupta and his co-authors [6-9] suggested that security risk management in the process industries should involve threat analysis, vulnerability analysis, security countermeasures, and emergency response. Reniers and his co-authors [10-15] conducted security research in the chemical clusters, from the management factors to the technic factors. A model estimating the vulnerabilities of industrial facilities to attacks with improvised devices are proposed by Landucci et al. [16]. Argenti et al. [17-20] employed Bayesian network for assessing the attractiveness and vulnerabilities of chemical facilities, conditional probabilities of which were estimated based on interviews with industrial practitioners. Khalil [21] proposed a probabilistically timed dynamic model for bettering physical protection of critical infrastructures. His model fails on capturing the intelligent interactions between the defender and attacker. Meanwhile, game theory is mentioned in Khalil [21] for future extension of his model. Song et al. [22] developed a graphical approach for visualizing the vulnerabilities of a chemical facility to an intrusion attack. Besides Argenti et al. [17-19], Bayesian

network is also employed in some other literatures for visualizing and quantifying security risks in chemical industries [23-26].

Besides fixed security countermeasures within each plant, the patrolling of security guards is also scheduled, for securing these chemical facilities at different points and times, e.g. at night. The patrolling can either be single-plant oriented, which can be completely scheduled by the plant itself, or it can be multiple-plants oriented, which should be scheduled by an institute at a higher level than the single-plant level, for instance a multiple plant council (MPC) [27].

In the current patrolling practise, some patrollers follow a fixed patrolling route (i.e., the same patrolling route is used in different days). If a fixed patrolling route is scheduled, the patroller's real-time location is deterministic to human/intelligent attackers since intelligent attackers would collect useful information before an attack. Other patrollers purely randomize their patrolling, without taking into consideration the hazardousness level that each installation/facility/plant holds, and if this is the case, an intelligent attacker may focus to attack the most dangerous installations/facilities/plants since all installations/facilities/plants are equally patrolled. Therefore, both the fixed patrolling strategy and the purely randomized patrolling strategy have a drawback of not being able to deal with intelligent attackers.

Game theory [28], a methodology proposed by mathematicians and economists, has the advantage on modelling strategic decision making in a multiple stakeholders' situation. The outcome (e.g., catch an attacker or nothing happens) of a security patrolling in a chemical cluster depends on both the patroller's behaviour and the attacker's behaviour. Furthermore, both the patroller and the attacker are intelligent human beings. Therefore, game theory is a promising approach for improving the security patrolling in the chemical clusters. Actually, game theory has been introduced for improving patrol scheduling in some other domains. Among others, Shieh et al. [29] proposed a game theoretic model for optimizing patrolling of

protecting ferries in the Boston port. The model proposed by Shieh et al. is innovative on optimizing patrols for the protection of moving targets. Fang et al. [30] developed the so-called green security game (GSG) for scheduling patrolling for the conservation of wild animals. The GSG is a repeated game and the statistic learning technique is employed for modelling the poacher's behaviour. Amirali et al. [31] introduced a model based on game theory to better patrol pipelines. Alpern et al. [32] tried to analytically solve the patrolling game in graph, and they achieved theoretical results in the patrolling game in line graph [33, 34]. However, no research has been done thus far for employing a game theoretic model to optimize patrolling in chemical industrial parks.

The patrolling in a chemical cluster is different to the patrolling in a port or in a wildlife conservation area or for a pipeline. In the latter cases, the patrolling object (i.e., a port, an area, or a pipeline) is modelled as a graph. The patroller travels in the graph passing by different nodes of the graph (without staying at the nodes), and the attacker would be detected if the patroller and the attacker meet each other on one of the nodes in the graph. For instance, in a pipeline patrolling task, if the patroller arrives the point where the attack is happening, then the attacker would be definitely detected. For analysing the patrolling in a chemical cluster, the patrolling object (i.e., the cluster) is also modelled as a graph, of which the nodes are the plants in the cluster. The patroller travels in the graph and she[1] stays a certain period of time in some nodes which means that she patrols the plant. The attacker has a probability of being detected if the patroller patrols the target plant when the attack is happening. Therefore, the above mentioned patrolling games are not directly applicable for the scheduling of the chemical cluster patrolling.

The present paper therefore proposes a Chemical Cluster Patrolling (CCP) game, answering the question how to optimally randomize patrolling in a chemical cluster, in a way that it is

[1] In this paper, we denote the patroller/defender as she/her/her, and denote the attacker as he/him/his.

better secured, by using a game theoretical approach. The remainder of the paper is organized as follows: Section 2 briefly introduces how patrolling is organized in chemical clusters. Section 3 proposes the chemical cluster patrolling game. An illustrative case study is investigated in section 4. Section 5 discusses the implementation and observation errors of the model. Conclusions are drawn in section 6.

2. Patrolling in chemical clusters

2.1 A brief patrolling scenario within a chemical cluster

The patrolling scenario is assumed to be the following. A patroller team (e.g. two guards) drives a car randomly, patrolling in each of the plants. In each plant, the team drives into the plant and conducts a patrolling task and/or some other security related actions (e.g., record the arrival of the team, TBD) in the plant. Besides each plant's own countermeasures (e.g., entrance control, cameras, employee awareness etc.), if during the attacker's attack and intrusion procedure, the patroller is patrolling in the plant, then the attacker would have a probability of being detected. After patrolling during a specified period of time in a plant, the patrolling team moves to another plant belonging to the geographical cluster, via the (public) road. However, the attacker may know the patroller's daily patrolling routes, for instance, by long-term observation or by stealing the patroller's security plan.

2.2 Formulating the research question

2.2.1. Graphic modelling

A chemical cluster can be described as a graph $G(V, E)$ where V represents the number of vertices (or nodes) of the graph, and E is the number of edges of the graph. The vehicle entrances of every plant and the crossroads that are situated on the road form the nodes of the graph. The roads between different plants (to be more specified, it should be "between

different entrances") are modelled as edges of the graph. Furthermore, all entrance nodes which belong to the same plant are modelled to be fully connected, which means edges also exist between every two nodes in these cases.

For example, Figure 1 gives the layout of a small part of the Antwerp port chemical industrial park. There are five plants in this picture, indexed as plant 'A', plant 'B', and so forth. The yellow dot lines demonstrate the roads, which is the only infrastructure where the patroller can drive. Figure 2 shows the graph model of the cluster shown in Figure 1. As we may notice, plants 'A', 'C', 'D', and 'E' in Figure 1 are modelled as a node (with the same name) in Figure 2. The cross point of the vehicle road between plant 'D' and 'E' in Figure 1 is also denoted as a node in Figure 2 (i.e., node 'cr'). Moreover, plant 'B' has two vehicle entrances, and therefore two nodes (i.e., nodes 'B1' and 'B2') are used in Figure 2 to denote these two different entrances of plant 'B'. Edges 'e1' to 'e6' reflect the vehicle roads between different plants, while edge 'e7' is added between nodes 'B1' and 'B2' because these two nodes belong to the same plant and hence should be connected.

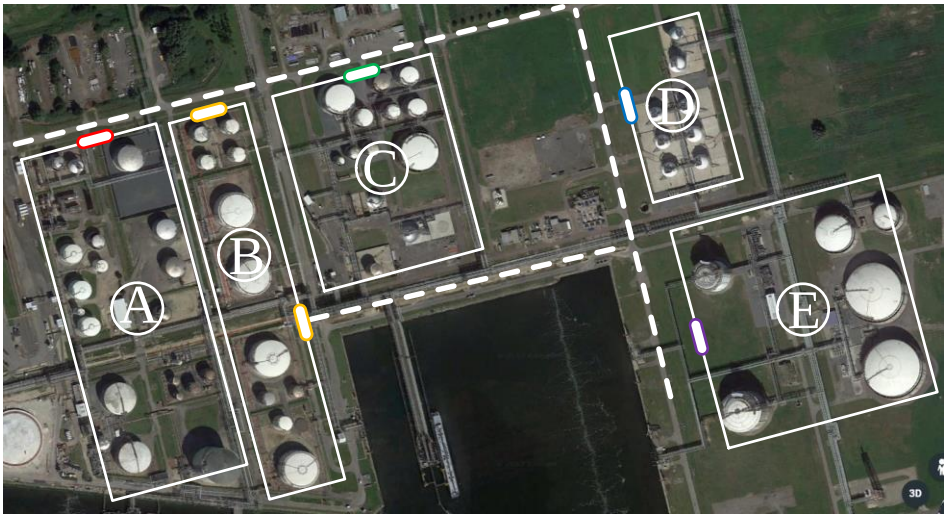

Figure 1. layout of a chemical park in Antwerp port

Based on the graphic model, the patrolling scenario in section 2.1 can be described as a graphic patrolling problem: 1) a patroller (team) starts her patrolling from a node (the base

camp); 2) she moves in the graph; 3) when arriving at a node, she may decide whether to stay at the node for a specific period of time t_k^p (i.e., patrol the plant) or not (i.e., move to another plant without patrolling the current plant); 4) after a period T, the patroller terminates the patrolling and goes back to her base camp.

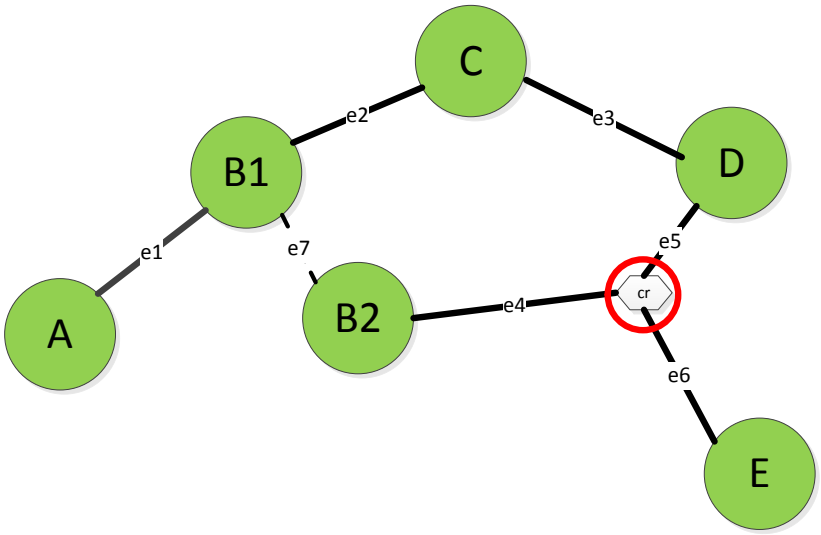

Figure 2. Graphic modelling of the chemical park

In the above statement, t_i^p represents the patrolling time in plant i. t_i^p is determined both by the plant and by the patrolling scenario. For instance, territorially big plants may have a longer t_i^p. Moreover, the patroller, if coming into plant i, can also have several different patrolling intensities, and more intensive patrolling needs a longer t_i^p, and vice versa. t_i^p would also be slightly influenced by the entrances where the patroller comes into and leaves the plant. In this paper, for the sake of simplicity, we assume that each plant has a fixed t_i^p, without considering the influence of different entrances and without considering the multiple patrolling intensities. T represents the total patrolling time, and its typical value can be, for instance, 3 hours. Table 1 further demonstrates all the notations used in this paper.

Table 1. Definitions of Notations

Notation	Definition	Type*
$\boldsymbol{G(V,E)}$	The graphic model of the chemical cluster, defined in section 2.2.1.	MG
$\boldsymbol{t_e^d}$	The patroller's travelling time on edge $e \in E$.	IN
$\boldsymbol{t_i^p}$	The patroller's patrolling time within plant i.	IN
$\boldsymbol{k^i}$	The intrusion and attack continuing time, in plant i.	IN
$\boldsymbol{T}$	Total patrolling time.	IN
$\boldsymbol{pG(pV,pE)}$	The patrolling graph of the chemical cluster, defined in section 2.2.2.	MG
$\boldsymbol{\lvert V\rvert}$	Nodes number of graph G.	MG
$\boldsymbol{sC}$	Superior connection matrix of graph G.	MG
$\boldsymbol{dis(bcn,nd)}$	The shortest distance (in time) in the graph G from the base camp node bcn to node $nd \in G$.	MG
$\boldsymbol{c_{s-e}}$	Probability that the patroller takes the action represented by edge $(s,e) \in pE$.	MG
$\boldsymbol{R^d}$	The patroller's reward by catching an attacker.	IN
$\boldsymbol{L^d}$	The patroller's loss if an attack is succeed.	IN
$\boldsymbol{P^a}$	The attacker's penalty if being caught.	IN
$\boldsymbol{G^a}$	The attacker's gain from a successful attack.	IN
$\boldsymbol{f_{cpp}, \tilde{f}_{cpp}}$	Probability that the attacker would be detected by the countermeasures of the plant, estimated from the defender and from the attacker's perspective respectively.	IN
$\boldsymbol{f_p}$	Probability that the attacker would be detected by the patroller.	MG
$\boldsymbol{f}$	Probability that the attacker would be detected.	MG
$\boldsymbol{\sigma_r}$	Probability that the patroller would detect the attacker in overlap situation r, defined in section 3.4.	IN
$\boldsymbol{\tau_r}$	Probability that the patroller would be in the overlap situation r, defined in section 3.4.	MG
$\boldsymbol{s_a}$ $\boldsymbol{(s_d)}$	An attacker (defender) pure strategy.	MG
$\boldsymbol{S_a}$ $\boldsymbol{(S_d)}$	Strategy set of the attacker (defender).	MG
$\boldsymbol{\vec{c}}$	The vector form of representing a defender's strategy.	MG
$\boldsymbol{sP_{pv}}$	The probability that the patroller would be at node $pv \in pV$.	MG
$\boldsymbol{cP_{pv}^{pe}}$	The conditional probability that patroller would take the action $pe \in pE$, in condition that she currently locates at $pv \in pV$.	MG

* IN means model inputs, and this kind of data should be provided by security experts; MG means model generated data.

A superior connection matrix sC of graph G is defined. The entry $sC(i,j)$ denotes the time needed for the patroller to move from node i to node j (of graph G). There are three possible situations of the relationship of nodes i and j: (i) these two nodes belong to different plants or

at least one of them is a cross road node (e.g., nodes 'A' and 'B1' in Figure 2). In this case, $sC(i,j)$ equals the time that the patroller needs to drive from node i to node j. (ii) these two nodes are different entrances of a plant (e.g., nodes 'B1' and 'B2' in Figure 2). In this case, $sC(i,j)$ equals the patrolling time of the plant. And (iii) these two nodes are the same. In this case, $sC(i,j)$ equals the patrolling time of the plant that the node belongs to.

In practice, situation (ii) means that the patroller comes into a plant and patrols the plant, but she comes in and out from different entrances. For instance, in Figure 2, the patroller comes into plant 'B' through entrance 'B1' and after patrolling plant 'B', she leaves the plant through entrance 'B2'. Situation (iii) means that the patroller comes into the plant and patrols it, and she comes in and out using the same entrance/exit gate. For instance, in Figure 2, the patroller comes into plant 'B' through entrance 'B1' and after patrolling the plant, she leaves the plant through entrance 'B1' again.

Ideally speaking, the patroller may also pass a plant without patrolling it, for a purpose of shortening the traveling time of arriving at her next patrolling plant. In the cluster shown in Figure 1 and 2, if the patroller wants to move from plant 'A' to plant 'E', instead following the route "A→B1→C→D→cr→E", she may also go the route "A→B1→B2→cr→E" without patrolling plant 'B'. In the latter route, since plant 'B' is not patrolled, the time needed from entrance 'B1' to entrance 'B2' can be quite short, resulting a short traveling time for the latter route than the former route. However in practice, this behaviour (e.g., passing the plant without patrolling it) increases the risk for the passing-by plant (e.g., plant 'B' in the above example) and therefore unless an agreement exists, the patroller would not be allowed to pass a plant without patrolling it. Therefore, situation (ii) in this research is assumed to only represent the case that the patroller patrols the plant.

For the cluster and the graph shown in Figure 1 and 2, if we set: $t_1^d = 2, t_2^d = 3, t_3^d = 4, t_4^d = 3, t_5^d = 2, t_6^d = 2$, and further set $t^p('A','B','C','D','E') = [9,7,6,5,7]$, then the superior matrix sG of the example can be shown in Table 2. t_i^d represents the driving time of edge 'ei' in Figure 2. For instance, t_1^d is the driving time from node 'A' to 'B1'. $t^p('X')$ denotes the time needed to patrol plant $'X'$. All the time-related data are unified in minutes.

Table 2. Superior connection matrix for Figure 2 with the illustrative numbers

	A	**B1**	**B2**	**cr**	**C**	**D**	**E**
A	9	2	∞	∞	∞	∞	∞
B1	2	7	7	∞	3	∞	∞
B2	∞	7	7	3	∞	∞	∞
cr	∞	∞	3	∞	∞	2	2
C	∞	3	∞	∞	6	4	∞
D	∞	∞	∞	2	4	5	∞
E	∞	∞	∞	2	∞	∞	7

2.2.2. Patrolling graph modelling

A directed patrolling graph $pG(pV, pE)$ is defined based on the graphic model of the chemical cluster. A node of pG is defined as a tuple of (t, i), in which $t \in [0, T)$ denotes the time dimension and $i \in \{1,2, \dots, |V|\}$ denotes a node in graph $G(V, E)$ (i.e., a plant (entrance) in the chemical cluster). Node (t, i) means that at time t the patroller arrives or leaves node i. A directed edge of pG from node (t_1, i_1) to node (t_2, i_2) therefore denotes a patroller action where she moves from node i_1 at time t_1 to node i_2, and arrives at t_2. Table 3 shows an iterative algorithm for generating the patrolling graph $pG(pV, pE)$. $dis(bcn, nd \in G)$ is the shortest distance (in time) in graph G from the base camp node bcn to node nd.

Figure 3 shows the patrolling graph pG for the chemical cluster shown in Figure 1, with the data in Table 2 and further assuming a patrolling time $T = 30$. The patroller's base camp is assumed to be close to the cross road node, thus 'cr' is chosen as the patroller's base camp.

Table 3. an algorithm of generating the patrolling graph

Algorithm: generating the patrolling graph

1. *Construct an empty temporary node list tNL, an empty node list pV, an empty edge set pE;*
2. *Construct node $pv = (0, bcn)$, in which bcn is the patrolling base camp node in graph G;*
3. *Initialize $tNL \leftarrow pv, pV \leftarrow pv$;*
4. *While tNL not empty, do*
   - *4.1. Get the first node in tNL, denoted as the current node $cv = (ct, cn)$;*
   - *4.2. Construct follow-up nodes of cv;*
     - *4.2.1. in graph G, find all the connected nodes of cn, representing as $ccn = \{nd \in V | sC(cn, nd) < \infty\}$;*
     - *4.2.2. for each $nd \in ccn$, if $ct + sC(cn, nd) \leq T + dis(bcn, nd)$, construct a new node $nv = (ct + sC(cn, nd), nd)$ and a directed edge ne from cv to nv should also be constructed;*
     - *4.2.3. add ne to pE;*
     - *4.2.4. if nv in pV already, continue; otherwise, insert nv into tNL, add nv to pV;**
   - *4.3. remove cv from tNL*
5. *end*

* tNL should be sorted according to the nodes' time

In Figure 3, the x axis denotes the time dimension, while the y axis represents the different nodes in Figure 2. Therefore, any coordinates in Figure 3 can be a possible node for pG. As we may see, node 1 (at the left-hand side of the figure) in Figure 3 is $(0, 'cr')$, which means that at time 0, the patroller starts from her base camp (i.e., 'cr'). Thereafter she has 3 choices: (i) to come to plant 'B' (more accurately, entrance 'B2') with a driving time t_4^d, and reaches node 2 (i.e., (3,'B2')); (ii) to come to plant 'D' with a driving time t_5^d, and reaches node 3 (i.e., (2,'D')); and (iii) to come to plant 'E' with a driving time t_6^d, and reaches node 4 (i.e., (2,'E')). Subsequently, at new nodes (e.g., 2, 3, or 4), the patroller has the same choice problem, that is, to patrol the current plant or to come to another plant. Finally, when time comes to the end of the patrol, the patroller terminates the patrol and comes back to her base camp. In Figure 3, the indexes of some nodes and the weight of some edges are not shown, for the purpose of improving the visibility of the figure. Furthermore, the actions (edges) that the patroller comes

back to her base camp are not shown, since these actions do not have an influence on the patrolling results.

A fixed patrolling route is a series of edges $(pe^1, pe^2, \dots, pe^{len})$ in the patrolling graph that satisfies the following three conditions: (i) the in-degree of the start node of pe^1 is 0; (ii) the out-degree of the end node of pe^{len} is 0; and (iii) pe^i and pe^{i+1} $(i = 1,2, \dots, len-1)$ are linked, which means that the end node of pe^i is the start node of pe^{i+1}. For instance, the bold (and black) line in Figure 3 denotes a fixed patrolling route, and it is: $'cr' \rightarrow 'D' \rightarrow 'C' \rightarrow$ patrol plant $'C' \rightarrow 'B1' \rightarrow$ patrol plant $'B' \rightarrow$ leave plant $'B'$ from 'B2' $\rightarrow 'cr' \rightarrow$ $'E' \rightarrow' cr' \rightarrow 'E'$.

A purely randomized patrolling route is defined as: "at a node of the patrolling graph, the patroller goes to each edge outgoing from the node with an equal probability." For instance, in Figure 3, at node 1 $(0, 'cr')$, the patroller goes to node 2, 3, or 4 with a probability 1/3, and at node 2 $(3,' B2')$, the patroller goes to node 9, 10, or 11 all with a probability 1/9, and so forth.

To keep the continuity of coverage of each plant, the patroller is required to prolong her patrolling in the plant until the next patroller team might be able to arrive at the plant (see step 4.2.2 in Table 3). For instance, in Figure 3, though the patrolling time is set as $T = 30$, however, the patrolling in plant 'A' is not stopped until $t = 41$. The idea is that, the shortest time that the next patrolling team can arrive at plant 'A' (from 'cr') is 11 (By following a path $'cr' \rightarrow 'B2' \rightarrow 'B1' \rightarrow$ 'A'). If the current patroller team does not prolong her patrolling, and the next patroller team starts at time 30 and starts from her base camp (i.e., 'cr'), then plant 'A' would definitely not be covered during time $(30,41)$. This approach may increase the patroller's workload. However, if we set T slightly smaller than the patroller's real workload, the problem will be solved. For example, if a patroller team's workload is 240 minutes per day, for modelling reasons we set it at $T = 220$.

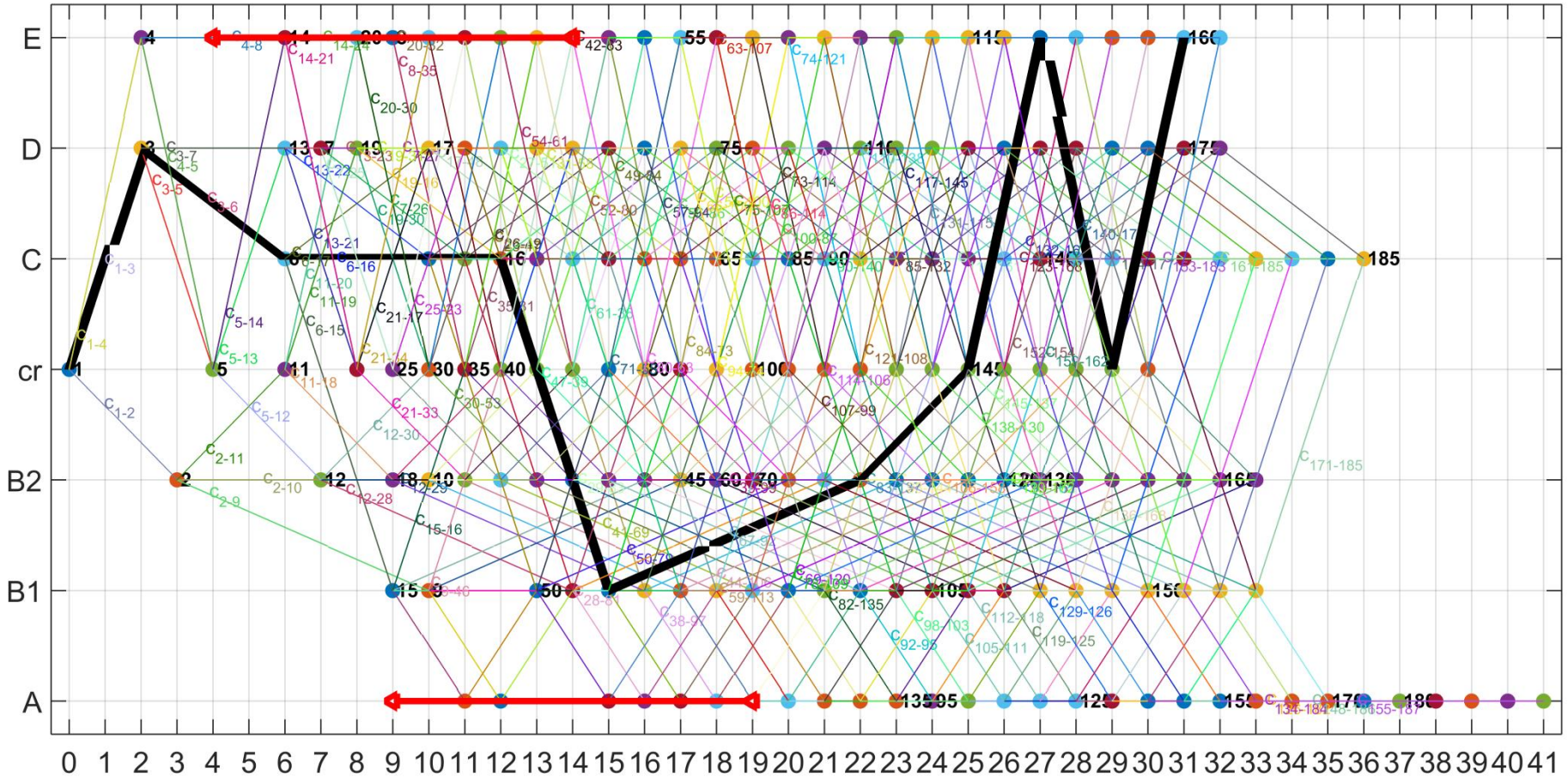

Figure 3. Patrolling Graph of the illustrative example

The way that we deal with the continuity of patrolling coverage (or the periodic patrolling problem) implies that during time $[T, T + \max(dis(bcn, nd \in G))]$, there might be two patrolling teams in the industrial park at the same time. Nevertheless, in each plant, there is maximally one patrolling team present. The second patrolling team starts from her base camp at time T and also probabilistically schedules her actions according to the patrolling graph. Therefore, the time period of $[30,41]$ of Figure 3 should actually be an overlap.

2.2.3. Time discretization

The time dimension (x axis) of the patrolling graph is continuous. Therefore, the patroller's traveling time t^d and patrolling time t^p are not necessarily integers. Moreover, the adversary's attack can happen at any time belonging to the continuous time interval $[0, T)$.

In our model, we discretize the time dimension of the patrolling graph. The time interval $[0, T)$ is divided to be multiple equal time slices and the length of each time slice can be, for instance, a second or a minute. All the time-related parameters (e.g., the patroller's traveling time and patrolling time, the attacker's attack period) are rounded to their closest integer numbers of the time slice. For instance, if there is a $t^p = 6.3\ minutes$ and the time slice is defined as $1\ minute$, then we would have $t^p = 6$. Moreover, the attacker can only start his attack at the beginning of each time slice and his attack period lasts for several time slices. Consequently, any actions of the patroller and the attacker would happen at the beginning points of each time slice, and we therefore denote the time interval $[0, T)$ as $\{0, \dots, \overline{T} - 1\}$, of which the latter means all the non-negative integers smaller than $\overline{T}$ and $\overline{T}$ is the number of time slices.

Discretization of the time axis simplifies the model. As we will see in section 3.2, by discretizing the time axis, all the attacker's actions can be enumerated. Furthermore, discretizing the time axis also makes it easier to calculate the detection probabilities, as shown

in section 3.4. Discretization of the time dimension is also reasonable from a practical point of view. Although time is continuous in reality, we would stop at a certain accuracy, for instance, at seconds. Therefore, if the length of a time slice is short enough, the discretization model describes the reality very well.

3. Chemical Cluster Patrolling game

The Chemical Cluster Patrolling (CCP) game is proposed in this section. We introduce the game from four aspects, namely, the players modelling, the strategies (set) modelling, the payoffs modelling, and the solutions of the game.

3.1 Players

Players of the chemical cluster patrolling (CCP) game are the patroller team on the one hand (defender) and the potential adversaries on the other (attacker). The CCP game is a two players game and both players are assumed with perfect rationality. Future research efforts can be given to extend the model to deal with boundedly rational attackers.

3.2 Strategies

Attacker strategy

An attacker's strategy consists of three parts: (i) determine a target plant to attack; (ii) determine a time to start the attack; and (iii) determine an attack scenario to use. Different attack scenarios may need different intrusion and attack efforts, resulting in different attack continuing times. For instance, generally speaking, an attack scenario with a suicide bomber needs less time than an attack scenario aiming to steal hazardous materials from the chemical plant, since there is an exit step for the latter scenario.

An attacker's pure strategy can be denoted as Formula (1).

$$s_a = (t, i, k_i) \quad \cdots\cdots (1)$$

In which t denotes the attack start time, i represents the target plant, k_i is the attack continuing time (e.g., 10 minutes) which should be determined by both the attack scenario and the target plant.

Example: the two horizontal bold dot red lines in Figure 3 represent attacks of attacking plant 'A' at time 9 (the line at below) and of attacking plant 'E' at time 4 (the line at above), with an intrusion and attack continuing time of ten time units, respectively.

Formula (1) implies that the attacker would only attack one plant. The number of the attacker's pure strategies can be calculated by Formula (2). In which m is the number of pure strategies of the attacker; n denotes the number of plants in the cluster; T is the total time slices; and Sce is the number of different attack scenarios.

$$m = n \cdot T \cdot Sce \quad \cdots\cdots (2)$$

Patroller strategy

The patroller's strategy is to randomize her patrolling and to bring maximal uncertainties (about her location) to the attacker. According to the patrolling graph we constructed in section 2.2, at each node of pG, the defender may choose to patrol the current plant or move to other adjacent plants, and her choices are represented as the edges in pG. Therefore, if we assign a probabilistic number to each edge of pG, and define the number as the probability that the defender may go that edge (please recall the meaning of an edge in pG, as stated in section 2.2), then the patroller's strategy is the combination of these probabilistic numbers. A mathematic formulation of the defender's strategy is shown in Formula (3).

$$s_d = \prod_{(s,e)\in pE} c_{s-e} \quad \cdots\cdots (3)$$

In which c_{s-e} denotes the probabilistic number assigned to the edge from node s to node e, $\prod$ denotes the Cartesian product of all edges in pG (i.e., all $(s, e) \in pE$).

An intermediate node of pG is a node that has both income edges and outcome edges. A root node of pG is a node that has no income edges. For instance, node $(0,' cr')$ in Figure 3 is a root node, but not an intermediate node, while node $(2,'D')$ is an intermediate node but not a root node. An important property of probabilities c_{s-e} is that, for each intermediate node (of pG), the sum of all the income probabilities must equal the sum of all the outcome probabilities. This is a result of the definition of the probabilities. The sum of all the income probabilities (of a node) represents how likely the patroller will be at the node, while the sum of all the outcome probabilities represents the probability that the patroller would take an action (either goes to adjacent plants or patrols the current plant) at the node. Another property of probabilities c_{s-e} is that, the sum of probabilities coming out from the root node equals 1. The idea behind this property is that, the patroller deterministically (since a probability of 1) starts from the root node, and then she chooses to go to the next step. Formulas (4) and (5) illustrate the abovementioned two properties.

$$sP_{pv} = \sum_{in\in\{s \in pV|(s,pv) \in pE\}} c_{in-pv} = \sum_{out\in\{e \in pV|(pv,e) \in pE\}} c_{pv-out} \cdot (4)$$

$$\sum_{out\in\{e \in pV|(root,e) \in pE\}} c_{root-out} = 1 \cdots\cdots (5)$$

Furthermore, in patrolling practice, when the defender is already situated at node pv (of pG), her conditional probability of choosing a specific action (i.e., an edge in pG) can be calculated by Formula (6). For instance, if a purely randomized patrolling strategy would be implemented on the patrolling graph shown in Figure 3, then the probability that the patroller will be at node 2 $(3,' B2')$ is $sP_2 = 1/3$, and the probabilities that the patroller goes to node 9, 10, and 11 are all $c_{2-9} = c_{2-10} = c_{2-11} = 1/9$. Therefore, we have $cP_2^9 = cP_2^{10} = cP_2^{11} =$

$1/3$, and this result means that at node 2, the patroller takes each action at the same probability. Figure 5 in the case study section also illustrates how Formula (6) works.

$$cP_{pv}^{out} = \frac{c_{pv-out}}{sP_{pv}}, \text{ for all } out \in \{v \in pV | (pv, v) \in pE\} \quad \cdots\cdots(6)$$

3.3 Payoffs

There are two possible results in the CCP game, being: (i) the attack fails, either stopped by the multiple-plant patroller team or by the countermeasures in the target plant and (ii) the attack is successfully implemented. If the attack fails, the patroller gets a reward R^d (e.g., obtaining a bonus) and the attacker suffers a penalty P^a (e.g., being sent to prison). If the attacker succeeds, the patroller suffers a loss L^d and the attacker obtains a gain G^a.

R^d is a number decided by the chemical cluster council. For instance, the cluster rewards $1k€$ to the defender (consists of the patroller and the plant's own security department). P^a is scenario-related since different attack scenarios need different attack costs and the attacker, if being caught, will also be punished differently. L^d and G^a are determined by both the attack scenario and the target plant. All these parameters should be evaluated by security experts, for instance, by a API SRA team [35].

Formulas (7) and (8) further define the patroller and the attacker's payoff, in which f ($\tilde{f}$) is the probability that the attack would fail, from the defender's (the attacker's) perspective.

$$u_d = R^d \cdot f - L^d \cdot (1 - f) \quad \cdots\cdots(7)$$

$$u_a = G^a \cdot (1 - \tilde{f}) - P^a \cdot \tilde{f} \quad \cdots\cdots(8)$$

In the following paragraphs, we focus on calculating the probability f ($\tilde{f}$) that the attacker would be detected, under the condition that the attacker plays a strategy (t, i, k_i) and the defender plays $\vec{c}$ (a vector whose entries are the c_{i-j} in Formula (3)).

We denote the probability that the security countermeasures in the target plant would detect the attacker as f_{cpp}, which can be calculated by the Chemical Plant Protection game [36] or which can be evaluated by a security assessment team as well [35, 37]. Furthermore, we represent the probability that the patroller would detect the attacker as f_p. Considering that the attacker can be detected either by the countermeasures of the target plant or by the patroller team, the probability that the attacker would be detected can be calculated by Formula (9):

$$f = 1 - (1 - f_{cpp}) \cdot (1 - f_p) \quad \cdots\cdots (9)$$

Note that f_{cpp} is a plant-specific parameter (a real number belonging to [0,1]). We focus on calculating f_p. An intrusion and attack procedure in plant i lasts for k_i time slices, while patrolling in the plant lasts for t_i^p time units. If there are any overlaps between the intrusion and attack procedure and the patroller's staying in the plant, then there is a probability that the attacker would be detected by the patroller. Otherwise, the adversary would only be possibly detected by the countermeasures of the target plant, i.e., $f_p = 0$. Theoretically speaking, the longer the overlap is, the higher the f_p would be.

Furthermore, which time period of the intrusion and attack procedure is covered by the overlap also influences the probability. For instance, the intruder can easier be noticed by the patroller team at the beginning of his intrusion procedure since at this time, he is moving into the plant. After reaching the target, it may be difficult for a patroller to detect the attacker. For instance, if his target is inside a room, then the patroller would not be able to detect him at all. The situation can also be opposite.

Therefore, in order to calculate f_p, not only the length of the overlap should be calculated, but also which part of the intrusion and attack procedure is covered should also be identified. The overlap of the patroller's staying in plant i and the attacker's intrusion and attack procedure in

plant i can be calculated by Formula (10), in which st denotes the start time that the patroller stays in plant i. There are two situations of the exact overlap period.

$$Overlap = [max\{t, st\}, min\{t + k_i, st + t_i^p\}] \cdots\cdots(10)$$

Situation 1: if $t_i^p \leq k_i$, then there are $k_i + t_i^p - 1$ possible overlap situations. Each of the situation covers the intrusion and attack procedure at time $[t, t+1]$, $[t, t+2]$, …, $[t, t + t_i^p]$, $[t+1, t + t_i^p + 1]$, …, $[t + k_i - t_i^p, t + k_i]$, $[t + k_i - t_i^p + 1, t + k_i]$, $[t + k_i - t_i^p + 2, t + k_i]$, …, $[t + k_i - 1, t + k_i]$, respectively. Figure 4 shows an example of the overlap situations with $k_i = 5$, $t_i^p = 2$. In Figure 4, the horizontal line denotes the intrusion and attack procedure which lasts for 5 time slices, while the red dot line means the overlap with the patroller's staying in the plant.

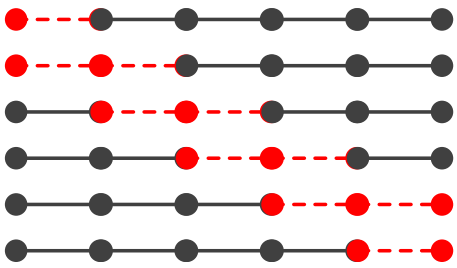

Figure 4. An illustrative figure of the overlap situation

Situation 2: if $t_i^p > k_i$, then there are $k_i + k_i$*-1* possible overlap situations. Each of the situations cover the intrusion and attack procedure at time $[t, t+1]$, $[t, t+2]$, …, $[t, t + k_i]$, $[t+1, t + k_i]$, …, $[t + k_i - 1, t + k_i]$, respectively.

For each of the possible overlap cases, define a detection probability σ_r and $r = 1,2, \dots, t_i^p + k_i - 1$ in situation 1 and $r = 1,2, \dots, k_i + k_i - 1$ in situation 2. Furthermore, denote the probability that the patroller would be in situation r as τ_r. The probability that the attacker

would be detected by the patroller can then be calculated by Formula (11). σ_r are user inputs and should be provided by security experts.

$$f_p = \sum_r \sigma_r \cdot \tau_r. \quad (11)$$

Table 4 shows how to calculate τ_r, under the condition of an attacker strategy (t, i, k_i) and a defender strategy $\vec{c}$.

Table 4. The procedure of calculating τ_r

Calculating τ_r
1. Initialize $\tau_r = 0$.
2. If an edge $pe \in pE$ in the patrolling graph pG satisfies Condition 1 and Condition 2, then $\tau_r = \tau_r + c_{pe}$, in which c_{pe} is the weight (the probability) of the edge.

Denote the start and end node of an edge (of pG) as $sn = (snt, sni)$ and $en = (ent, eni)$ respectively, and define:

Condition 1: both the corresponding entrances of node sni and eni belong to plant i, the attacker's target. For instance, in the illustrative example shown in Figure 1 and 2, if the target plant is 'A', and $sni = eni = 'A'$, then condition 1 holds; or if the target plant is 'B' and $sni =' B1'$ and $eni =' B2'$, then condition 1 holds as well.

Condition 2: the overlap (in time dimension) of the edge and the attacker strategy satisfies situation r. Rigorously, $[snt, ent] \cap [t, t + k_i]$ equals the corresponding time zone of the overlap situation r. Figure 5 and Table 7 in the case study section illustrate this condition.

In condition 1, if $sni = eni$, the edge would be a horizontal line when shown in a figure like Figure 3, and it indicates that a patrolling team comes in and out the same gate of the plant, otherwise if $sni \neq eni$ but both of them belong to the same plant, it denotes a patrolling comes in and out from different entrances of the plant.

In condition 2, if $t + k_i > T$, then edges satisfying the condition that $[snt, ent] \cap [0, t + k_i - T]$ equals the corresponding time zone of the overlap situation r, are also said to fulfil condition 2. This results from the way that we deal with the periodic patrolling problem. When time exceeds T, the next patrolling team has already started her patrolling, therefore the attacker not only can be detected by the current patroller, but also can be detected by the next patrolling team.

It is worth noting that τ_r is a linear polynomial of $\vec{c}$, denoted as $\tau_r = Coe_r \cdot \vec{c}^T$, and f_{cpp} and σ_r are user provided parameters. Therefore, f is a linear polynomial of $\vec{c}$ as well. Furthermore, the definitions of f, u_d, u_a can be rewritten as:

$$f = \left[\sum_r \left(1 - f_{cpp}\right) \cdot \sigma_r \cdot Coe_r, f_{cpp}\right] \cdot [\vec{c}, 1]^T \quad \cdots\cdots(12)$$

$$u_d = [(R^d + L^d) \cdot \left(\sum_r \left(1 - f_{cpp}\right) \cdot \sigma_r \cdot Coe_r\right), (R^d + L^d) \cdot f_{cpp} - L^d] \cdot [\vec{c}, 1]^T \cdots\cdots(13)$$

$$u_a = [-(G^a + P^a) \cdot \left(\sum_r \left(1 - \tilde{f}_{cpp}\right) \cdot \sigma_r \cdot Coe_r\right), G^a - (G^a + P^a) \cdot \tilde{f}_{cpp}] \cdot [\vec{c}, 1]^T \cdots(14)$$

3.4 Solutions for the game

3.4.1 Stackelberg equilibrium

In the Chemical Cluster Patrolling (CCP) game, the attacker is assumed to be able to collect information about the patroller's patrolling route. For instance, as already mentioned, the attacker may achieve this goal by long term observation or by stealing the patroller's security plan. Therefore, we assume that the CCP game is played sequentially. The patroller (being the game leader) firstly commits a patrolling strategy $\vec{c}$, and subsequently, the attacker moves optimally according to the defender's strategy (being the game follower). The patroller could also work out the attacker's optimal solution, thus she can arrange her strategy $\vec{c}$ optimally as well.

A Stackelberg equilibrium $(s_d^*, s_a^*) = (\overrightarrow{c^*}, (t^*, i^*, k_i^*))$ for the CCP game is a patroller-attacker strategy pair that satisfies the following condition:

$$(t^*, i^*, k_i^*) = \underset{(t,i,k_i)\in S_a}{\operatorname{argmax}}\{u_a(\vec{c}, (t, i, k_i))\} \quad \cdots\cdots(15)$$

$$\overrightarrow{c^*} = \underset{\vec{c}\in S_d}{\operatorname{argmax}}\{u_d(\vec{c}, (t^*, i^*, k_i^*))\} \quad \cdots\cdots(16)$$

Formula (15) indicates that observing the defender's strategy $\vec{c}$, the attacker would play a strategy which maximizes his own payoff (i.e., a best response strategy). Formula (16) represents that the defender can also work out the attacker's best response to her strategy, thus she plays accordingly.

By discretizing the time dimension (in section 2.2.3), the attacker has a finite number of strategies. Moreover, Formulas (13) and (14) show that for a given attacker strategy, payoff functions u_a and u_d would both be linear polynomials of $\vec{c}$. Therefore, a multiple linear programming algorithm [38] can be introduced to compute the Stackelberg equilibrium for the CCP game, as shown in Table 5.

Table 5. MultiLPs algorithm for computing the Stackelberg equilibrium for the CCP game

MultiLPs
○ *Initialization* for each attacker strategy (t, i, k_i), calculate u_a and u_d, which are linear polynomials of $\vec{c}$;
○ *Linear Programming (LP)* suppose that the attacker strategy $(t^\#, i^\#, k_i^\#)$ is the attacker's best response, which means: $u_a(t^\#, i^\#, k_i^\#, \vec{c}) \geq u_a(t, i, k_i, \vec{c}), \quad \forall (t, i, k_i) \in S_a$ (17) The defender would then aims at: $Pof_d\left(t^\#, i^\#, k_i^\#, \overrightarrow{c^\#}\right) = \max_{\vec{c}\in S_d} u_d(t^\#, i^\#, k_i^\#, \vec{c})$ (18)
○ *Summary* The Stackelberg equilibrium $(\overrightarrow{c^*}, (t^*, i^*, k_i^*)) = arg \max_{(t^\#, i^\#, k_i^\#)\in S_a} Pof_d\left(t^\#, i^\#, k_i^\#, \overrightarrow{c^\#}\right)$.

In the linear programming step, the defender is solving a linear programming problem. The cost function of the linear programming is Formula (18) and the constraints are Formulas (17), (4) and (5). Furthermore, the LP step should be implemented for each attacker strategy. In the linear programming step, if we further constraint c_{s-e} to be either 0 or 1, then the MultiLPs algorithm would output the optimal fixed patrolling route for the patroller.

The Stackelberg equilibrium calculated by the MultiLPs algorithm is a Strong Stackelberg Equilibrium [39], and it is therefore based on the "breaking-tie" assumption[2]. By running again the LP step in the MultiLPs algorithm, and supposing that (t^*, i^*, k_i^*) is the attacker's best response as well as revising Formula (17) to be Formula (19), in which α is a constant small positive number, the Strong Stackelberg Equilibrium will be slightly modified, resulting in a Modified Stackelberg Equilibrium which does not rely on the "breaking-tie" assumption and is still optimal enough [39].

$$u_a(t^*, i^*, k_i^*, \vec{c}) \geq \alpha + u_a(t, i, k_i, \vec{c}), \quad \forall (t, i, k_i) \in S_a \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (19)$$

3.4.2 Robust solution considering distribution-free uncertainties

The Stackelberg Equilibrium can be calculated for the CCP game only in case that the patroller knows the exact numbers of all the parameters (shown in Table 1) of the game. In security practice, the patroller may obtain some of these parameters by using conventional security risk assessment methods such as the API SRA [35]. However, there are at least two parameters of which the values are difficult to obtain: the attacker's gain from a successful attack G^a and the attacker's estimation of being detected by the intrusion detection system of each plant $\tilde{f}_{cpp}$. Therefore, similar to Zhang et al. [40], we assume that the patroller can obtain an interval of these two parameters and how these two parameters distribute in the interval

[2] The 'breaking-tie' assumption in the Strong Stackelberg Equilibrium requires that, when the game follower (i.e., the attacker in the CCP game) is indifferent on payoffs by playing different pure strategies (i.e., he faces a tie), he will play the strategy that is preferable for the game leader (i.e., the patroller in the CCP game).

zones are not known. Further assume that $G^a \in [G^{a\_min}, G^{a\_max}]$ and $\tilde{f}_{cpp} \in [\tilde{f}_{cpp}^{min}, \tilde{f}_{cpp}^{max}]$ and therefore the patroller can have the lower and upper bound of the attacker's payoff, as shown in Formulas (20) and (21) respectively. Note that Formula (8) demonstrates that u_a is monotonically increasing on G^a and monotonically decreasing on f. Formula (9) demonstrates that f is monotonically increasing on f_{cpp}.

$$u_a^{min} = [-\left(G^{a\_min} + P^a\right) \cdot \left(\sum_r \left(1 - \tilde{f}_{cpp}^{max}\right) \cdot \sigma_r \cdot Coe_r\right), G^{a\_min} - \left(G^{a\_min} + P^a\right) \cdot \tilde{f}_{cpp}^{max}] \cdot [\vec{c}, 1]^T \quad (20)$$

$$u_a^{max} = [-(G^{a\_max} + P^a) \cdot \left(\sum_r \left(1 - \tilde{f}_{cpp}^{min}\right) \cdot \sigma_r \cdot Coe_r\right), G^{a\_max} - (G^{a\_max} + P^a) \cdot \tilde{f}_{cpp}^{min}] \cdot [\vec{c}, 1]^T \quad (21)$$

Knowing the lower and upper bound of the attacker's payoff, the patroller can play the game as follows: (i) she commits to a patrolling strategy c; (ii) she works out the attacker's lower and upper bound payoffs in the case that the attacker responds with different pure strategies to c; (iii) she gets the attacker's highest lower bound payoff R; (iv) she picks out all the attacker's possible best responses, which are, the attacker's pure strategies that have higher upper bound payoffs than R; (v) among all the attacker's possible best responses, assume that the one that is worst to the patroller is the attacker's real best response and the patroller then optimizes c accordingly.

Furthermore, if two pure strategies of the attacker (e.g., s_{a1} and s_{a2}) have the same target plant, then the attacker's payoffs by responding these two pure strategies will share the same G^a and $\tilde{f}_{cpp}$ and therefore the payoffs (of responding these two strategies) will be correlated. In this situation, we have that $u_a(s_{a1}, c) \geq u_a(s_{a2}, c) \Leftrightarrow f_p(s_{a1}, c) \leq f_p(s_{a2}, c)$ and vice versa.

Formula (22) illustrates an algorithm for calculating the patroller's robust solution considering her distribution-free uncertainties on the attacker's parameters. In Formula (22), the variables are c, q, R and γ, which denote the patroller's patrolling strategy, indication of the attacker's possible best response strategy, the attacker's highest lower bound payoff, and the defender's optimal payoff, respectively.

$$\underset{c,q,R,\gamma}{\text{maximize}}\ \gamma$$

$$s.t.\begin{cases} c1. & NstFlwLef \cdot c = NstFlwRgt \\ c2. & R = u_a^{\min}(J,c) \\ c3. & R \geq u_a^{\min}(j,c), \forall j \in \{1,2,...,\mathrm{m}\} \\ c4. & -q_j \cdot \Gamma \leq R - u_a^{\max}(j,c) \leq (1-\mathrm{q}_j)\cdot\Gamma, \forall j \in \{1,2,...,\mathrm{m}\} - Plt_J \\ c5. & -q_j \cdot \Gamma \leq f_p(j,c) - f_p(J,c) \leq (1-\mathrm{q}_j)\cdot\Gamma, \forall j \in Plt_J \\ c6. & (1-q_j)\cdot\Gamma + u_d(j,c) \geq \gamma, \forall j \in \{1,2,...,\mathrm{m}\} \\ c7. & q_j \in \{0,1\}, c_i \in [0,1], \forall j \in \{1,2,...,\mathrm{m}\}, \forall i \in \{1,2,...,n\} \end{cases} \quad \cdots\cdots\cdots\cdots\cdots (22)$$

Constraint $c1$ reflects the features of the patroller's strategy c, as explained in Formula (4) and (5). Constraints $c2$ calculates the attacker's lower bound payoff R by playing strategy J. $c3$ ensures that strategy J has the highest lower bound payoff, among all the attacker's pure strategies. Constraints $c4$ and $c5$ pick out all the attacker's possible best responses. Plt_J denotes all the attacker's strategies that have the same target plant with strategy J. Note that in these two constraints, if $u_a^{max}(j,c) > R$ or $f_p(j,c) < f_p(J,c)$, then $q_j = 1$, and vice versa. Therefore $q_j = 1$ indicates that strategy j is in the attacker's possible best response set. Constraint $c6$ represents the patroller conservatively thinking that among all the attacker's possible best responses, the one that is the worst to her is the attacker's real best response. The cost function further represents the patroller optimizing her payoff.

In Formula (22), the attacker's strategy J is assumed to have the highest lower bound payoff. Therefore, the optimal solution and payoffs generated by the formula are conditional. By

implementing Formula (22) for m times and each time setting a different J, we obtain a result, denoting as $rlt^J = (c^J, q^J, R^J, \gamma^J)$. If Formula (22) is not feasible for a certain J, then we set $rlt^J = (null, null, -inf, -inf)$. Finally, we pick out the rlt^J that has a highest γ^J as the final robust solution of the game.

4. Case study

4.1 Case study setting

The layout of the cluster, the graph model, and the patrolling graph model of the case study are given in Figure 1 through 3. The total patrolling time T is set as 30 time slices. The patroller's driving time between different plants and patrolling time in each plant are shown in Table 2. Some more parameters and simplification assumptions of the case study are given hereafter.

For the sake of simplicity, we assume that the attacker has only one attack scenario and this scenario lasts for ten time slices in each plant. Table 6 gives the model inputs, i.e., the defender's reward (R^d) and loss (L^d) of detecting and not detecting an attacker; the attacker's gain (G^a) and penalty (P^a) from a successful and from a failed attack; the probability (f_{cpp}) that countermeasures in each plant can detect the attacker. The probability that the patroller can detect an attacker (i.e., σ_r, definition given in Figure 4) should also be provided by security experts. However, for the sake of simplicity, we assume that in each time slice, if the attacker and the patroller stay in the same plant (i.e., an overlap situation), there is a probability of 0.05 that the attacker would be detected by the patroller. The unit of all the monetary parameters can be, for instance, k€.

Table 6. Further model inputs for the case study of CCP game

	R^d	L^d	G^a	G^{a_min}	G^{a_max}	P^a	f_{cpp} & $\tilde{f}_{cpp}$	$\tilde{f}_{cpp}^{min}$	$\tilde{f}_{cpp}^{max}$
'A'	1	16	10	9.5	10.2	3	0.45	0.44	0.46
'B'	1	11.2	6	5.5	6.4	3	0.3	0.29	0.31
'C'	1	14	8.3	8	8.5	3	0.42	0.41	0.43
'D'	1	12	7.1	7	7.4	3	0.45	0.44	0.46
'E'	1	15	10	9.5	10.3	3	0.5	0.49	0.51

It is worth noting that all these data concern estimations from the patroller. Therefore, the numbers of rewards (R^d), losses (L^d), and the detection probability (f_{cpp}) of countermeasures of each plant are the patroller's estimation of her own data. The amounts of the attacker's gains (G^a), penalties (P^a), and the attacker's estimation of the detection probability ($\tilde{f}_{cpp}$) of countermeasures of each plant, are the patroller's estimation of the attacker's data. For instance, "the gain of a successful attack on plant 'A' is 10" means that the patroller thinks the attacker will receive a value of 10 from this attack. The patroller may have uncertainties on guessing the attacker's parameters. Therefore, G^{a_min} and G^{a_max} are introduced to denote the patroller's minimal and maximal guesses of the attacker's gain of a successful attack. Similarly, $\tilde{f}_{cpp}^{min}$ and $\tilde{f}_{cpp}^{max}$ denote the patroller's minimal and maximal guesses of the attacker's estimation of the detection probability of countermeasures of every plant. The attacker's penalty of a failed attack is easier to estimate. Therefore we assume that the patroller can correctly guess the exact number of it.

4.2 Game modelling

There are two players in the case study game, namely the patroller and the attacker. Since only one attack scenario is considered, the attacker therefore has $m = 5 \times 30 \times 1 = 150$ pure strategies, being attack a plant (i.e., one of 'A', 'B', 'C', 'D', and 'E') at a time (i.e., at a time

$t \in \{0, 1, 2, \ldots, 29\}$). The patroller has 435 possible actions that she can take, shown as edges in Figure 3 and therefore the patroller's strategy can be represented as a vector of 435 entries.

According to Formulas (13), (14), (20), and (21), the attacker and the patroller's payoffs can be calculated. Payoffs will be represented as linear polynomials of the patroller's strategy (i.e., $\vec{c}$), while the attacker's strategy decides the coefficients of the polynomials.

4.3 CCP Game results

4.3.1 Stackelberg equilibrium

Figure 5 shows the modified Stackelberg Equilibrium (mSE) of the game developed for the case study, calculated by the MultiLPs algorithm shown in Table 5 and then slightly moved with a $\alpha = 0.1$. The black (and bold) lines demonstrate the patroller's optimal patrolling strategy. The associated number on the line demotes the probability that the defender will take this action. For instance, $c1 = 0.2275$ means that at time 0, the patroller should drive to node 'B2' at a probability of 0.2275.[3] Furthermore, in patrolling practice, if the patroller arrives at a node in the figure, the conditional probabilities of the following actions can be calculated by Formula (6). For instance, the probability that the patroller would arrive at the red node $(6, 'C')$ in Figure 5 is $sP_v = 0.4173$, and the conditional probabilities that the patroller should take the two actions (i.e., either patrolling in plant 'C' for a period of six time slices or driving to entrance 'B1' by a driving time of three time slices) are $cP_1 = \frac{0.2078}{0.4173} = 0.4979, cP_2 = \frac{0.2096}{0.4173} = 0.5021$ respectively.

The attacker's best response strategy in the mSE is to attack plant 'E' at time 9, shown in Figure 5 as a red bold line. The short blue lines above the attacker's best response strategy line (i.e., the red bold line) represent the defender's patrolling actions which have a

[3] In this paper, all the results are rounded to their ten-thousandth.

probability of being taken (i.e., $c > 0$) and would have overlap with the attacker's best response strategy. Table 7 shows the detail information of the defender's actions that have overlaps with the attacker's best response strategy. The 'Edge' column denotes the edge index (in Figure 3) of the action. The c column shows the probability that the actions would be taken in the mSE, and these numbers are also shown in Figure 5. The 'Overlap' column illustrates the time period that the actions overlap with the attacker's best response strategy. The 'σ' column provides the probability that the attacker would be detected by the corresponding action, and this probability is simply calculated as 0.05 multiplied by the overlapping time slices. For instance, edge 25 represents the patroller's action of patrolling plant 'E' from time 6 until time 13 while the attacker starts his attack in plant 'E' at time 9. Therefore, edge 25 overlaps with the attacker's attack in time zone [9,13], and the σ is $0.05 \times (13 - 9) = 0.20$.

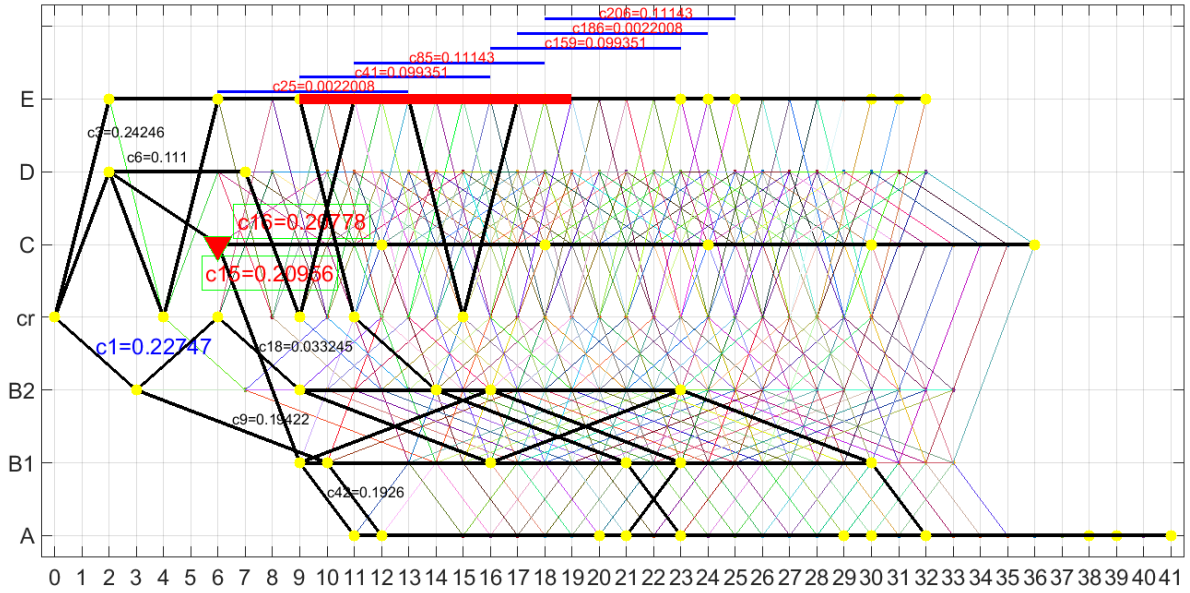

Figure 5. The optimal patrolling strategy and the attacker's best response

Based on the results in Table 7, recalling Formula (11) and the τ_r calculation algorithm, we have that:

$f_p = \sum_r \tau_r \cdot \sigma_r = 0.0891$

$f = 1 - (1 - 0.5) * \left(1 - f_p\right) = 0.0949,$

$u_a = 2.88311 \; and \; u_d = -6.2407.$

Table 7. The patroller's actions that may detect the attacker

Edge	c	Overlap	σ
25	0.0022	[9,13]	0.20
41	0.0994	[9,16]	0.35
85	0.1114	[11,18]	0.35
159	0.0994	[16,19]	0.15
186	0.0022	[17,19]	0.10
206	0.1114	[18,19]	0.05

Let us now compare the modified Stackelberg Equilibrium with the purely randomized patrolling strategy. In current patrolling practice, patrollers may randomly schedule their patrolling route. This situation, as demonstrated in Figure 3, is simply assigning equal probabilities to edges that start from the same node. For instance, at the starting node (i.e., $(0, 'cr')$), the patroller would come to plant (entrance) 'B2', 'D', and 'E' with the same probability, being $1/3$.

Table 8. Comparison of the CCP mSE strategy and the purely randomized strategy

Edge	Overlap	c	rc	σ
82	[11,19]	0.1926	0.0046	0.4
98	[12,19]	0.1942	0.0139	0.35
156	[15,19]	0	0.0019	0.2
176	[16,19]	0	0.0071	0.15
196	[17,19]	0	0.0024	0.1
216	[18,19]	0	0.0039	0.05
425	[9,10]	0	0.0100	0.05
430	[9,11]	0.3358	0.0274	0.1

In the case study, if the defender would purely randomize her patrolling, then the attacker's best response would be attacking plant 'A' at time 9. The attacker and the defender would obtain a payoff of 4.0653 and -8.2393, respectively. Compared to the Modified Stackelberg Equilibrium of the CCP game, the defender's payoff reduces from -6.2407 to -8.2393.

Table 8 illustrates the differences between the CCP mSE strategy and the purely randomized strategy. The edge column shows the edges in the patrolling graph showing an overlap with the attacker's best response strategy to the defender's purely randomized strategy (i.e., attack plant 'A' at time 9). The overlap column shows the period of the attack procedure being overlapped by the edge. The 'c' and 'rc' columns show the probability that the patroller will follow the edge, resulting from the CCP mSE strategy and from the purely randomized strategy, respectively. The 'σ' column shows the probability that the attacker will be detected by the patroller by the action she undertakes, represented by this edge.

With the results in Table 8, the probability that the attacker would be detected can be calculated, being $f_p^c = 0.1786$ and $f_p^{rc} = 0.0118$, for the defender's CCP mSE strategy and for the defender's purely randomized strategy, respectively. This result reveals that the CCP mSE strategy is characterized with a higher probability that the attacker is detected at plant 'A', and thus enforces the attacker to attack plant 'E' instead of attacking plant 'A'.

Furthermore, in current patrolling practice, some patrollers may follow a fixed patrolling route. In the patrolling graph, if we further constraint the probability that an action (an edge) is taken to be either 0 or 1, that is, $c \in \{0,1\}$ instead of $c \in [0,1]$, then a vector of c that satisfies Formulas (4) and (5), represents a fixed patrolling route. The bold route shown in Figure 6 is the optimal fixed patrolling route considering intelligent attackers. The route is that: the patroller starts from 'cr'; she goes to plant 'D' and patrols plant 'D'; after then, she goes to plant 'A' and patrols 'A'; she further goes to entrance 'B1' and then comes back to

plant ‘A’ and patrols plant ‘A’. The red dot line in Figure 6 denotes the attacker’s best response strategy to the optimal fixed patrolling route, and it is, attacking plant ‘C’ at time 21. If the defender follows the fixed patrolling route and the attacker plays his best response, as shown in Figure 6, the payoffs for the defender and for the attacker are -7.7 and 3.5540 respectively.

It is worth noting the defender’s optimal fixed patrolling route is not unique and the attacker’s best response is not unique as well. For instance, knowing the patroller’s fixed route, the attacker would be indifferent by starting his attack at any time. However, the defender and the attacker’s payoff would not be different. Therefore, here we only show one optimal fixed patrolling route and one attacker’s best response strategy.

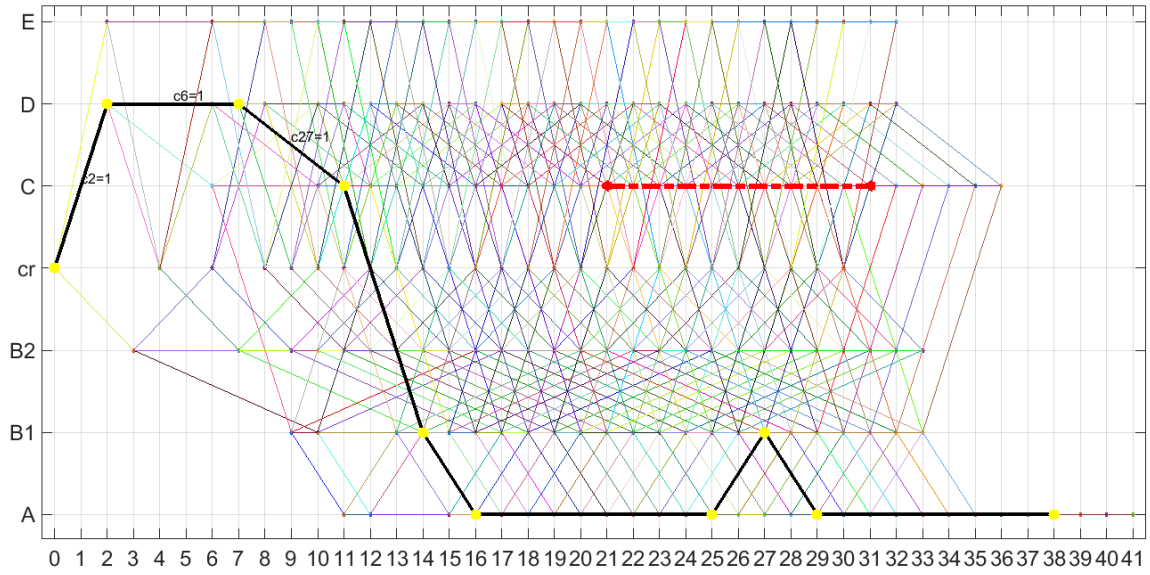

Figure 6. The patroller’s optimal fixed patrolling route and the attacker’s best response

4.3.2 Robust equilibrium

Figure 7 shows the robust solution of the Interval Chemical Cluster Patrolling game, based on the input data from Table 6. Notations of Figure 7 are the same as defined in Figure 5. The attacker’s strategy of attacking plant ‘E’ at time 0 has the highest lower bound payoff, shown as a red bold line in Figure 7. Furthermore we have:

$$f_p = 0.10805 \cdot 0.35 + 0.06043 \cdot 0.05 + 0.00751 \cdot 0.05 + 0.03415 \cdot 0.10 = 0.0446$$

$$f = 1 - \left(1 - \tilde{f}_{cpp}^{max}\right) \cdot \left(1 - f_p\right) = 0.5319$$

$$\mathrm{R} = G^{a\_min} \cdot (1 - f) - P^a \cdot f = 2.8516$$

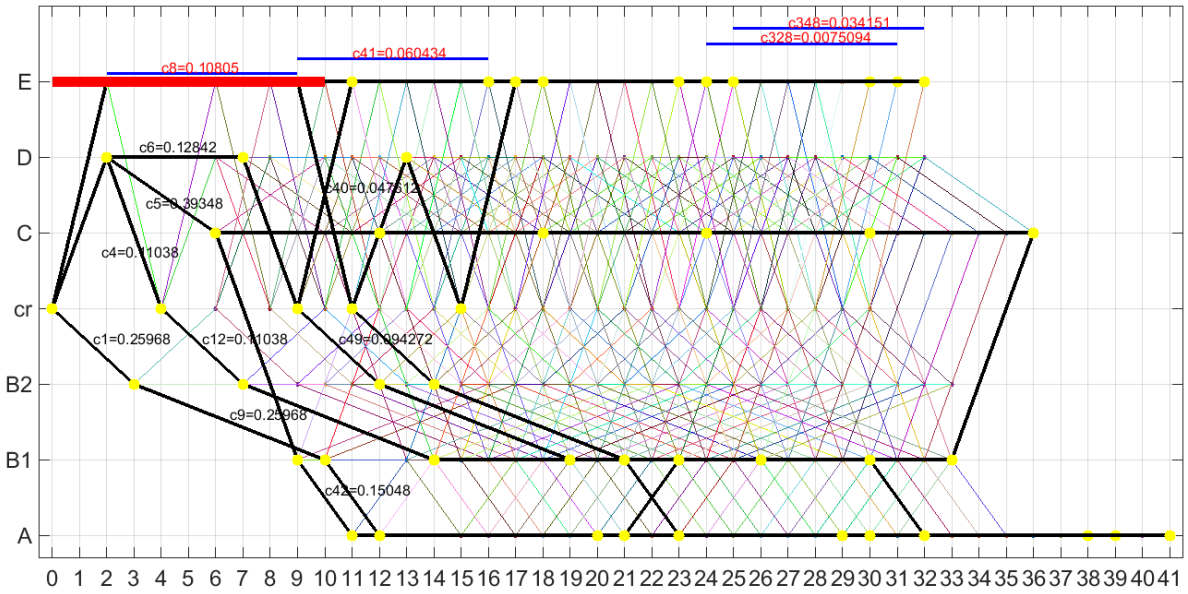

Figure 7. Robust solution of the interval CCP game

Figure 8 shows the attacker's payoff information of the robust solution of the Interval CCP game. As also demonstrated in the figure, different sub-figures denote the attacker's payoff by attacking different plants. The x-axis denotes the start time of attacks and therefore a combination of an x coordinate and a certain sub-figure represents an attacker strategy. The vertical lines denote the range of the patroller's estimation of the attacker's payoffs, under the conditions that the patroller plays her strategy and the attacker plays the corresponding strategy (i.e., the sub-figure and the x coordinate). Horizontal lines in all sub-figures have the same y value, and it is the attacker's highest lower bound payoff (i.e., R). A red square dot means that the corresponding attacker strategy is the attacker's possible best response strategy while a green circle dot means that the corresponding strategy is not a possible best response strategy for the attacker.

As shown in Figure 8, for an attacker strategy, if the attack target is not plant 'E' and, if the strategy has an upper bound payoff higher than R, then the attacker strategy is thought to be a possible best response for the attacker (i.e., a red square is used), otherwise if the strategy has an upper bound payoff lower than R, then it is considered not to be a possible best response (i.e., a green dot is used). If an attacker strategy aims to attack plant 'E', then the above rule does not work, as shown in sub-figure 'Plant E'. The reason is that, the robust solution is achieved when the attacker plays a strategy of attacking plant 'E' at time 0. Therefore, whether strategies which aim at attacking plant 'E' should be possible best response strategies will determined by constraint c5 in Formula (22), instead of by the payoff range constraint (i.e., Constraint c4 in Formula (22)).

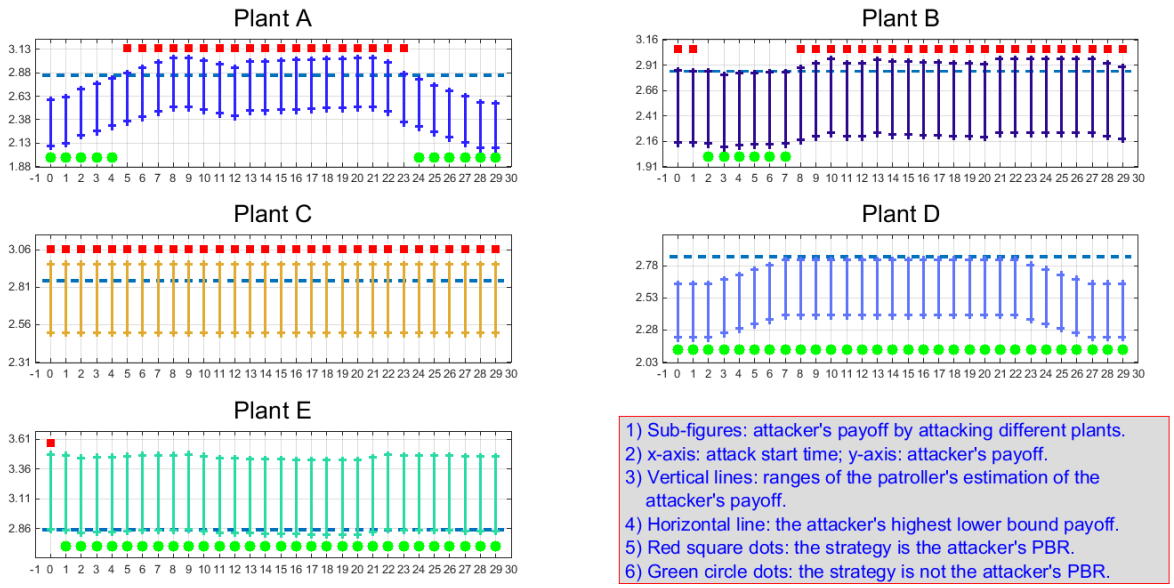

Figure 8. Attacker payoff information of the robust solution of the Interval CCP game (PBR: possible best response)

5. Discussion on the implementation errors and observation errors

Besides the defender's uncertainties on the attacker's parameters, there are other two types of uncertainties, namely, the patroller's implementation error and the attacker's observation error.

In reality, the patroller would always have errors while implementing her patrolling strategy. For instance, the patroller may have to go to the toilet or she has to deal with some detected security issues. Therefore, to make the patrolling strategies generated by the CCP game more robust, we can assume that the real patrolling strategy c^{real} may deviate slightly from the planned strategy c, that is, $c^{real} \in [c-\epsilon, c+\epsilon] \cap [0,1]$, in which ϵ is a small positive number denoting the tolerance of the implementation error.

The attacker's observation error of the patroller's implemented strategy can be modelled in two different approaches. The first approach is similar to the modelling of the patroller's implementation error by introducing a small positive number δ, denoting the error between the attacker's observation and the defender's implemented strategy. Subsequently, we have $c^{obs} \in [c^{real}-\delta, c^{real}+\delta] \cap [0,1]$. The second approach is by employing the anchoring theory. The anchoring theory says that when there is no external information about a set of discrete events, humans assume that the occurrence probability of each event is the same. When further information is provided (e.g., the attacker observes the patroller's daily patrolling), humans are able to calibrate their estimation of probability of each event to the real probability. In the CCP game, this procedure can be described as $c^{obs} = (1-\beta) \cdot c^{PureRandom} + \beta \cdot c^{real}$, in which β denotes the observation ability of the attacker and $c^{PureRandom}$ denotes a purely randomized patrolling strategy.

For integrating these two types of uncertainties to the CCP game, the algorithm proposed by Nguyen et al. [41] can be employed. However, the algorithm in Nguyen et al. [41] has a very high computational complexity if being applied on the CCP game. Therefore, developing a quicker and more efficient algorithm for dealing with these two types of uncertainties in the CCP game can be a fruitful future research.

6. Conclusion

Terrorism is a global problem. Geographically clustered chemical plants throughout the world can be quite interesting targets for terrorists, due to the possibility of inducing domino effects. Besides the countermeasures that each plant takes, and that a multi-plant council may take, also security patrolling at the cluster level is recommended. To this end, a so-called chemical cluster patrolling game (CCP game) is developed and proposed in this paper. The game is played by the patroller and the potential attackers, taking into account intelligent interactions between them. Two solution concepts, namely the Stackelberg Equilibrium and the robust solution, are put forward.

Results of the case study show that by strategically randomizing patrolling routes, the patroller would have higher expected payoffs, indicating that patrolling more hazardous plants would be more likely (that is, they are accompanied by higher probabilities for the patroller). Performance of the patrolling strategy from the Stackelberg equilibrium overcomes the performance of the purely randomized patrolling routes and the performances of any fixed patrolling routes.

The CCP game can be further investigated from several aspects. Firstly, the current model only allows a fixed patrolling time in a plant. In reality, the patroller may also patrol the same plant with different intensity, resulting in different patrolling time in the plant. Secondly, more robust solutions should be studied. For instance, the patroller can be difficult to perfectly follow the optimal patrolling strategy and an implementation error can occur. Thirdly, the attacker is assumed only knowing the probabilities that the patroller would take each action (i.e., $\vec{c}$). A possible situation is that the attacker not only knows the probability, but also knows the current location of the patroller. To model this situation, a stochastic game might be employed [42].

Acknowledgements

This study is supported by China Scholarship Council, and partly by National Key Research & Development (R&D) Plan under Grant No. 2017YFC0803300 and the National Natural Science Foundation of China under Grant Nos. 71673292, 61503402.

Reference

[1] Baybutt P. Strategies for protecting process plants against terrorism, sabotage and other criminal acts. Homeland Defence Journal. 2003;2:1-7.

[2] Baybutt P. Issues for security risk assessment in the process industries. J Loss Prev Process Ind. 2017;49(Part B):509-18.

[3] Baybutt P. Assessing risks from threats to process plants: Threat and vulnerability analysis. Process Saf Prog. 2002;21(4):269-75.

[4] Baybutt P. An Asset-based Approach For Industrial Cyber Security Vulnerability Analysis. Process Saf Prog. 2003;22(4):220-92.

[5] Baybutt P. Cyber security risk analysis for process control systems using rings of protection analysis (ROPA). Process Saf Prog. 2004;23(4):284-91.

[6] Bajpai S, Gupta J. Site security for chemical process industries. J Loss Prev Process Ind. 2005;18(4):301-9.

[7] Bajpai S, Gupta J. Securing oil and gas infrastructure. Journal of Petroleum Science and Engineering. 2007;55(1-2):174-86.

[8] Bajpai S, Sachdeva A, Gupta J. Security risk assessment: Applying the concepts of fuzzy logic. J Hazard Mater. 2010;173(1-3):258-64.

[9] Gupta J. The Bhopal gas tragedy: could it have happened in a developed country? J Loss Prev Process Ind. 2002;15(1):1-4.

[10] Reniers, Cremer, Buytaert. Continuously and simultaneously optimizing an organization's safety and security culture and climate: the Improvement Diamond for Excellence Achievement and Leadership in Safety & Security (IDEAL S&S) model. J Clean Prod. 2011;19(11):1239-49.

[11] Reniers G, Dullaert W. TePiTri: A screening method for assessing terrorist-related pipeline transport risks. Secur J. 2012;25(2):173-86.

[12] Reniers G, Herdewel D, Wybo JL. A threat assessment review planning (TARP) decision flowchart for complex industrial areas. J Loss Prev Process Ind. 2013;26(6):1662-9.

[13] Reniers G, Van Lerberghe P, Van Gulijk C. Security risk assessment and protection in the chemical and process industry. Process Saf Prog. 2015;34(1):72-83.

[14] Reniers GLL. Multi-Plant Safety and Security Management in the Chemical and Process Industries: Wiley-VCH; 2010.

[15] Reniers GLL, Sörensen K, Khan F, Amyotte P. Resilience of chemical industrial areas through attenuation-based security. Reliab Eng Syst Saf. 2014;131:94-101.

[16] Landucci G, Reniers G, Cozzani V, Salzano E. Vulnerability of industrial facilities to attacks with improvised explosive devices aimed at triggering domino scenarios. Reliab Eng Syst Saf. 2015;143:53-62.

[17] Argenti F, Landucci G, Cozzani V, Reniers G. A study on the performance assessment of anti-terrorism physical protection systems in chemical plants. Safety Science. 2017;94:181-96.

[18] Argenti F, Landucci G, Reniers G, Cozzani V. Vulnerability assessment of chemical facilities to intentional attacks based on Bayesian Network. Reliability Engineering & System Safety. 2018;169:515-30.

[19] Argenti F, Landucci G, Spadoni G, Cozzani V. The assessment of the attractiveness of process facilities to terrorist attacks. Safety Science. 2015;77:169-81.

[20] Landucci G, Argenti F, Cozzani V, Reniers G. Assessment of attack likelihood to support security risk assessment studies for chemical facilities. Process Saf Environ Prot. 2017.

[21] Khalil Y. A novel probabilistically timed dynamic model for physical security attack scenarios on critical infrastructures. Process Saf Environ Prot. 2016;102:473-84.

[22] Song G, Khan F, Yang M. Security Assessment of Process Facilities– Intrusion Modeling. Process Saf Environ Prot. 2018.

[23] van Staalduinen MA, Khan F, Gadag V. SVAPP methodology: A predictive security vulnerability assessment modeling method. J Loss Prev Process Ind. 2016;43:397-413.

[24] van Staalduinen MA, Khan F, Gadag V, Reniers G. Functional quantitative security risk analysis (QSRA) to assist in protecting critical process infrastructure. Reliability Engineering & System Safety. 2017;157:23-34.

[25] Fakhravar D, Khakzad N, Reniers G, Cozzani V. Security vulnerability assessment of gas pipelines using Discrete-time Bayesian network. Process Saf Environ Prot. 2017;111:714-25.

[26] Misuri A, Khakzad N, Reniers G, Cozzani V. A Bayesian network methodology for optimal security management of critical infrastructures. Reliability Engineering & System Safety. 2018.

[27] Reniers G, Pavlova Y. Using game theory to improve safety within chemical industrial parks: Springer; 2013.

[28] Gibbons R. A primer in game theory: Harvester Wheatsheaf; 1992.

[29] Shieh E, An B, Yang R, Tambe M, Baldwin C, DiRenzo J, et al., editors. Protect: A deployed game theoretic system to protect the ports of the united states. Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1; 2012: International Foundation for Autonomous Agents and Multiagent Systems.

[30] Fang F, Stone P, Tambe M, editors. When Security Games Go Green: Designing Defender Strategies to Prevent Poaching and Illegal Fishing. IJCAI; 2015.

[31] Rezazadeh A, Zhang L, Reniers G, Khakzad N, Cozzani V. Optimal patrol scheduling of hazardous pipelines using game theory. Process Saf Environ Prot. 2017;109:242-56.

[32] Alpern S, Morton A, Papadaki K. Patrolling games. Operations research. 2011;59(5):1246-57.

[33] Alpern S, Lidbetter T, Morton A, Papadaki K, editors. Patrolling a pipeline. International Conference on Decision and Game Theory for Security; 2016: Springer.

[34] Papadaki K, Alpern S, Lidbetter T, Morton A. Patrolling a border. Operations Research. 2016;64(6):1256-69.

[35] API. Security Risk Assessment Methodology for the Petroleum and Petrochemical Industries. In: 780 ARP, editor. 2013.

[36] Zhang L, Reniers G. A Game‐Theoretical Model to Improve Process Plant Protection from Terrorist Attacks. Risk Anal. 2016;36(12):2285-97.

[37] Zhang L, Reniers G, Chen B, Qiu X. Integrating the API SRA methodology and game theory for improving chemical plant protection. J Loss Prev Process Ind. 2018;51(Supplement C):8-16.

[38] Conitzer V, Sandholm T, editors. Computing the optimal strategy to commit to. Proceedings of the 7th ACM conference on Electronic commerce; 2006: ACM.

[39] Von Stengel B, Zamir S. Leadership with commitment to mixed strategies. 2004.

[40] Zhang L, Reniers G, Qiu X. Playing chemical plant protection game with distribution-free uncertainties. Reliability Engineering & System Safety. 2017.

[41] Nguyen TH, Jiang AX, Tambe M, editors. Stop the compartmentalization: Unified robust algorithms for handling uncertainties in security games. Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems; 2014: International Foundation for Autonomous Agents and Multiagent Systems.

[42] Vorobeychik Y, An B, Tambe M, editors. Adversarial patrolling games. Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3; 2012: International Foundation for Autonomous Agents and Multiagent Systems.