Applying game theory for adversarial risk analysis in chemical plants

Zhang, Laobing

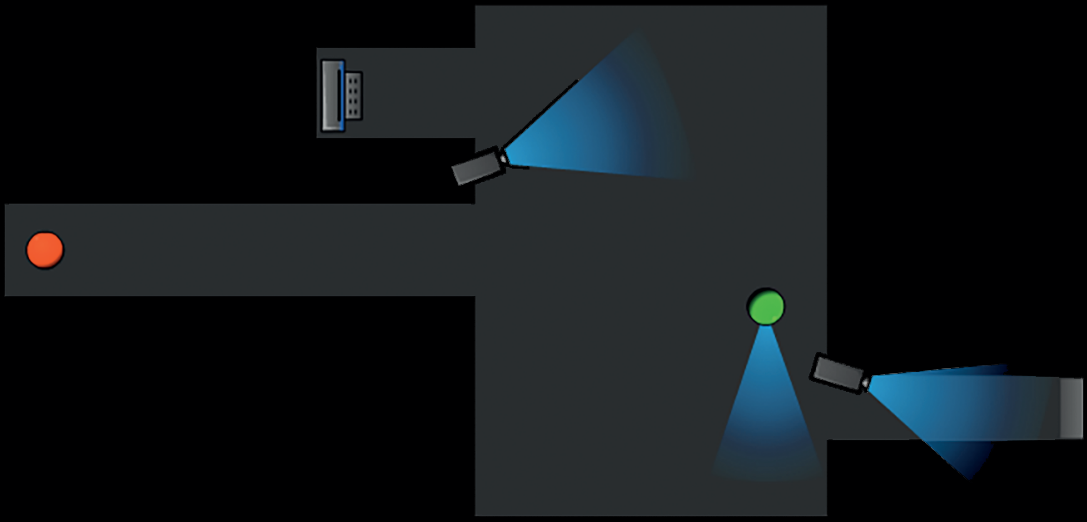
DOI
10.4233/uuid:eec6ef3b-3d9d-4b7d-8d9b-02fa5a4d9245

Publication date
2018

Document Version
Final published version

Citation (APA)
Zhang, L. (2018). *Applying game theory for adversarial risk analysis in chemical plants*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:eec6ef3b-3d9d-4b7d-8d9b-02fa5a4d9245

Applying Game Theory for Adversarial Risk Analysis in Chemical Plants



Laobing Zhang

Applying Game Theory for Adversarial Risk Analysis in Chemical Plants

Laobing Zhang

Delft University of Technology

Applying Game Theory for Adversarial Risk Analysis in Chemical Plants

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology,
by the authority of the Rector Magnificus Prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates,
to be defended publicly on
Monday 17 December 2018 at 15:00 o'clock
by

Laobing ZHANG
Master of Engineering in Control Science and Engineering
National University of Defence Technology, China
born in Leiyang, Hunan, China

This dissertation has been approved by the promotor:
Prof. dr. ir. G.L.L.M.E. Reniers

Composition of the doctoral committee:

Rector Magnificus	Chairman
Prof. dr. ir. G.L.L.M.E. Reniers	Delft University of Technology, promotor

Independent members:

Prof. dr. B.A. Van de Walle	Delft University of Technology
Prof. dr. ir. P.H.A.J.M. van Gelder	Delft University of Technology
Prof. dr. W.E.H. Dullaert	VU. Amsterdam
Prof. dr. S.W. Pickl	Universität der Bundeswehr München
Prof. dr. F. Zhou	China U. Mining and Technology
Prof. dr. D.R. Insua	Spanish Royal Academy of Sciences

This research was funded by China Scholarship Council (CSC) under Grant 201403170340.

Cover design by Ipskamp printing
Published and distributed by: Laobing Zhang
E-mail: laobingzhang.nudt@gmail.com
ISBN: 978-94-028-1307-4

Printed in The Netherlands.

To my family

Summary

Since the 9/11 attack in New York in 2001, a lot of attention has been paid to the protection of critical infrastructures. Chemical industries are without doubt critical infrastructures due to their extreme importance for society in combination with their vulnerability. They play important roles in modern-life society, from producing and providing daily necessities such as food and energy, to making modern medicine. They are thus truly essential to our modern way of living. Process plants usually store dangerous goods in large quantities, which may pose an important threat to themselves as well as to their surroundings. Moreover, due to a variety of benefits of scale, process plants tend to build their factories geographically together, potentially aggravating the danger. Therefore, the importance of protecting industrial process plants (including those in the chemical industry, the food industry, the energy industry, and others) cannot be overestimated.

Risks caused by human behaviours with the intention to cause losses are defined as security risks. For instance, thieves intentionally intruding a plant for stealing valuable materials, or terrorists maliciously setting a fire on a chemical facility to cause societal fear. Initiators of security events (henceforth, attackers) would intelligently observe the defender's defence plan and then schedule their attack accordingly. Literature has actually shown how resources can be misallocated if intelligent interactions between the defender and the attacker are not considered.

Game theory was developed in the economic domain for modelling both cooperative and competitive behaviours in a multiple actors system. In the last 100 years, game theory has been theoretically improved and practically applied to various domains, such as evolutionary biology, computer science etc. These researches have demonstrated the capability of game theory in modelling intelligent interactions. Several security management systems based on game theory have been developed and deployed in reality, such as the ARMOR system for the Los Angeles airport, the PROTECT system for the US coast guard, etc.

In this research, game theory is employed to study the protection of chemical industrial areas. Four models are proposed: i) DAMS – an agent-based modelling and simulation approach for assessing domino effects in chemical plants; ii) CPP game – a game theoretic model for single plant protection; iii) CCP game – a game theoretic model for multiple plants protection, by optimizing patrolling; and iv) PPG – a game theoretic model aiming at optimizing pipeline patrolling within or between chemical plants. These models are briefly explained hereafter.

Domino Effect Assessment by using Agent-Based Modelling and Simulation (DAMS): Domino effects, worsening the consequences of a primary accident scenario, regularly happen during chemical industrial major accidents. With regard to security, causing a domino effect accident can be the motivation for an attack on a chemical plant/cluster. The DAMS model innovatively employs an agent-based modelling and simulation approach for studying domino effects in the process industry, being able to calculate both the probabilities of domino escalation as well as the timing of the domino escalation. Emergency response with regard to domino effects can, for instance, be more efficiently planned with the support of temporal data.

Chemical Plant Protection Game (CPP game): The current mainstream security risk assessment methodologies are mainly based on the "$risk = threat \times vulnerability \times consequence$" concept, which can be problematic due to being very qualitative in its nature. The lack of quantitative calculations as well as the failure of modelling dynamic interactions between the defender and attacker, are obvious downsides of the currently widely used security risk assessment concept. To this end, the CPP game has been elaborated. The CPP game is developed based on the general intrusion detection approach in the chemical industry, with the purpose to better setting security alert levels at each entrance and in each zone of a chemical plant. The CPP game calculates the attractiveness of each asset to each type of threat as well as the overall security risk of the plant. The results follow quantitative calculations, and the intelligent interactions (e.g., the attacker may plan his attack according to the defender's defence) between the defender and the attacker are considered. Moreover, the defender's uncertainties on the attackers' parameters and on the attackers' rationality are considered in the advanced forms of the CPP game.

Chemical Cluster Patrolling Game (CCP game): A patrol is scheduled in chemical plants as well as in chemical clusters, to detect unauthorized intrusions. In a chemical cluster, the limited availability of patrollers cannot cover each chemical plant 24/7, raising the importance of optimizing the patrolling routes. The CCP game aims to generate random, but strategic, patrolling routes for patrollers, taking certain features/characteristics of dangerousness into consideration. Randomized patrolling brings high uncertainties about the patroller's real-time location to potential attackers. Strategic patrolling enables the patroller to patrol the more hazardous plants more frequently, but still randomly.

Pipeline Patrolling Game (PPG): Not only in fixed chemical sites, but also for protecting pipelines in chemical industrial areas, should patrolling be adequately scheduled. However, the current patrolling strategy (e.g., purely randomized patrolling or oscillation) has the drawback of being predictable and failing to cover more hazardous pipeline segments more intensively. Furthermore, the patrolling of a pipeline has a different form when compared to the patrolling of a chemical cluster. The PPG model therefore employs game theory to optimize pipeline patrolling and aims to generate random but strategic patrolling routes (similar to the CCP game).

Case studies are used for each model, for demonstrating how the models work and for verifying the models. Robustness of the models is validated by sensitivity analyses. The models are evaluated from a practical (feasibility) point of view by six security managers from industry for assessing the possibility of industrial application. All experts think that the proposed models have the potential to be implemented in industrial practice and are therefore convinced that the protection of chemical facilities can be improved. However, further improvements for the model are needed. At least ten gaps between the models and current industrial practice have been mentioned by the experts. Future research will be oriented to fill these gaps and to implement the models in practice for solidly improving chemical security.

CONTENTS

FIGURE CONTENTS

TABLE CONTENTS

1
INTRODUCTION

This chapter reports the motivation of this dissertation. Research questions are formulated and contribution of this dissertation is clarified. Finally, the organization of the dissertation is illustrated.

1.1 Motivation

After the 9.11 disaster in New York, people suddenly became aware of the fact that even if they were living in a peaceful country, a large-scale intentional terrorist attack could also happen to them. Similar to airplanes, chemical installations, if being attacked, will cause big losses on both facilities (the economy) and human lives (society). Furthermore, attacking chemical installations can easily result in a cascade disaster, also called domino effect, due to the strong interconnectedness between chemical plants as well as due to the numbers of installations within plants. Chemical industrial areas can therefore be argued as critical infrastructures which should be well protected from intentional attacks.

Until now, luckily no major terrorist attack has yet happened on a chemical facility in the Western world. The downside however of this happy observation is that industrial security managers are not willing to invest too much on the prevention and/or protection and mitigation of events that they assess to be extremely unlikely. A quantitative way is needed to show the managers how their limited security budgets can be efficiently used. According to the report of The 9/11 Commission, terrorists make decisions in response to the potential victim's observed strategies. Therefore making defence decisions without taking the intelligent adversaries into consideration will lead to a wrong or non-optimal allocation of the resources.

Game theory provides one way to account for the actions of intelligent adversaries. With its rigor and mathematical depth, significant recent research interest can be observed in game-theoretic approaches to security. With game theory, both the defender and the attacker, their actions, security resources, and attacking costs, can be quantitatively modelled. Quantitative models lead to quantitative security recommendations, such as which installation should be better protected? How intensively should a facility be protected? What would be the benefit and cost if a certain countermeasure is implemented? And so on.

Industrial managers are not interested in theoretical models and mathematical results, instead, they need easy-to-handle and user-friendly decision-support tools. What they prefer is an easy approach to input the system input data, reasonable (understandable) and correct (adequate) results, and an acceptable computation time. Therefore, this dissertation does not stop at proposing models and algorithms, but also attention is paid to the interface of industrial practise and mathematical (i.e., game theoretical in this dissertation) models.

1.2 Research questions

Safety science is still a young discipline, while the sub-topic of security is still in its infant stage. Among many other interesting research topics in the security domain, this dissertation intents to address the following question:

RQ: How to optimize the use of the limited security resources to improve security in a chemical industrial area, taking intelligent interactions between the attackers and the defender as well as the defender's uncertainties about the attackers, into consideration?

This research question contains three assumptions. The first assumption, namely, the limited security resources assumption, assumes that the defender always has a limited budget for security. Therefore, the defender can neither cover all the installations at a high security alert level at the same time, nor she[1] can defend all the entrances with an intensive scenario. Without the limited security resources assumption, there is no need for a PhD dissertation for studying the allocation of the resources. The second assumption is that, the attackers (e.g.,

[1] In this dissertation, we use she/her/her to represent the defender and he/him/his to represent the attacker.

disgruntled employees, activists, terrorists etc.) are intelligent and they would plan their attack according to the defender's plan. This is a defendable assumption since the security adversaries are human beings and some academic literature (1, 2) as well as some governmental reports (3, 4) also revealed this. The third assumption addresses the uncertainties. A characteristic of an attack is that it may suddenly happen at some time and at some locations, being difficult if not impossible, to be predicted. Furthermore, there is a lack of historical data about security risks (especially risks of terrorist attacks on critical infrastructures) in the chemical industry. Therefore, the defender has uncertainties about her adversaries and these uncertainties must be taken into consideration.

To answer the overall research question RQ, a list of sub-questions should be addressed.

SRQ1. What are the disadvantages of the non-game-theoretic methods that are currently used for security risk assessment in the chemical industry, for instance, the Security Vulnerability Assessment (SVA)?

To answer this question is to further clear the research motivation and the background. The SVA method (5, 6), after its birth, has been extensively conducted in American chemical industries, and has been the dominant method in the chemical security domain. However, despite its popularity, there are some criticisms on the SVA method. For instance, Cox (7) listed at least eight shortages of the RAMCAP SVA methods. Zhang et al. (8) proposed several aspects that are important for security assessment but which are missed from the American Petroleum Institute recommended Security Risk Assessment standard (hereafter in this dissertation called: the API SRA).

SRQ2. How to model the adaptive actions and the uncertainties about the attackers, being the two main differences between security research and safety research?

There is no 'intelligent adversary' in safety research at all, and the uncertainties of accidents can be modelled, for instance, in a statistical approach with the help of usually available safety-related data. Adversaries in security events, however, are intelligent. The interactions between the defender and the attackers are dynamic: one's decision would be affected by the other's decision and furthermore resulting in a nested decision problem (9, 10). Moreover, the lack of security-related data in the chemical industry makes the defender difficult to address uncertainties about the adversaries.

Previous research has shown some possible answers for these questions in other domains, such as that game theory can be employed for dealing with intelligent interactions; Bayesian games and convex analysis of utility functions can be used for modelling uncertainties.

SRQ3. How to enhance the security defence in a multi-plant area?

For economic as well as managerial reasons, chemical plants are often geographically located close to each other, therefore forming so-called chemical cluster, such as the Chemelot area (Netherlands), the port of Antwerp (Belgium) etc. Due to the possible existence of induced domino effects, security protection is very important in these clusters. In fact, plants in one cluster share some security risks: if one plant is attacked, an explosion or a leakage of polluted gas may lead to problems and cascading effects in nearly plants as well. Patrolling is generally regarded as an important approach for protecting chemical clusters. Literature has however shown that a fixed patrolling route, which is currently used by most patrollers, is inefficient since the patroller's real-time location is predictable by the adversaries. A purely randomized patrolling route, though unpredictable, fails on covering higher hazardous targets more frequently, and thus is inefficient as well. This dissertation therefore pays attention to assist patrollers to generate random but strategic patrolling routes, for chemical cluster patrolling as well as for pipeline patrolling.

1.3 Contribution

This dissertation aims at improving the protection of chemical facilities from deliberate attacks. Game theory is employed as the research methodology. Several important contributions to the chemical security community can be mentioned.

Contribution 1: An approach that combines conventional methods (e.g., the API SRA) and game theory is proposed, for improving the protection of chemical industrial areas. As stated in Chapter 2 of this dissertation, conventional security risk assessment methods in the chemical domain have drawbacks such as being failed on modelling the attackers as strategic decision makers and on providing quantitative recommendations. Game theory, especially the so-called 'security game' (see Section 2.3.2), is created for quantitatively modelling strategic decision making in a multiple stakeholders situation. However, game theory needs quantitative inputs and only generates numerical outputs, leading to a discrepancy between theory and chemical security practise. Therefore, conventional security methods are suggested to act as a bridge between chemical security practise and game theory. This contribution is illustrated in Chapter 2 and Chapter 4 (especially section 4.5).

Contribution 2: DAMS, an agent-based modelling and simulation approach for assessing domino effects in chemical industries, is proposed in this dissertation. The DAMS model is innovative on being able to assess not only the probabilistic aspect, but also the timing-related aspect, of the domino effect propagation procedure. The model can be used for calculating the consequence of a successful attack, taking into consideration domino effects. This contribution is illustrated in Chapter 3.

Contribution 3: The Chemical Plant Protection game, abbreviated as CPP game, is proposed, for the purpose of single plant protection. The CPP game is developed based on the general intrusion detection approach in chemical plants and it successfully captures the intelligent interactions between the defender and the potential attackers (see Chapter 4). Furthermore, the CPP game is extended to be able to deal with the defender's uncertainties on the attacker's parameters as well as on the attacker's rationalities (see Chapter 5).

Contribution 4: The Chemical Cluster Patrolling game, shortened as CCP game, is proposed, for generating random but strategic patrolling routes for a cluster patrolling team. Randomized routes increase the uncertainties (for the attackers) of the patroller's real-time location while strategic routes guarantee that the patroller patrols more hazardous plants more frequently. The CCP game is explained in Chapter 6.

Contribution 5: Security game is employed for optimizing the patrolling of oil/gas pipelines and a model named Pipeline Patrolling game (PPG) is proposed. Patrolling routes generated by the PPG are strategic, resulting from the fact that the PPG model firstly generates an optimal coverage rate for each segment of the pipeline, according to the importance of the segment. The implemented route is also unpredictable, since in the second step, the PPG model generates multiple patrolling routes according to the optimal coverage rate. The PPG model is elaborated in Chapter 7.

1.4 Organization of the dissertation

Nine chapters are employed to demonstrate the use of game theory for improving the protection of a chemical industrial area. Figure 1.1 shows an overview of the book.

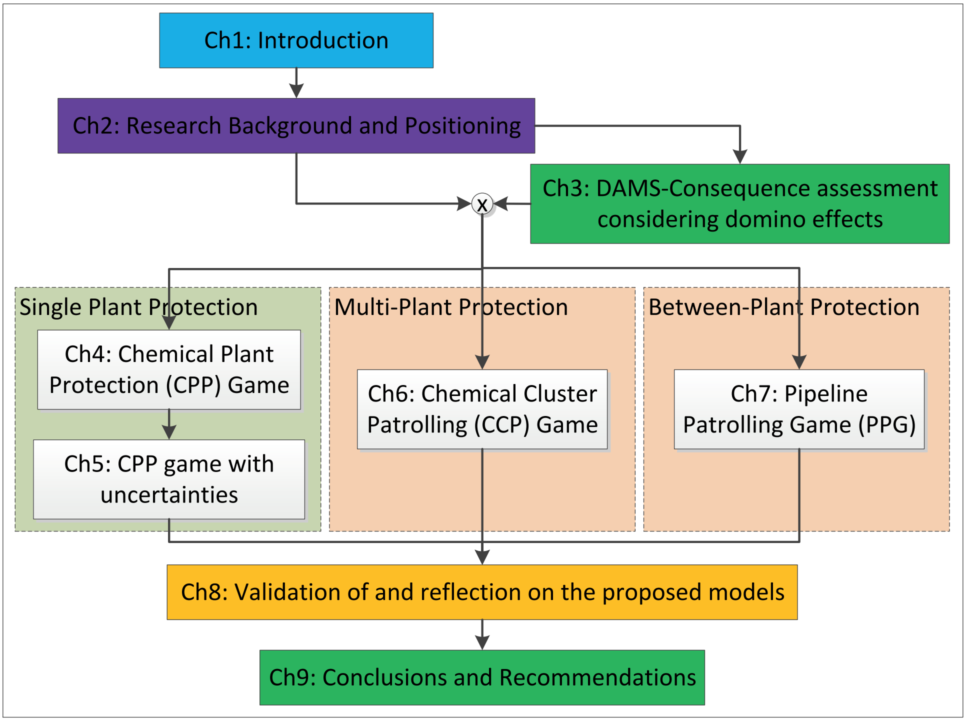

Ch1: Introduction
Ch2: Research Background and Positioning
Ch3: DAMS-Consequence assessment considering domino effects
X
Single Plant Protection
Multi-Plant Protection
Between-Plant Protection
Ch4: Chemical Plant Protection (CPP) Game
Ch5: CPP game with uncertainties
Ch6: Chemical Cluster Patrolling (CCP) Game
Ch7: Pipeline Patrolling Game (PPG)
Ch8: Validation of and reflection on the proposed models
Ch9: Conclusions and Recommendations

Figure 1. 1. Organization of the dissertation

Chapter 1 illustrates the motivation, the research questions, and the research philosophy of this dissertation. The main contributions of this dissertation are also summarized in this chapter.

Chapter 2 firstly points out that 'intentionality' is the key difference between a (deliberate) security event and a (coincidental) safety event. The importance of protecting chemical facilities is illustrated in the chapter. State-of-the-art literature and governmental regulations are discussed. The lack of historical data and the existence of intelligent adversaries are identified as the main challenges for improving security in chemical industrial areas. Secondly, Chapter 2 introduces game theory, which is the main mathematical approach used in this dissertation. Games with a discrete set of strategies are also discussed (and further used), since they are easier to solve as well as they better reflect reality than games with continuous strategies.

Chapter 3 demonstrates the DAMS model, which is developed for assessing domino effects in chemical plants, by employing agent-based modelling and simulation. Comparing to previous research conducted in the domino effect assessment domain, the DAMS model has advantages on being able to calculate higher level domino effects and on modelling the synergistic effects. Furthermore, temporal aspects of the propagation of domino effects are also captured. The DAMS model can be seen as a support model for other game theoretic models developed in this dissertation, since all those game theoretic models need quantitative inputs which include the consequences of a successful attack while domino effects have an important role on worsening the consequences of chemical accidents.

Chapter 4 and Chapter 5 concern the physical protection of chemical plants belonging to a single operator. In Chapter 4, a Chemical Plant Protection (CPP) game is developed, based on the so-called multiple layers protection approach for chemical plants. The CPP game is able to model intelligent interactions between the defender and the attackers. An analysis of the inputs and outputs of the CPP game is also provided.

However, the CPP game suffers a drawback, that is, a large amount of quantitative inputs is required. Chapter 5 therefore addresses this disadvantage, by proposing an Interval CPP game, which is an extension of the CPP game where the exact numbers of the attacker's parameters are no longer needed. Instead, in this game, only the intervals that the parameters will be situated in, are required. Thus, the Interval CPP game considers the defender's distribution-free uncertainties on the attackers' parameters and hence the inputs for the Interval CPP game are easier to obtain, for instance, by using the outputs from the API SRA method [(6)].

A second drawback of the CPP game concerns the "rational attacker" assumption. Chapter 5 therefore models bounded-rational attackers into the CPP game. Three robust solutions are proposed for the CPP game, namely, the Robust solution with epsilon-optimal attackers, the MoSICP solution and the MiniMax solution, for addressing attackers who may deviate to strategies having close payoffs to their 'best response' strategy, for addressing attackers who may play strategies with higher payoffs with higher probabilities, and for addressing attackers who only aim at minimizing the defender's maximal payoffs, respectively.

The CPP game is applied to a refinery to show how the game works and what results can be obtained by implementing the game. The refinery case is also used in the API SRA document for illustrative purposes. Therefore, the outputs from the API SRA method are used as one part of the inputs for the CPP game while other inputs of the CPP game are illustrative numbers.

Chapter 6 employs game theory for optimizing the scheduling of patrolling in chemical clusters or chemical industrial parks. A Chemical Cluster Patrolling (CCP) game is formulated. Both the hazardousness level of each plant and the intelligence of adversaries are considered in the CCP game, for generating random but strategic and implementable patrolling routes for the cluster patrolling team. the CCP game is applied to a chemical cluster composed of several plants each belonging to different operators, for optimizing the patrolling of security guards in the multi-plant area. Results show that the patrolling route generated by the CCP game well outperforms the purely randomized patrolling strategy as well as all the fixed patrolling routes.

Chapter 7 demonstrates a pipeline patrolling game (PPG). The PPG firstly calculates the coverage rate of different segments of a pipeline, according to the hazardousness of the segments. Secondly, the PPG generates multiple patrolling routes which satisfy the coverage rate calculated in the first step. The patroller then randomly chooses a route from the generated routes.

In Chapter 8, reflections from security practitioners from chemical plants on the models proposed in this dissertation are given and reflected upon. Six security managers from Sitech, Solvay, the Antwerp port, BASF SE, Shell, and CIMIND are interviewed. Their opinions about the possibilities of applying the proposed models in industrial practise and the gaps between the practise and the models are shown.

Nine conclusions and nine recommendations are given in Chapter 9, answering the research questions formulated in section 1.2. We conclude that security risk assessment in the chemical industry is still a young research domain and several future research directions are given, based on the researched conducted in this dissertation.

References

[1] Powell R. Defending against terrorist attacks with limited resources. American Political Science Review. 2007;101(03):527-41.

[2] Cox Jr LAT. Game theory and risk analysis. Risk Anal. 2009;29(8):1062-8.

[3] FAS. Al qaeda training manual. 2006.

[4] (DHS) DoHS. National strategy for homeland security. 2002.

[5] Moore DA. Security Risk Assessment Methodology for the petroleum and petrochemical industries. J Loss Prev Process Ind. 2013;26(6):1685-9.

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

[7] Cox Jr LAT. Some limitations of "Risk= Threat× Vulnerability× Consequence" for risk analysis of terrorist attacks. Risk Anal. 2008;28(6):1749-61.

[8] 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.

[9] Rios Insua D, Rios J, Banks D. Adversarial risk analysis. Journal of the American Statistical Association. 2009;104(486):841-54.

[10] Rios J, Insua DR. Adversarial risk analysis for counterterrorism modeling. Risk Anal. 2012;32(5):894-915.

2
BACKGROUND AND RESEARCH POSITIONING

We are convinced that physical security in chemical industrial areas can and should be improved, throughout the world. Chemical substances are stored and processed in large quantities in chemical plants and chemical clusters around the globe, and due to the materials' characteristics such as their flammability, explosiveness, and toxicity, they may cause huge disasters and even societal disruption if deliberately misused. Dealing with security implies dealing with intelligent adversaries and deliberate actions, as will also be further expounded in the next chapters. Such intelligent adversaries require smart solutions and flexible models and recommendations from the defender's side. Such is only possible via mathematical modelling and through the use of game-theory as a technique for intelligent strategic decision-making support.

Part of this chapter has been published in:

Zhang, L., Reniers, G., Chen, B., & Qiu, X. (2018). Integrating the API SRA methodology and game theory for improving chemical plant protection. Journal of Loss Prevention in the Process Industries, 51, 8-16.

2.1 Protecting process industries from intentional attacks: the state of the art

2.1.1 Safety and security definitions and differences

Definition

Safety and security are two related concepts but they have a different basis. Table 2.1 [(1)] gives an overview of differences between safety and security. In summary, while safety risks concern possible losses caused by non-intentional events, such as natural disasters, failure of aging facilities, and mis-operations, etc, security risks are related to possible losses caused by intentional human behaviour, such as terrorist attacks, sabotage by disgruntled employees, criminals, etc.

Table 2. 1. Non-exhaustive list of differences between safety and security

Safety	Security
The nature of an incident is an inherent risk	The nature of an incident is caused by a human act
Non-intentional	Intentional
No human aggressor	Human aggressor
Quantitative probabilities and frequencies of safety-related risks are often available	Only qualitative (expert-opinion based) likelihood of security-related risks may be available
Risks are of 'rational' nature	Threats may be of symbolic nature

The importance of the differences between safety and security

A key difference, amongst others, between safety risks and security risks is whether there are intelligent interactions between the risk holder and the risk maker. "Intelligent interactions", in this statement, means that the risk maker must have the ability to schedule his behaviour to meet his own interests, according to the risk holder's behaviour. In a safety event, due to the mere characteristics of such event, risk makers do not have the ability to plan their behaviour.

For instance, a typical type of safety event is a natural disaster, such as an earthquake, a flood, extreme weather etc. In this kind of events, nature can be seen as the risk maker. The risk holders are targets (for instance, people, property, reputation, etc.) who suffer losses from these events. The risk holder may defend itself against nature (e.g., build higher dams or use lightning deflectors), but the risk maker, nature in our example, does not have its own interests and hence does not plan its behaviour.

A more complicated example is that the risk initiator behaves in a way that he would like to achieve a goal, but non-intentionally causes an unplanned accident. A typical scenario of this situation can be a thief stealing a computer from an organization for obtaining the hardware device, and accidently he steals a computer with important technical and confidential information (without backup available). This scenario concerns a security risk since it satisfies the following conditions: i) the thief has the ability to plan his behaviour according to the organization's defence; and ii) the thief has his own interests to meet.

The most difficult part of distinguishing a safety event from a security event is to judge whether the risk maker has his own interests with respect to the event or not. An industrial accident caused by a mis-operation, for example, is defined as a safety event. Nevertheless, an accident caused by a disgruntled employee (thus causing intentional mis-operation) would be defined as a security event. In both events, the risk maker has the ability to plan his action.

However, in case of the coincidental mis-operation (without the aim to cause losses), the employee does not have his own interest in causing the event and doesn't obtain anything from the event. In case of the disgruntled employee, the employee's interest is to obtain mental satisfaction from the event. This theoretical difference makes it extremely difficult in some cases to distinguish whether an accident can be classified as a security event or as a safety event and is rather ford for lawsuits.

The risk maker from a security viewpoint, although being able to behave according to the risk holder's behaviour, doesn't necessarily do so, and thus doesn't need to act intelligently. To have the ability to act intelligently is one thing, while to use this ability is another thing. Therefore, in security events, we may also see some random behaviours. For instance, an attacker with so-called 'bounded rationality' does exist in the real world. Furthermore, whether the risk maker (actually) behaves randomly is not a clear criterion to unambiguously decide whether the event can be classified as a safety or as a security related event. As an obvious example of this reasoning, in a terrorist attack scenario, when the defender enhances her defence, the attacker is supposed not to implement an attack any more. However, the attacker can behave irrationally (see also definition of 'rationality' in section 2.2.1), and despite the extra defence measures, attack the defender anyway.

2.1.2 The need of improving security in chemical plants

Security research has a long history. It has obviously been stimulated by the 9/11 attack in New York, and ever since, people ever more perceive terrorism as an urgent problem. Zhou et al. (2) summarized data from the Global Terrorism Database (3), indicating that, despite a number of academic studies and societal financial efforts for preventing terrorist attacks, the global amount of terrorist attacks sharply increased during the past decade.

Moreover, our highly connected modern societies are vulnerable and fragile to possible targeted attacks. Many networked sub-systems of the modern society such as the internet, interlinked financial institutions, airline networks, etc., satisfy the so-called "power-law" degree distribution. This means that only few nodes in these networks exhibit a high degree of importance in the network if compared to most other nodes belonging to the network. If these high-importance nodes would be intentionally attacked, the network would suffer severely.

In the process industries, we see that on the one hand chemical plants tend to 'cluster' together in industrial parks and to build geographically close to each other, due to all kinds of benefits of scale. However, due to the existence of so-called 'domino effects' (4, 5), if one plant or installation would be attacked intelligently, the whole cluster as well as its surrounding area could be affected. On the other hand, plants/companies are also highly dependent on their upstream and downstream plants, through the supply chain. Thus if one plant would be attacked and stops its operation, many more plants would be economically affected as well.

Summarizing the above observations, not only the frequency of terrorist attacks seems to be increasing, but due to the characteristics of our modern societies and the inter-connectedness between people and between companies, also the potential devastation of malicious attacks is growing.

Chemical and process plants have important roles for our modern way of life. They provide materials for our clothes, food, medicines etc. Chemical industries also form the foundation of modern transportation systems, by providing energies (mainly oil and gas) and stronger materials. Moreover, considering the fact that the chemical industry can be seen as the foundation of a lot of other industries, e.g., the manufacturing industry, its role in the regional economic surrounding cannot be overestimated.

Besides its importance for our modern way of live, the chemical industry may also pose an important threat to today’s society. Toxic and flammable materials, as well as extreme pressure and temperature conditions, may be involved in production processes. Therefore, if these materials are not operated and managed correctly, and/or the extreme production conditions are not controlled well, disastrous events might result. Many disasters can be mentioned as examples. For instance, Seveso in 1976 and Bhopal in 1984 are examples of the leakage of toxic gas causing huge consequences for industry and society. The Mexico City disaster in 1984 is the example of the worst-ever happened domino effect, causing 650 casualties [(4)].

All these abovementioned disasters were initiated by coincidence (for example, mis-operation or poor industrial management), and therefore they can be classified as safety events. If intentional attacks would have been involved in these disasters, they would have been even more difficult to predict and their consequences could in most cases be even higher. Actually, the worst ever industrial accident that happened in the chemical industry is the Bhopal gas tragedy in 1984, and the company operating the Bhopal plant at that time has always claimed that this disaster was a security event. However, the accident has been extremely thoroughly investigated, and we now know without any doubt that it was a safety related event. Nonetheless, two important observations can be made from this example: (i) the fact that the company always claimed that the event was security related indicates that without thorough investigation it is difficult to be sure of the nature of a disaster, and (ii) disasters could indeed be caused intentionally and if so, the consequences may be much higher than if caused coincidentally.

Before the 9/11 terrorist act, an intentional attack on a chemical plant was always believed to be extremely unlikely. In the post-9/11 era, more attention has been paid to the protection of chemical plants from malicious human behaviour. Chemical and process plants were listed as one of the 16 critical infrastructures in the United States that should be well protected from terrorist attacks [(6)]. In 2007, the Department of Homeland Security (DHS) implements the Chemical Facility Anti-Terrorism Standards (CFATS) Act for the first time, which obliges to identify high-risk chemical facilities and ensures corresponding countermeasures are employed to bound the security risk. Pasman [(7)] points out that three possible terrorism operations may happen within the chemical industry: (i) causing a major industrial incident by intentional behaviour, for example, by using a bomb or even simply by switching off a valve; (ii) disrupting the production chain of some important products, e.g., medicines; and (iii) stealing materials for a further step attack, e.g., obtaining toxic materials and release it in a public place.

In Iraq, frequent attacks to oil pipelines and refineries caused more than 10 billion dollars in the period 2003 - 2005 [(8)]. Furthermore, an analysis carried out by Khakzad [(7)] reveals that chemicals are involved in more than half of the terrorist attacks which happened in the world in 2015.

Reniers and Pavlova [(9)] categorize accidents into three different types, namely Type I, Type II and Type III, according to the available historical data of these accidents. Type I accidents are accidents with abundant data, and are mainly referring to individual level events, such as falling, slipping, little fires etc. Type II accidents are accidents with extremely/very little records of data, and are mainly referring to industrial disasters, such as the Bhopal disaster, the Seveso disaster etc. Type III accidents are accidents with no historical data at all, so-called black swans, and are mainly referring to accidents where multiple plants are involved. Type III accidents can however be seen as the extremum of Type II accidents. In security terminology, Type I events can be seen as thefts, manslaughter and murder, while Type II events are terrorist attacks.

Reniers and Khakzad [(10)] further argue that although two safety revolutions happened in the last century, dramatically reducing the number of Type I accidents, a new revolution is needed for further reducing the Type II accidents. Moreover, previous methodologies and theories for reducing Type II events are mainly conducted from a safety point of view. In the post-9/11 era, accidents initiated by intentional behaviour should also be considered, and if so, one can no longer be confident to say that the probability of a Type II event is extremely low.

2.1.3 Challenges with respect to improving chemical security

Two challenges make security research related to chemical plants particularly difficult: (i) the lack of research data (statistical historic data or experimental data); and (ii) the existence of intelligent adversaries.

Security events, in particular terrorist attack events, do not happen frequently in chemical plants, and for those that did happen, the data collection is not sufficient. Therefore, only scarce security data is available. To make matters even more difficult, most security related data is protected very well, at least to the public and to academic researchers. Due to the lack of available data, statistical models and methods for modelling risk makers' behaviour are not applicable. Statistical modelling has nonetheless a long history of being used in the safety domain. For instance, by collecting data, industrial managers know which segment of a pipeline is the most vulnerable part.

Statistical modelling may also be used in the security domain. For instance, by collecting the number of detected intruders, we can evaluate the efficiency of the intrusion detection system (IDS). In any case, statistics-based learning doesn't work when there are only a limited number of records. Furthermore, intruders might be deterred due to an enhanced IDS, which will further reduce the number of detected intruders.

The existence of intelligent adversaries is another challenge for improving security. As we stated in the previous section, security risk makers would plan their behaviour according to the risk holder's defence, in order to meet the risk maker's own interests. Therefore, in security events, the defender has to always take the attacker's response into consideration. Figure 2.1 illustrates how resources can be mis-allocated if the defender does not take intelligent attackers into account. In Figure 2.1, comparison of security investments to a non-strategic terrorist (the left hand side figure) and to a strategic terrorist (the right hand side figure) is shown. Ten resources are being allocated to two sites which values three and two respectively. The curve in the left hand figure is plotted as $DEL = \alpha_1 \cdot L_1 \cdot v_1(r) + \alpha_2 \cdot L_2 \cdot v_2(R-r)$, which denotes the conventional security vulnerability assessment (SVA) methodology. The curves in the right hand side figure are plotted as $DEL1 = L_1 \cdot v_1(r)$ and $DEL2 = L_2 \cdot v_2(R-r)$, for the decreasing curve and for the increasing curve respectively, and they denote the game theoretic results. It reveals that the SVA methodology without considering the strategic terrorists suggests to allocate $r^* \approx 8.3$ resources to site 1 while the game theoretic model which models the intelligent interactions between the defender and the attacker, suggests to allocate $\hat{r} \approx 5.8$ resources to site 1. Figure 2.1 and its corresponding explanation are adopted from Powell [(11)].

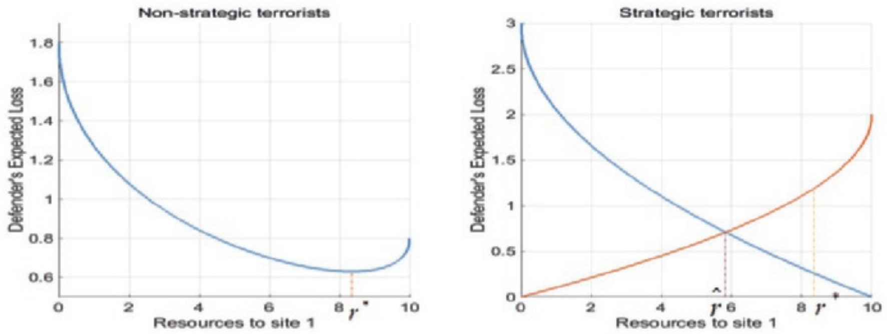


Figure 2. 1. Security investment w.r.t. strategic vs. nonstrategic terrorist

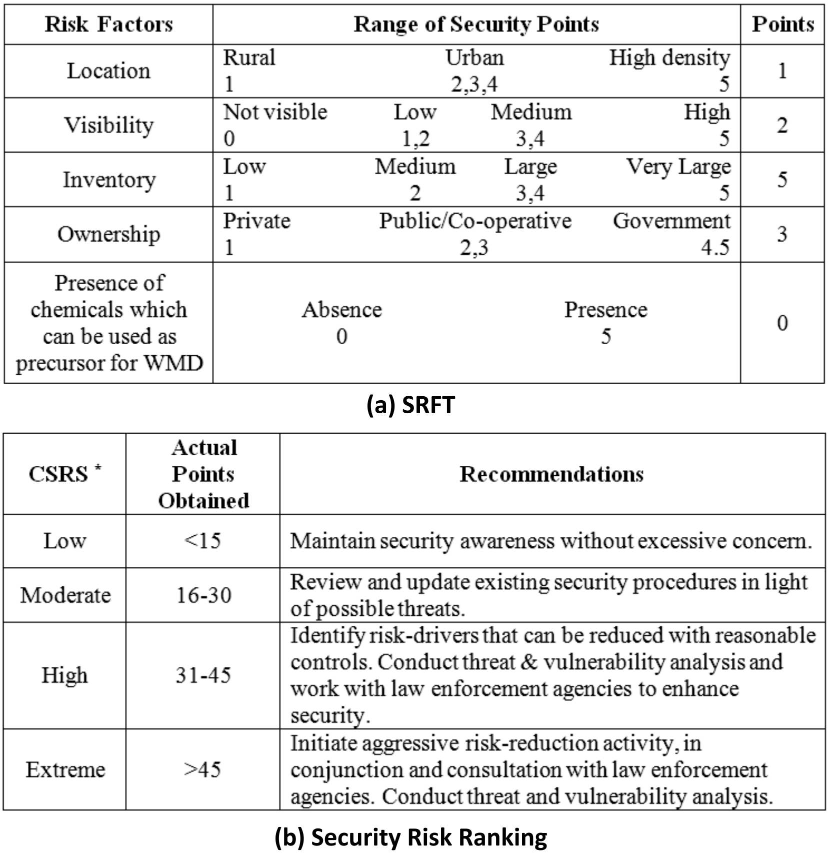
Moreover, the existence of intelligent adversaries also highlights the challenge with respect to the lack of data. Since security adversaries are so-called 'intelligent', the statistical data based approach, if being used in security risk assessment, can be misleading. For instance, some security risk assessment methods also try to employ a data based approach for predicting security events. The API SRA standard (12), among others, suggests a historic data based approach for estimating threat ranking for the chemical industries. According to the API SRA standard, most chemical plants have the same – very low – level of terrorist threat ranking, since most of them have "no expected attack in the life of the facility's operation". However, whether an intelligent attacker would attack the plant or not, does not depend much on the historic data, instead, it depends on whether the plant can meet their own interest and on whether their attack on the plant would easily be successful or not.

Furthermore, it is difficult to collect experimental data for behaviour modelling of an intelligent adversary. Security adversaries would not join any security experiments and they can hide their behaviours during the experiments as well. For instance, for a safety research purpose, psychological experiments can be employed to estimate the probability of human errors in different situations. However, if this experiment would be carried out for a security purpose, then finding attacker participants is difficult (if not impossible) and if ordinary people would be invited to act as attackers, the data would not be reliable since attackers and ordinary people can behave totally differently.

2.1.4 Security risk assessment in chemical plants: state-of-the-art research

Academic research on the topic of security has been dramatically stimulated by the 9/11 attack, while the research efforts on better protecting process industries from deliberate attackers are still not enough yet. Baybutt (13-23) emphasized the necessity of protecting chemical facilities from terrorists, criminal acts, as well as sabotages. Not only the physical security perspective is important, but also the cyber perspective should be taken into consideration, as indicated in Baybutt's publications (23). An asset-based approach and a scenario-based approach are proposed, for bettering security in the process industries (21, 22). Gupta and his co-authors (24-27) 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 (1, 5, 28-35) conducted security research in the chemical clusters, from the management factors to the technological factors. A model estimating the vulnerabilities of industrial facilities to attacks with improvised explosive devices are proposed by Landucci et al. (36). Argenti et al. (37-40) employed a Bayesian network approach for assessing the attractiveness and vulnerabilities of chemical facilities, conditional probabilities of which were estimated based on interviews with industrial practitioners.

Most of the current literature assesses security risks in chemical plants from a threat/vulnerability/consequence framework and research conducted in this literature is qualitative or semi-quantitative. Amongst the existing researche and regulations, two systematic methods can be mentioned: the Security Risk Factor Table (SRFT) [(24)] and the Security Vulnerability Assessment Methodology (SVA) [(12)].



Risk Factors	Range of Security Points	Points
Location	Rural Urban High density 1 2,3,4 5	1
Visibility	Not visible Low Medium High 0 1,2 3,4 5	2
Inventory	Low Medium Large Very Large 1 2 3,4 5	5
Ownership	Private Public/Co-operative Government 1 2,3 4.5	3
Presence of chemicals which can be used as precursor for WMD	Absence Presence 0 5	0

(a) SRFT

CSRS *	Actual Points Obtained	Recommendations
Low	<15	Maintain security awareness without excessive concern.
Moderate	16-30	Review and update existing security procedures in light of possible threats.
High	31-45	Identify risk-drivers that can be reduced with reasonable controls. Conduct threat & vulnerability analysis and work with law enforcement agencies to enhance security.
Extreme	>45	Initiate aggressive risk-reduction activity, in conjunction and consultation with law enforcement agencies. Conduct threat and vulnerability analysis.

(b) Security Risk Ranking

Figure 2. 2. SRFT example from Bajpai (CSRS: Current Security Risk Status)

SRFT

In 2002, SRFT was first proposed by the "Advanced Chemical Safety Company" to carry out a security risk assessment for a given chemical facility. The basic idea is to identify security-related factors of the given facility, to rate them on a scale from 0 to 5, with 0 being the "lowest risk" and 5 being the "highest risk", and finally to sum up the scores of each factor to measure the security risk status of the facility. Figure 2.2 (a) shows an example of a part of an SRFT table [(24)]. In the example given by Bajpai, the chosen factors are Location/Visibility/Inventory etc.; for each factor , scoring criteria are given, and each factor obtains a score of 1, 2, 5 etc.; the total score of this facility is 35. He finally concludes that security risk of this plant is High, according to Figure 2.2 (b).

In summary, the SRFT method divides the facility into various zones and identifies the factors influencing the overall security of the facility by rating them on a scale. It is a systematic approach to do security risk assessment, and it allows vulnerability ranking.

Some drawbacks of the SRFT method are obvious: (i) it is a qualitative and very subjective method; (ii) different factors have different weights in the security assessment, simply summing up the points of each factor can mislead the ranking; and (iii) intelligent interactions between defender and attacker are not considered at all.

The API SRA standard

In 2003, the first "SVA method" as it has become known afterwards, was developed by the American Petroleum Institute (API) to perform security risk assessment in the petroleum and petrochemical industries. In this security risk assessment, a security risk was defined as a function of Consequences and Likelihood; Likelihood being a function of Attractiveness, Threat, and Vulnerability. Table 2.2 shows detailed definitions of some terminologies used in this first so-called SVA method. The SVA methodology consists of 5 steps: (i) Characterization- Characterize the facility or operation to understand what critical assets need to be secured, their importance, their infrastructure dependencies and interdependencies; (ii) Threat Assessment- Identify and characterize threats against those assets, and evaluate the assets in terms of attractiveness of the targets to each threat and the consequences if they are damaged, compromised, or stolen; (iii) Vulnerability Assessment- Identify potential security vulnerabilities that enhance the probability that the threat would successfully accomplish the act; (iv) Risk Assessment- Determine the risk represented by these events or conditions by determining the likelihood of a successful event and the maximum credible consequences of an event if it were to occur; rank the risk of the event occurring and, if it is determined to exceed risk guidelines, make the recommendations to risk reductions; (v) Countermeasures analysis- Identify and evaluate risk mitigation options (both net risk reduction and benefit/cost analyses) and re-assess risks to ensure the adequate countermeasures are being applied. Evaluate the appropriate response capabilities for security events and the ability of the operation or facility to adjust its operations to meet its goals in recovering from the incident. In 2013, API published a new version of SVA and in this version, SVA was named as Security Risk Assessment (SRA). But the basic terms and steps are the same. Hereafter in the book, we name this methodology as "the API SRA methodology".

Table 2. 2. Definitions of terminologies in the API SRA method

Terminology	Definition
Consequence	The degree of injury or damage that would result if there were a successful attack
Threat	Any indication, circumstance or event with the potential to cause loss of, or damage, to an asset
Vulnerability	Any weakness that can be exploited by an adversary to gain unauthorized access and subsequent destruction or theft of an asset
Attractiveness	An estimate of the real or perceived value of a target to an adversary

Figure 2.3 in combination with Table 2.3, briefly illustrate the security risk assessment and management procedure of the API SRA methodology. The left-hand side of Figure 2.3 shows the sub-steps of the methodology, while the right-hand side shows the output data of each step. Explanations of the outputs are given in Table 2.3.

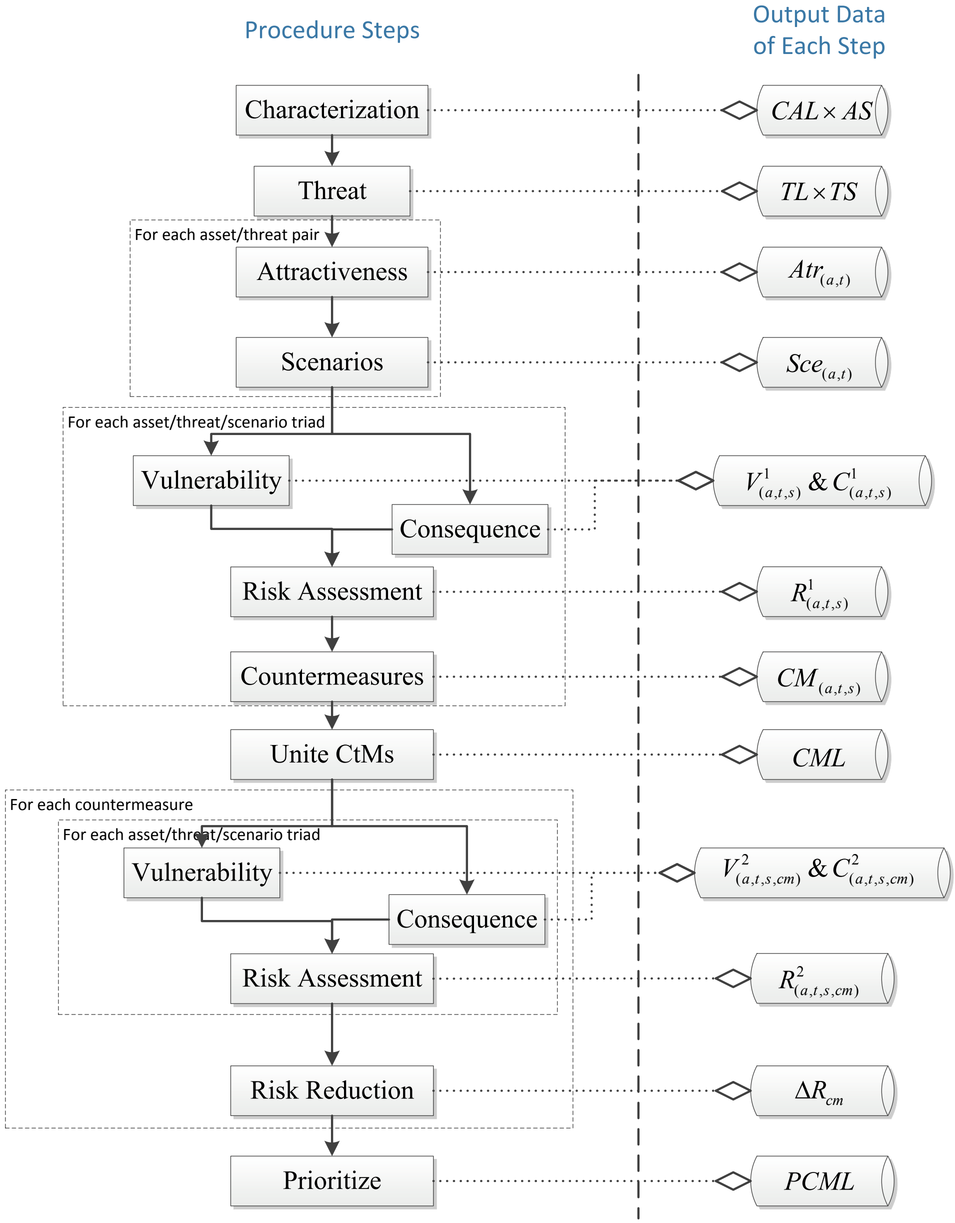


Figure 2. 3. The API SRA procedure

In the characterization step, the SRA team roughly scans the given petrochemical plant, and provides a critical assets list CAL as well as asset severity scores AS, according to functions of assets, interconnectivities among assets, and possible consequences. In the threat assessment step, the SRA team decides a threats list TL and threat levels TS that the plant is faced with, based on historical security data (site-specific, national, worldwide) and intelligence. For each asset and threat pair $\{(a,t)|a \in CAL, t \in TL\}$, the asset's attractiveness to the threat $Atr_{(a,t)}$ and possible attack scenarios linking the threat with the asset $Sce_{(a,t)}$ are evaluated. Based on current (situation '1') security countermeasures, vulnerabilities $V^1_{(a,t,s)}$ and consequences $C^1_{(a,t,s)}$ are estimated for each asset, threat, and scenario triad $\{(a,t,s)|a \in CAL, t \in TL, s \in Sce_{(a,t)}\}$. Furthermore, the SRA team calculates the likelihood of an attack from a given threat $t \in TL$ to a given asset $a \in CAL$ as $L^1_{(a,t)} = TS_t \times Atr_{(a,t)}$, and calculates the likelihood of a successful attack from t to a by using scenario $s \in Sce_{(a,t)}$ as $L_{(a,t,s)} = L^1_{(a,t)} \times V^1_{(a,t,s)}$. The risk matrix method is used to calculate a security risk $R^1_{(a,t,s)}$ for each asset, threat, and scenario triad, and in this step, the likelihood of a successful attack $L_{(a,t,s)}$ and the scenario-specific consequence $C^1_{(a,t,s)}$ are used to determine the risk value in the risk matrix. Based on the gaps between the current security risk and the desirable level of risk, scenario-specific countermeasures $CM_{(a,t,s)}$ are proposed by the SRA team, and subsequently all the scenario-specific countermeasures are united into one countermeasure list CML.

The SRA team further re-estimates the vulnerabilities $V^2_{(a,t,s,cm)}$, consequences $C^2_{(a,t,s,cm)}$, and security risks $R^2_{(a,t,s,cm)}$, presuming that a countermeasure $cm \in CML$ is implemented (situation '2'). Based on the recalculation, the risk reduction of each countermeasure ΔR_{cm} can be calculated as the summation of risk reduced in each asset, threat, and scenario triad, as shown in Formula (2.1). Finally, the proposed countermeasures are ranked according to their potential risk reduction ΔR_{cm} as well as some other practical information (e.g., costs).

$$\Delta R_{cm} = \sum_{a \in CAL} \sum_{t \in TL} \sum_{s \in Sce(a,t)} (R^2_{(a,t,s,cm)} - R^1_{(a,t,s)}). \qquad (2.1)$$

Table 2. 3. Output data of the API SRA methodology

Notation	Definition	Comments*
$\boldsymbol{CAL}$	Critical assets list	e.g., control centre, gasoline tanks etc. Ref to "assets" column in form 1.
$\boldsymbol{AS}$	Asset score	Measuring asset severity. Ref to "asset severity ranking" column in form 1.
$\boldsymbol{TL}$	Threat list	e.g., terrorists, disgruntled employee etc. Ref to "threat" column in form 2.
$\boldsymbol{TS}$	Threat score	Measuring threat ranking. Ref to "threat ranking" column in form 2.
$\boldsymbol{At}_{(\boldsymbol{t},\boldsymbol{a})}$	A given asset's (a) attractiveness to a given threat (t).	$t \in TL, a \in CAL$. Numbers, ref to column 2a1, 2b1 etc. in form 3.
$\boldsymbol{Sce}_{(\boldsymbol{a},\boldsymbol{t})}$	A given threat's possible attack scenarios to a given asset.	Ref to "scenario" column in form 4.
$\boldsymbol{V}^{\boldsymbol{1}}_{(\boldsymbol{a},\boldsymbol{t},\boldsymbol{s})}$, $\boldsymbol{C}^{\boldsymbol{1}}_{(\boldsymbol{a},\boldsymbol{t},\boldsymbol{s})}$	Vulnerability '1' and Consequences '1' (in case the attack is successful) of an attack scenario from a given threat to a given asset.	$t \in TL, a \in CAL, s \in Sce_{(a,t)}$. Ref to the "V" and "C1" column in form 4.

$\boldsymbol{R}^{\boldsymbol{1}}_{(\boldsymbol{a},\boldsymbol{t},\boldsymbol{s})}$	Security risk ‘1’ of a given asset from a given threat by using a given attack scenario.	Ref to the “R1” column in form 4.
$\boldsymbol{CM}_{(\boldsymbol{a},\boldsymbol{t},\boldsymbol{s})}$	Recommended countermeasures to reduce security risk of a given asset from a given threat by using a given attack scenario.	Ref to “proposed countermeasures” column in form 4.
$\boldsymbol{CML}$	Recommended countermeasure list	$CML = \bigcup_{a,t,s} CM_{(a,t,s)}$.
$\boldsymbol{V}^{\boldsymbol{2}}_{(\boldsymbol{a},\boldsymbol{t},\boldsymbol{s},\boldsymbol{cm})}$, $\boldsymbol{C}^{\boldsymbol{2}}_{(\boldsymbol{a},\boldsymbol{t},\boldsymbol{s},\boldsymbol{cm})}$	Vulnerability ‘2’ and Consequences ‘2’ (in case the attack is successful) of an attack scenario from a given threat to a given asset, presuming a suggested countermeasure is implemented.	$cm \in CML$. Ref to “Residual Risk” column in form 5.
$\boldsymbol{R}^{\boldsymbol{2}}_{(\boldsymbol{a},\boldsymbol{t},\boldsymbol{s},\boldsymbol{cm})}$	Security risk ‘2’ of a given asset from a given threat by using a given attack scenario, presuming a suggested countermeasure is implemented.	
$\Delta\boldsymbol{R}_{\boldsymbol{cm}}$	Risk reduction by a proposed countermeasure.	Ref to “Risk Reduction” column in form 6.
$\boldsymbol{PCML}$	Countermeasure list with priority ranking	Ref to “overall priority” column in form 6.

* Forms in this column refer to the forms in the API SRA standard document (12).

The API SRA methodology is a systematic process that (i) evaluates the successful likelihood of a threat against a facility, (ii) considers the potential severity of consequences to the facility itself, to the surrounding community, and to the energy supply chain, (iii) provides clear definitions of terminologies used in the methodology, (iv) clearly points out what inputs the methodology needs for the security risk assessment procedure and what outputs the methodology will provide, and (v) gives guidance on how to organize a SRA team.

Focusing on minimizing the defender’s maximal loss and taking into account uncertainties during the security risk assessment procedure, the API SRA methodology would output robust results. Though it is not a fully quantitative risk assessment based methodology, it is performed qualitatively using the best judgment of the SRA Team. Comparing to the SRFT, the API SRA methodology is more concrete to execute, and considers not only the facility itself, but its surroundings as well.

The API SRA methodology suffers two drawbacks. First, the methodology fails to model the dynamic (intelligent) interactions between defender and attackers. As shown in Figure 2.3, the SRA team estimates the attractiveness of each asset to each threat at the very beginning of the procedure. However, after presuming that the recommended countermeasures are implemented, the SRA team does not re-estimate the attractiveness. Therefore, in reality, the attackers would change their targets according to the defender’s plan. Second, risk scoring methods and risk matrices are employed in the API SRA methodology. For example, Cox (41) and Baybutt (42) have criticized the use of risk scores and risk matrices and proposed improvements.

2.2 Intelligent interaction modelling: game theory

2.2.1 Preliminaries of game theory, setting the scene

Introduction

Game theory is a mathematical tool for supporting decision making in a multiple players situation where one player's utility will be determined not only by his own decision, but also by other players' decisions. An illustrative example of this situation is the Rock/Scissors/Paper game ("RSP" game). In an RSP game, whether a player wins or loses depends on both what he plays and what his opponent plays. This is a well-known game between mostly children with very simple rules. Two 'players' hold their right hands out simultaneously at an agree signal to represent a rock (closed fist), a piece of paper (open palm), or a pair of scissors (first and second fingers held apart). If the two symbols are the same, it's a draw. Otherwise rock blunts scissors, paper wraps rock, and scissors cut paper, so the respective winners for these three outcomes are rock, paper and scissors. The RSP game is what is called a 'two-player zero-sum non-cooperative' game. There are obviously many other types of game and the field of game theory is very powerful to provide (mathematical) insights into strategic decision-making.

Game theory was formulated as a research domain after von Neumann and Morgenstern's work (43). Before their work, there was scattered research on interactive decision making, in which the idea of game theory was employed. Among others, Cournot's duopoly model, for example, studied how to predict the production of two monopolistic companies. The Stackelberg leadership model, on the other hand, investigated how to predict production of different companies when there is a leader/dominant company. von Neumann and Morgenstern (43) systematically studied strategic behaviours in the economic area, and proposed the famous MaxiMin theory based on a zero-sum game. Nash (44) studied general sum games, and proved that in a game with finite players and finite strategies, a Nash equilibrium always exists. Harsanyi (45) investigated games with incomplete information, and proposed the Harsanyi transformation to transfer an incomplete information game to a complete but imperfect information game. In the 20th century, game theoretic research is mainly stimulated by mathematicians, economists and sociologists, and several game theorists were awarded the Nobel prize, such as John Nash, Robert Aumann, and Lloyd Shapley etc. Furthermore, actually, all five game theorists who have won Nobel Prizes in economics, have been employed as advisors to the U.S. Pentagon at some stage in their careers.

Since the end of the 20th century, with the advances in computer science and the power of computer technology, game theory has been introduced to the computer science community. In the application perspective, game theory can be used for the allocation of network resources, for the modelling of intelligent agents in the artificial intelligence domain, for adversarial machine learning etc. Some computer scientists focus on theoretically developing efficient algorithms to calculate equilibria for large-scale game theoretic models. It is worth noting that Nash proved the existence of Nash equilibrium (NE) (see also section 2.2.2.2) in finite games, as mentioned above, however, his proof is not a constructive proof. Therefore, algorithms for computing the NE must be developed. Lemke and Howson (46) proposed an algorithm for searching one NE in a bi-matrix game. Chen and Deng (47) further proved that the task of computing a NE in a two-player game cannot be finished in polynomial time. Interested readers for computational issues in game theory are referred to Nisan et al. (48)

Basically, a game theoretic model consists of players (that is, decision-makers), strategies, and payoffs. Two assumptions, namely the 'common knowledge' assumption and the ‘rationality’ assumption, are often discussed in game theoretic models. Furthermore, different game solutions need to be employed for simultaneous games and for sequential games.

Players

Players need to be seen as strategic actors involved in the game. Actors can be people, but also institutions, organisations, etc., and even countries. A game theoretic model must contain at least two players. If there are only two players, the game is called a two-player game, otherwise the game is called a multiple-player game.

If cooperation can be achieved between players, the game is called a cooperative game, otherwise it is called a non-cooperative game. For instance, in a chemical cluster security case, different plants may cooperate with each other to jointly strategically invest in security measures, and thus a cooperative game can be applied to model this situation. The defender (cluster organisation) and the potential attacker (e.g., terrorists), however, will never cooperate of course, and thus a non-cooperative game should be employed in this case.

In this dissertation we mainly focus on two-player non-cooperative games.

Strategy (set)

Each player in a game in principle has a set of strategies. A strategy set defines the player’s behaviour rules when playing the game. A behaviour rule means that the player chooses a certain action at a certain step in the game (in game theory terminology this is called an information set). Therefore, a strategy can be seen as a series of actions of a player.

For instance, in a simple defender-attacker game, assume that the defender (the attacker) only has two actions, namely, “not defend (no attack)” and “defend (attack)”. Figure 2.4 (left-hand side) shows the game tree when the attacker does not know the defender’s action yet when he has to make a decision (i.e., the game is played simultaneously, see also section 2.2.1). Figure 2.4 (right-hand side) shows the game tree when the defender plays first and the attacker plays afterwards knowing what action the defender has played (i.e., the game is played sequentially, see also section 2.2.1). In both cases, the defender’s strategy set equals her action set, that is:

$S_d = \{\text{ND: not defend, D: defend}\}$.

In the simultaneous game, the attacker’s strategy set equals his action set, being the following couple of strategies:

$S_a = \{\text{NA: no attack, A: attack}\}$,

while in the sequential game, the attacker’s strategy set is different from his action set, that is:

$$
\begin{aligned}
S_a = \{&NA - NA\text{: if ND, then NA, if D, then NA;}\\
&\text{NA} - \text{A: if ND, then NA, if D, then A;}\\
&\text{A} - \text{NA: if ND, then A, if D, then NA;}\\
&\text{A} - \text{A: if ND, then A, if D, then A}\}.
\end{aligned}
$$

In the simultaneous game as shown in Figure 2.4 (left-hand side), the attacker only has one information set, which is shown as a dotted line oval. However, in the sequential game, the attacker has two information sets, which is shown as a circle in Figure 2.4 (right-hand side). The attacker’s strategy set should define the rules how the attacker could move at each information set, hence resulting in 4 strategies in the sequential case.

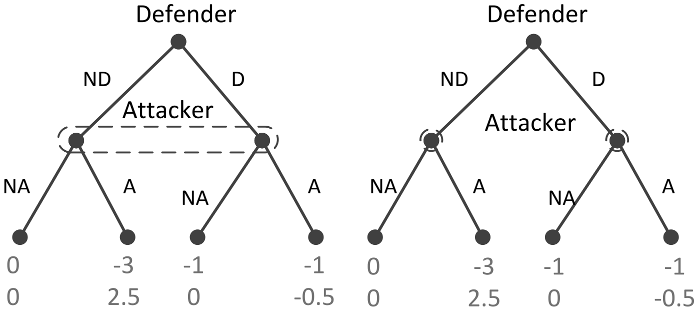


Figure 2. 4. Game tree of a illustrative defend-attack game

The strategy set we have defined is also called the "pure strategy set". A so-called 'mixed strategy' allows the player to probabilistically play each pure strategy. In the latter case, the sum of the probabilities that each pure strategy is played, equals to 1. For instance, in the illustrative game shown in Figure 2.4, the mixed strategy space for the defender can be defined as:

$\mathrm{X} = \{\mathrm{x} \in R^2 | x_{ND}, x_D \geq 0, x_{ND} + x_D = 1\}$,

in which x_{ND} represents the probability that the pure strategy ND will be played and x_D denotes that probability that the pure strategy D will be played.

Payoff

The payoff of a game models (and measures) the player's interests/goals/aims/preferences in the game. A payoff needs to be defined for each player and for every combination of strategies, that is, $u_i(\prod_{p=1}^{N} s_p) \to R$, in which, N represents the number of players in the game, s_p denotes the p^{th} player's pure strategy, u_i denotes the i^{th} player's payoff. For instance, in the game shown in Figure 2.4 (left-hand side), we have $u_D(ND, NA) = 0, u_D(ND, A) = -3, u_A(ND, NA) = 0, u_A(ND, A) = 2.5$ etc. For the game shown in Figure 2.4 (right-hand side), we have $u_D(ND, NA - NA) = 0, u_D(ND, NA - A) = 0, u_D(ND, A - NA) = -3, u_D(ND, A - A) = -3$ etc.

Explanation of the payoff numbers in Figure 2.4: It is assumed that the defender's defence cost (C_d) and the attacker's attack cost (C_a) are 1 and 0.5 respectively; the defender's losses from a failed attack (L_0) and from a successful attack (L) are 0 and 3 respectively; the attacker's penalty (P) from a failed attack is 0 while his reward (R) from a successful attack is 3. Further assume that if the defender defends the target and the attacker attacks this target, then the attack will fail, otherwise if the defender does not defend the target and the attacker attacks this target, then the attack will succeed. Therefore, if the defender plays 'ND' and the attacker plays 'NA', then both players have a payoff of 0; if the defender plays 'ND' and the attacker plays 'A', then the attack will succeed and the defender loses her value from the target (i.e., -3) and the attacker gets his reward but loses his attack cost (i.e., 3-0.5=2.5); if the defender plays 'D', and the attacker plays 'NA', then the defender loses her defence cost (i.e., -1) and the attacker has a payoff of 0; if the defender plays 'D' and the attacker plays 'A', then the attack will fail and the defender only loses her defence cost (i.e., -1) and the attacker loses his attack cost and suffers a penalty (i.e., -0.5-0=-0.5).

A zero-sum game is a game that under every combination of the players' pure strategies, the sum of all the players' payoffs equals zero, i.e., $\sum_{i=1}^{N} u_i(\prod_{p=1}^{N} s_p) = 0$. A game is called a non-zero-sum game, if there exists such a combination of players' pure strategies $\prod_{p=1}^{N} s_p$ that

$\sum_{i=1}^{N} u_i(\prod_{p=1}^{N} s_p) \neq 0$. Zero-sum games have some special properties, such as being easier to solve. It is worth noticing that the idea of duality theory in linear programming was originally developed from the study of two-player zero-sum games.

In a two-player game where each player has a finite set of pure strategies, if player 1 and 2 play mixed strategies x and y respectively, then the expected payoff for the players can be calculated as shown in Formula (2.2).

$$\begin{cases} EU_1 = \sum_{i=1}^{m} \sum_{j=1}^{n} x_i \cdot U_1(i,j) \cdot y_j \\ EU_2 = \sum_{i=1}^{m} \sum_{j=1}^{n} x_i \cdot U_2(i,j) \cdot y_j \end{cases} \quad \text{.........} \quad (2.2)$$

In which EU_1 and EU_2 are the expected payoffs for player 1 and for player 2 respectively; m and n denote the number of pure strategies of player 1 and player 2 respectively; U_1 and U_2 are the payoff matrices for player 1 and player 2 respectively. Both U_1 and U_2 have m rows and n columns, and entries (i, j) of these two matrices represent the player's payoff when player 1 plays his i^{th} pure strategy and player 2 plays her j^{th} pure strategy.

The assumption of 'Common knowledge'

The 'common knowledge' assumption in a game theoretic model assumes that each player in the game has global information of the game. A game where the 'common knowledge' assumption holds is called a 'complete information' game. In such a game, players know their own strategies and payoff functions and also those of the other players. In addition, each player knows that the other players have complete information. In reality, this is a very strong, perhaps in some cases much too strict, assumption.

For instance, in an RSP game, every player knows the rules of the game. In the illustrative game shown in Figure 2.4, a common knowledge assumption means that the figure is known to both the defender and the attacker, and they know whether the game is played simultaneously or sequentially.

The common knowledge assumption can also be interpreted in an iterative way:

1) the players know their own information (i.e., strategies and payoffs) and know how the game will be played;
2) the players also know all other players' information;
3) the players know that other players know their information;
4) and so forth.

The common knowledge assumption is satisfied in many famous games, such as in the rock/scissors/paper game, the chess game, and the Go game. Conversely, in most poker games (e.g., the Texas hold'em), the common knowledge does not hold.

In the security domain, the defender and the attacker are often modelled as the two players in the game. Defenders even have difficulties to exactly know their own information, for instance, to know how severe an attack could be. Therefore, the common knowledge assumption is rather a strong requirement for security game modellers.

Rios and Rios Insua [(49)] proposed an adversarial risk analysis (ARA) approach for relaxing the common knowledge assumption in the security game models. In the ARA framework, all the data are estimated by the defender. And she:

i) knows her own data u_d^0;
ii) estimates the attacker's data $u_a^0 \sim F_1$, uncertainties distribution F exist in this step and afterwards since the defender would not be able to know the exact numbers of these data;

iii) estimates the attacker's estimation of the defender's data $u_d^1 \sim F_2$;

iv) estimates the attacker's estimation of the defender's estimation of the attacker's data $u_a^1 \sim F_3$;

v) …

vi) At a certain step, presumes that the attacker behaves randomly.

This iterative estimation of data results in the fact that in a security game (or, in the ARA framework), the defender and the attacker are both behaving according to their own data as well as according to (what they think will be) their opponent's decision. In step i), the defender knows her own data. However, her optimal decision should also be influenced by the attacker's decision. Therefore, she has to move to step ii). In step ii), the defender has an estimation of the attacker's data. However, the attacker's decision should also be influenced by the defender's decision, thus the attacker's estimation of the defender's data is required, as stated in step iii). In theory, this iterative step will go deeper infinitely. In modelling practise, at a certain step, the defender presumes that the attacker plays randomly. Knowing the attacker's behaviour, the defender can then stop this data iteration, and move backwards by calculating both players' optimal strategies until step i) again.

The ARA framework theoretically relaxes the common knowledge assumption. However, it needs massive inputs (i.e., the u_d^0, F_1, F_2…) and these inputs are also difficult to obtain. Nonetheless, in a defender-attacker game, if the attacker would know the defender's strategy when he moves, the iteration would be stopped at step ii). At step ii), the defender knows the attacker's data $u_a^0 \sim F_1$ and she also knows that the attacker can observe her strategy, instead of guessing her strategy. Therefore, the defender can work out the attacker's behaviour, and then move back to step i), and combine this information with her own data, to play optimally.

The assumption of 'Rationality'

The assumption of rationality indicates that people always behave in their own best interests. A player with full rationality thus means that he/she is playing to maximize his/her own payoff. Kelly [50] argues that the assumption of rationality can be justified on a number of levels. At its most basic level, it can be argued that players behave rationally by instinct. However, experience suggests that this is not always the case, since decision makers frequency adopt simplistic algorithms which very often lead to sub-optimal solutions. Secondly, it can be argued that there is a kind of natural selection at work which inclines a group of decisions towards the rational and optimal. Successive generations of decisions are increasingly rational. Finally, it has been suggested that the assumption of rationality that underpins game theory is not an attempt to describe how players actually make decisions, but merely that they behave *as if* they were not irrational [51]. All theories and models are, by definition, simplifications and should not be dismissed simply because they fail to represent all realistic possibilities.

Near-rationality, or so-called 'bounded rationality', allows players to be rational, but only within certain limits. Players are allowed to play sub-optimal strategies as long as the payoff per iteration is within a certain (small) positive number of their optimal strategy. For instance, Pita et al. [52] studied a so-called 'epsilon-optimal' player in their security games. An 'epsilon-optimal' attacker is an attacker who would deviate from his optimal strategy to strategies that have close payoff to the payoff that he can obtain from the optimal strategy. Mckelvey and Palfrey [53, 54] proposed the Quantal Response Equilibrium, in which the players would play each pure strategy with a probability and the probability is calculated according to the payoff that the strategy can bring to the player. The $level - k$ thinking model [55, 56] has also been frequently studied in the game theory domain. A $level - k$ $(k = 0,1,2, \ldots)$ thinking player in game theory assumes that every other players in the game are $level-(k-1)$ thinking, and therefore the player plays optimally. Moreover, a $level-0$ player would choose strategies randomly, being totally irrational.

Both rationality and irrationality are well studied in the game theory framework and corresponding models and algorithms have been developed. However, the difficulties on handling irrationality are that how to decide which type of near-rationality model the players would follow and how to decide the parameters in such near-rationality models (e.g., the epsilon value in the 'epsilon-optimal' solution, the λ in the quantal response equilibrium, and the k in the $level-k$ thinking model). Nguyen et al. (57) employed simulated online games for learning the parameters in the quantal response equilibrium. Several real experiments have been carried out for studying the k for ordinary people, such as the beauty contest game (58).

Simultaneous and sequential game

As already mentioned, essentially two types of games are possible: (i) games where the moves of the players cannot be seen by the other players, hence these are hidden-move games which are also called 'simultaneous-move games', and (ii) games where the players make moves in some sort of order, hence these are transparent games which are also called 'sequential-move games' or dynamic games.

Table 2. 4. Strategic form of the simultaneous move game for the illustrative defend-attack game

		Defender	
		ND	D
Attacker	NA	0,0	0,-1
	A	2.5,-3	-0.5,-1

Table 2. 5. Strategic form of the sequential move game for the illustrative defend-attack game

		Defender	
		ND	D
Attacker	NA-NA	0,0	0,-1
	NA-A	0,0	-0.5,-1
	A-NA	2.5,-3	0,-1
	A-A	2.5,-3	-0.5,-1

The RSP game we mentioned in the beginning of this chapter, for instance, is a classic simultaneous game. Most table games, such as the chess game, the go game, and the Texas hold'em etc., are typical sequential games, since in these games, players are choosing their strategies with knowing their opponents' chosen strategies.

It is worth noting that the temporal order of choosing strategies does not determine whether a game is a simultaneous game or a sequential. Instead, only in case that when some players play first (being game leaders) and other players can observe these game leaders' played strategies (being game followers), the game is a sequential game. For instance, in the illustrative defend-attack game we discussed in section 2.2.1, the defender always moves first by deciding whether to defend or not, while the attacker follows by whether to attack or not. However, if the attacker cannot observe the defender's played strategy (i.e., either 'not defend' or 'defend'), then the game is a simultaneous game and the game tree is shown on the left-hand side of Figure 2.4. If the attacker knows the defender's played strategy when he makes his decision, then the game is a sequential game and the game tree is shown on the right-hand side of Figure 2.4. Table 2.4 and 2.5 further show the strategic forms for the simultaneous move game and for the sequential move game, respectively, of the illustrative defend-attack game.

2.2.2 Game theoretic models with a discrete set of strategies

2.2.2.1 Discrete and continuous set of strategies

A player's strategy set can either be discrete or continuous, depending on the modelling approach. In the defend-attack game illustrated in section 2.2.1, the defender's strategies were modelled as either "Defend" or "Not Defend" and the attacker's strategies were modelled as either "Attack" or "Not Attack", both thus being discrete sets of strategies. These discrete sets of strategies can be understood as defend and attack scenario based. For instance, the defender has only one security measure at her disposal, and she can decide whether to implement this measure to protect the target (i.e., "Defend") or not (i.e., "Not Defend"). Similarly, the attacker decides whether to use a specific attack scenario (e.g., a Vehicle-Born Improvised Explosive Device) or not.

Strategies of the game illustrated in section 2.2.1 can also be continuous. The players' strategies can be their defence and attacker efforts, for the defender and for the attacker respectively. These continuous strategies can be interpreted as resources-based. For example, the defender may have a maximal amount of security budget $\mathcal{B}$, and she can allocate arbitrary money $0 \leq b \leq \mathcal{B}$ to the target. The attacker's continuous strategies can also be interpreted in a similar way.

In this dissertation, we focus on game models with discrete strategy sets, since game theoretic models with continuous strategy sets suffer several drawbacks.

First of all, it is theoretically difficult to solve a game model with continuous strategy sets. Analytical approaches should be employed to solve these games. If the game becomes more complicated, there may be many variables and the difficulties of using analytical methods might increase dramatically. In the illustrative game in section 2.2.1, if there would be multiple targets – which is always the case in reality – and if the vulnerability of the target is a non-linear function of the defence effort and the attacker effort (e.g., by using a contest success function (59)) – which is often the case in reality – then a non-linear optimization problem with multiple variables will have to be solved in order to calculate the solution for the game. Furthermore, solutions are not guaranteed in these games, since the most famous solution of game theoretic models, which is, the Nash Equilibrium, is only guaranteed in a game with finite players and finite pure strategies (44). Conversely, if the game would be modelled with discrete strategy sets, a Nash Equilibrium always exists. The computation of a Nash Equilibrium in a finite security game is also difficult, as we stated in the beginning of this chapter. However, after Lemke and Howson's work (46), a lot of researchers have improved the algorithms that are capable to solve a finite game. Moreover, the computational capability of computers has improved dramatically over the past decades, and it keeps improving. Nowadays, we are able to solve game theoretic models with thousands of discrete strategies, and several security game based systems have been deployed in security practice, see for instance, Tambe and his co-authors' work (60).

Secondly, discrete strategies better reflect reality than continuous strategies. In practise, security performance is not a strictly increasing function of the security investments, instead, it is a stepwise function. The attack performance is analogous. The security or attack performance will not jump to a new level until the increment of the defence investment or attack effort reaches a certain level (e.g., a new scenario can be afforded to be included in the security policy). Hence, security policies are scenario-oriented. The defender firstly evaluates what gaps there are for the designed security level and the current security situation. In the meantime, possible attack scenarios are considered. Subsequently, the defender proposes corresponding defence scenarios. Cost and effectiveness analysis of these proposed scenarios are also carried out. The defender's strategies can therefore be discretely modelled as what kind of defence scenarios to deploy while the attacker's strategies can be discretely modelled as what kind of attack scenario to use. The cost effectiveness analysis procedure provides the information of calculating the payoffs related to the game.

Nonetheless, game theoretic models with continuous strategy sets have their roles in high level security investment problems. For instance, see Nikoofal and Zhuang's work on game theoretically allocating defensive budgets among different cities in the United States (61).

2.2.2.2 Nash equilibrium

The so-called 'Nash Equilibrium' (NE) is the most popular solution for non-cooperative games. In a non-cooperative game, players are not able to communicate with each other while their expected payoffs are determined by both their own strategy and other players' strategies. This situation brings a dilemma to players: what would be the most optimal strategy to play, in order to obtain the highest possible payoff?

A straightforward idea is that if one player would know other players' strategies, then he plays the strategy that can bring himself the highest payoff, or, in other words, he plays his best response strategies. For instance, in the game shown in Figure 2.4 (left-hand side), if the attacker knows that the defender plays a "ND", then the attacker's best response would be playing the strategy "A". However, if the defender would know that the attacker is playing an "A", then the defender's best response would be "D". Therefore, the answer to the dilemma is a group of strategies from each player, and each strategy in the group is a best response to the corresponding player with respect to all other strategies in the group.

A Nash Equilibrium (NE) is a set of players' strategies (one strategy per player), in which each player's strategy in the NE is the best response to all other players' strategies in the NE. For a two-player game, a pure strategy NE (i^*, j^*) satisfies the following condition:

$$U_1(i^*, j^*) \geq U_1(i, j^*), \forall i = 1,2, \dots, m \cdots\cdots (2.3)$$

and

$$U_2(i^*, j^*) \geq U_2(i^*, j), \forall j = 1,2, \dots, n \cdots\cdots (2.4)$$

While a mixed strategy NE (x^*, y^*) satisfies:

$$x^{*T} \cdot U_1 \cdot y^* \geq x^T \cdot U_1 \cdot y^*, \forall x \in X \cdots\cdots (2.5)$$

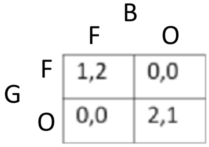
and

$$x^{*T} \cdot U_1 \cdot y^* \geq x^{*T} \cdot U_1 \cdot y, \forall y \in Y \cdots\cdots (2.6)$$

In other words, if players follow a Nash Equilibrium, then they are playing mutual best responses to each other.

The Nash Equilibrium is named after the American mathematician John Nash, since he proved the existence of the NE in any game with finite players and finite pure strategies (i.e., finite games). Finding a pure strategy NE in a finite game is easy while finding a mixed strategy NE can be quite difficult. Lemke and Howson [(46)] proposed a linear combinatorial algorithm for finding a mixed strategy NE for finite games with two players.

The NE is a theoretically perfect solution for a simultaneous game, since under the assumption of all players being rational, no player has the motivation to deviate from his/her NE strategy if the opponents do not either. In zero-sum games, the Nash Equilibria are interchangeable, thus there is no NE selection problem in such games, see also Proposition 2.1. However, when there is more than one NE in a two-player non-zero-sum game, it is impossible to predict which NE would be the outcome of the game. For instance, the battle of the sexes game (BoS), in which there are two players (a girl and a boy) and the girl prefers to go to the opera (O) while the boy prefers to play football (F). An illustrative payoff matrix for the BoS is shown in Figure 2.5, there are 2 pure strategy NE (i.e., (F,F) and (O,O)) and 1 mixed strategy NE (i.e., the girl plays x $= (1/3,2/3)$ which means that she agrees to play ‘F’ at probability 1/3 and to play ‘O’ at probability 2/3, and the boy plays y $= (2/3,1/3)$ which means that he agrees to play ‘F’ at probability 2/3 and to play ‘O’ at probability 1/3). It is obvious that the girl would prefer the (O,O) outcome while the boy would prefer the (F,F) outcome. In this case, the outcome of this game cannot be predicted by calculating the NE.



		B	
		F	O
G	F	1,2	0,0
	O	0,0	2,1

Figure 2. 5. A simple bi-matrix game with multiple Nash Equilibria (NE)

Proposition 2.1: Nash Equilibria in a two-player zero-sum game are interchangeable, that is to say, if mixed strategy pairs (x_1^*, y_1^*) and (x_2^*, y_2^*) are both NEs for a zero-sum game $G(U,-U)$, then strategy pairs (x_1^*, y_2^*) and (x_2^*, y_1^*) are also NEs for this game. Furthermore, the player’s payoffs for the different equilibria are the same.

Proof: (x_1^*, y_1^*) and (x_2^*, y_2^*) satisfy Formulas (2.7) to (2.10).

$$x_1^{*T} \cdot U \cdot y_1^* \geq x^T \cdot U \cdot y_1^*, \forall x \in X \tag{2.7}$$

$$x_1^{*T} \cdot (-U) \cdot y_1^* \geq x_1^{*T} \cdot (-U) \cdot y, \forall y \in Y \tag{2.8}$$

$$x_2^{*T} \cdot U \cdot y_2^* \geq x^T \cdot U \cdot y_2^*, \forall x \in X \tag{2.9}$$

$$x_2^{*T} \cdot (-U) \cdot y_2^* \geq x_2^{*T} \cdot (-U) \cdot y, \forall y \in Y \tag{2.10}$$

Formula (2.8) indicates $x_1^{*T} \cdot (-U) \cdot y_1^* \geq x_1^{*T} \cdot (-U) \cdot y_2^*$. Formula (2.9) indicates $x_2^{*T} \cdot U \cdot y_2^* \geq x_1^{*T} \cdot U \cdot y_2^*$. Formula (2.10) indicates $x_2^{*T} \cdot (-U) \cdot y_2^* \geq x_2^{*T} \cdot (-U) \cdot y_1^*$. Formula (2.7) indicates $x_1^{*T} \cdot U \cdot y_1^* \geq x_2^{*T} \cdot U \cdot y_1^*$. Together we have:

$$x_1^{*T} \cdot U \cdot y_1^* \leq x_1^{*T} \cdot U \cdot y_2^* \leq x_2^{*T} \cdot U \cdot y_2^* \leq x_2^{*T} \cdot U \cdot y_1^* \leq x_1^{*T} \cdot U \cdot y_1^* \tag{2.11}$$

Therefore, all above inequalities should be equal. Furthermore, we have:

$$x_1^{*T} \cdot U \cdot y_2^* = x_2^{*T} \cdot U \cdot y_2^* \geq x^T \cdot U \cdot y_2^*, \forall x \in X \tag{2.12}$$

$$x_1^{*T} \cdot (-U) \cdot y_2^* = x_1^{*T} \cdot (-U) \cdot y_1^* \geq x_1^{*T} \cdot (-U) \cdot y, \forall y \in Y \tag{2.13}$$

$$x_2^{*T} \cdot U \cdot y_1^* = x_1^{*T} \cdot U \cdot y_1^* \geq x^T \cdot U \cdot y_1^*, \forall x \in X \tag{2.14}$$

$x_2^{*T} \cdot (-U) \cdot y_1^* = x_2^{*T} \cdot (-U) \cdot y_2^* \geq x_2^{*T} \cdot (-U) \cdot y, \forall y \in Y$ ································ (2.15)

Thus, (x_1^*, y_2^*) and (x_2^*, y_1^*) are also NEs for game G. Formula (2.11) ensures that players will receive the same payoff for the different equilibria.

Proposition 1 is an important property of zero-sum games since in case there are multiple NEs in the game, the player can play any equilibrium. A non-zero-sum game does not have this property, see Figure 2.5 for instance (the reason is given in the first paragraph of section 2.2.2.3).

2.2.2.3 Stackelberg equilibrium

Simultaneous games may have multiple Nash Equilibria (NE). Therefore, a new dilemma arises, that is, the NE selection problem. One solution to this new dilemma is that some players of the game receive a capability to move first (being the so-called 'game leaders'), and other players follow (being the so-called 'game followers') with knowing the leaders' strategies. For instance, the battle of the sexes game shown in Figure 2.5 has 3 NEs. Player G prefers the equilibrium (O,O) which brings her a payoff of 2 and in contrast, Player B prefers the equilibrium (F,F) which brings him a payoff of 2. If both players play their preferable NE, i.e., the girl plays strategy "O" and the boy plays strategy "F", then the players are no longer playing a NE and both of them would obtain a payoff of 0. If the girl could move first, for instance, buy two opera tickets for herself as well as for the boy before the battle, then the boy knows that the girl is definitely going to the opera, therefore he will also follow since then his best response is to follow the girl.

A two-player game is called a Stackelberg game if one player moves first and another player follows the leader, knowing the leader's strategy. In the Stackelberg game, the game follower knows the leader's strategy, and therefore is able to play his/her optimal strategy. The leader also knows that the follower knows his/her strategy and the follower will try playing optimally, therefore the leader can also play accordingly.

A so-called 'Strong Stackelberg Equilibrium' (SSE) models the above procedure, and can be formulated as:

$\overline{y} = \max_{y \in Y} U_l(x(y), y)$ ··· (2.16)

$x(y) = \max_{x \in X} U_f(x, y)$ ··· (2.17)

Formula (2.17) denotes that knowing the leader's committed strategy y, the follower would play his best response strategy $x(y)$. Formula (2.16) indicates that the leader can also work out the $x(y)$, and therefore is able to play optimally. $U_l(x, y)$ and $U_f(x, y)$ denotes the leader and the follower's payoffs in case that the follower plays an "x" and the leader plays a "y".

The equilibrium is called 'Strong Stackelberg Equilibrium' instead of just 'Stackelberg Equilibrium', due to the fact that the so-called "breaking-tie" assumption is involved. The assumption requires that, when the follower has multiple best response strategies to the leader's committed strategy y (we say there is a tie for the follower), the follower would play the best response strategy that maximizes the defender's payoff (we say the follower breaks the tie preferably for the leader). The Stackelberg Equilibrium is not unique as well without the "breaking-tie" assumption.

For instance, for the game shown in Figure 2.4, if the defender is the game leader and she plays "NotDefend" at probability $1/6$ and "Defend" at probability $5/6$, then both the

"NoAttack" and the "Attack" would be the attacker's best response. In this case, both strategies would bring the attacker a payoff of 0 and on the contrary, the defender's payoff would be $-5/6$ if the attacker responds "NoAttack" and it would be $-8/6$ if the attacker responds "Attack". Under the "breaking-tie" assumption, the attacker would play the "NoAttack" strategy.

However, the "breaking-tie" assumption is anti-intuitive if applied in the security domain. In security, the defender is usually considered as the game leader, and the attacker is the follower. The "breaking-tie" assumption then means that the attacker is playing preferably before the defender! To fix this drawback, von Stengel and Zamir (62) point out that the game leader can deviate a little bit from her SSE strategy, promoting her preferable strategy to be the unique best response to the game follower. For instance, in the above example, the defender may play a mixed strategy $x = (0.99/6, 5.01/6)$ so that the attacker's best response becomes unique, and it is "NoAttack".

2.3 Research positioning

2.3.1 Drawbacks of current security vulnerability assessment methodologies

The current methodologies used for security risk assessment within the chemical and process industry mainly suffer from two drawbacks: (i) they are all qualitatively based and (ii) they fail to model intelligent interactions between the defender and the attacker.

Qualitative models can only inform industrial managers about which part of the plant needs to be better protected and it does not mention how many improvements are needed. Ideally, the defender needs quantitative guidance to make decisions, such as how to allocate the limited security resources. Qualitative models can also be theoretically not sound. For instance, Cox (63) lists several theoretical limitations of the security risk assessment methodologies which are based on the "$risk = threat \times vulnerability \times consequence$" formula. Zhang et al. (64) suggests several further impediments that the API SRA methodology needs to pay more attention to.

Despite the drawback of models being qualitative, being not able to model dynamic interactions between the defender and the attacker is the most important and essential problem of the above mentioned conventional security risk assessment methods. As also mentioned by Baybutt (7), these conventional security methods are mostly derived from safety risk management methods and can be compared with PHAs. Therefore, security risks are calculated by using a "$risk = probability \times consequence$" approach. However, adversaries related to security risks are to be considered as 'intelligent opponents', thus not acting randomly or probabilistically. Instead, intelligent attackers behave according to their goal and also based on the difficulties of reaching their goal. See in Figure 2.1 that how the security methodology without considering intelligent attackers can mislead allocation of security resources.

Game theory, which originated in economic sciences, is a good choice to handle problems that contain intelligent players. Game theory has very rigorous mathematical foundations, and if adequately used with respect to chemical security, we can obtain more accurate and more defensible quantitative results, besides the qualitative assessments and results used nowadays in chemical plant security management. In recent years, a lot of attention in academia has been laid on the combination of game theory and critical infrastructure protection. Tambe and his group (60) used game theory to improve the security situation in airport patrolling, air marshals' allocation, and coast line protection. They developed several decision support systems based on their research, and these systems now work in reality. Bier and her group (65) studied the combination of game theory and security assessment methods from a theoretical viewpoint. They answered the questions why game theory has an important role in security research, and illustrated the advantages and disadvantages of using game theory in operational security.

Although there are already some researches on using game theory to improve operational security, in fact in the chemical process security, very scarce research has been done as yet. Security problems in the process industries are different to those in aviation or the electric power grid for example, although they are all critical infrastructures. We cannot readily apply game theoretical models now being used in aviation, within the process industries directly. Different security models are implemented in different types of industries. For instance, in Tambe's model, air marshals are allocated to defend an air plane, and therefore the players' strategies are limited to "protect" (that is, to allocate an air marshal on the plane) or not (that is, no air marshal on the plane), and "attack" or not. However, in case of security in the process industries, the model is more complex, the strategies may be at a different alert level (discrete model) or at a different investment level (continuous model).

2.3.2 Criticisms on game theoretic models for security improvement

A special type of game theoretic models developed for the purpose of improving security, in which what is better for one player is bad for the other player, is defined as Security Game [(60)]. Security games have been widely studied in academia, and several security game based systems have been deployed in reality [(60)]. However, criticisms do exist.

Some security game models are criticized as 'magic mathematical games' due to sometimes unrealistic assumptions. Most researchers agree that (human) adversaries would plan and implement attacks adaptively. However, whether adversaries are rational (i.e., aiming at maximizing their payoff) is still a topic under study. Researchers also realize that security risk management involves huge uncertainties such that the 'common knowledge assumption' would not hold. For a more detailed discussion of these criticisms, interested readers are referred to Guikema [(66)].

Besides its possible unrealistic assumptions, game theoretic modelling is also criticized for its requirements with respect to quantitative input. As illustrated in Figure 2.4, parameters such as the defender's defence cost (C_d), the defender's loss from a successful attack (L), the attacker's attack cost (C_a), the attacker's penalty (P) from a failed attack, and the attacker's reward from a successful attack (R) should be provided in order to analyse the game. In practice, however, it can be quite difficult (almost impossible) to obtain these exact data. Let us take as an example R, which denotes the attacker's gain from successfully attacking the target: it is not possible, in practise, to know what would be the exact gain for the attacker, since it is largely dependable on the attacker's perception. In literature, the Chemical Plant Protection game proposed by Zhang and Reniers [(67)] requires quantitative data such as the success probabilities and consequences of an attack under any given attack scenarios and any given defence plans, from both the defender and the attacker's point of view. In the work of Feng et al. [(68)], the defender needs to know a prior probabilities of occurrence of different types of attackers, and also attackers' estimations of vulnerabilities and consequences under each of the players' strategy pairs. These above mentioned quantitative inputs are very difficult to obtain.

2.3.3 Integrating conventional SVA methodologies and game theory for improving chemical plant protection

Conventional security risk assessment methodologies, such as the SRFT and the API SRA methodology, being developed by chemical security experts and practitioners, are systematic and practically implementable for security risk assessment in the process and petrochemical industries. These methodologies, since released, have been extensively used in industrial practice and have been much referred to in academic research. A common drawback of these methodologies is their failure on modelling the intelligent interactions between the malicious attackers and the security defender.

Game theory was created to deal with intelligent interactions among multiple strategic actors, while its drawback on the application in the security domain is that it is too far away from the security practise. Security experts and practitioners are not familiar with game terminologies and they do not like the complex mathematical formulas in the development procedure of the games.

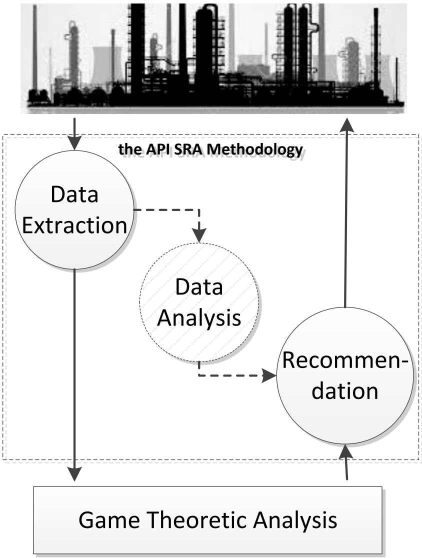


Figure 2. 6. A framework of integrating the API SRA methodology and game theory

Zhang et al. [(64)] proposed an approach for integrating the conventional security risk methods in the chemical security domain and game theoretic models, as shown in Figure 2.6. In their approach, conventional security methods (e.g., the API SRA methodology) act as a bridge between the industrial practise and the game modellers. At the first step, the conventional security method should be employed, for screening the chemical plant, identifying critical assets, evaluating threats/vulnerabilities/consequences etc. After the first step, a certain set of security related data will be obtained, for instance, see Figure 2.3 for the output data of the API SRA method. At the second step, instead of analysing the output data from the first step by using risk matrices or by using risk ranking systems (as used in conventional security methods), game theoretic analysis are introduced to deal with these data (output from the first step). At the third step, after analysing the data in a game theoretic approach, we must not show the game theoretic results in game terminologies to security experts and practitioners directly. Alternatively, the game theoretic results should be translated back to the conventional security risk assessment terminologies, for instance, by reflecting the attacker's mixed equilibrium strategy (in game theory terminology) to the target's attractiveness to the attacker (in API SRA terminology).

2.4 Conclusions

The protection of single chemical plants, small and large chemical clusters, as well as pipelines, have been an important task for risk analysts. The chemical industry, on the one hand fulfils an extremely important role for our modern lives, but on the other hand it poses huge threat to modern society. If installations storing toxic, flammable, or explosive materials would be damaged by intentional attacks, the consequences would be awful. Moreover, the two attacks on chemical plants in June and July 2015 in France proved the possibility of an attack to the chemical industry in the Western world.

There are plenty of academic studies concerning the protection of chemical installations. Also, regulations, standards, and guidelines on promoting chemical security have been published, especially in the U.S. However, due to the lack of historic data and the failure to model intelligent interactions between the malicious attackers and the defenders, the current researches and regulations etc. have their drawbacks. Moreover, there is a lack of effort with respect to the protection of chemical clusters, which, if being strategically attacked, may result in truly catastrophic consequences for society.

Game theory, being able to support strategic decision making, has been successfully applied in several domains for improving security. Hall Jr (69) (2009) mentions that “If the conditions creating the problems you had to deal with were natural or random, the answer was decision analysis (which looked a lot like what we now call risk analysis). If the conditions creating the problems you had to deal with were the result of deliberate choice, the answer was game theory.” Therefore, we conclude that game theory has the potential to be a proper methodology for improving security in the chemical and process industries.

References

[1] 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.

[2] Zhou J, Reniers G, Zhang L. Petri-net based attack time analysis in the context of chemical process security. Submitted to Reliability Engineering and System Safety (under review). 2018.

[3] Database GT. 2016. Available from: https://www.start.umd.edu/gtd/.

[4] Reniers G, Cozzani V. Domino Effects in the Process Industries: Modelling, Prevention and Managing: Elsevier B.V.; 2013. 1-372 p.

[5] 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.

[6] (DHS) DoHS. National strategy for homeland security. 2002.

[7] Argenti F, Bajpai S, Baybutt P, Cozzani V, Gupta J, Haskins C, et al. Security Risk Assessment: In the Chemical and Process Industry: Walter de Gruyter GmbH & Co KG; 2017.

[8] Luft G. Pipeline sabotage is terrorist's weapon of choice. Pipeline & gas journal. 2005;232(2):42-4.

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

[10] Reniers G, Khakzad N. Revolutionizing safety and security in the chemical and process industry: applying the CHESS concept. Journal of Integrated Security Science. 2017;1(1).

[11] Powell R. Defending against terrorist attacks with limited resources. American Political Science Review. 2007;101(03):527-41.

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

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

[14] Baybutt P. Inherent security: Protecting process plants against threats. Chemical Engineering Progress, accepted for publication. 2003.

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

[16] Baybutt P. Security vulnerability analysis : protecting process plants from physical and cyber threats. In: G Reniers, N Khakzad, Gelder Pv, editors. Security Risk Assessment: In the Chemical and Process Industry. 1: Walter de Gruyter GmbH & Co KG; 2017.

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

[18] Baybutt P. Process security management systems: Protecting plants against threats. Chemical Engineering. 2003;48.

[19] Baybutt P. Comprehensive cyber security vulnerability analysis for manufacturing plants. Hydrocarbon Engineering. 2005;10(1):12-8.

[20] Baybutt P. The role of people and human factors in performing process hazard analysis and layers of protection analysis. J Loss Prev Process Ind. 2013;26(6):1352-65.

[21] Baybutt P. Cyber security vulnerability analysis: An asset-based approach. Process Saf Prog. 2003;22(4):220-8.

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

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

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

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

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

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

[28] Reniers G. Terrorism security in the chemical industry: Results of a qualitative investigation. Secur J. 2011;24(1):69-84.

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

[30] 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.

[31] Reniers G, Soudan K. A game-theoretical approach for reciprocal security-related prevention investment decisions. Reliab Eng Syst Saf. 2010;95(1):1-9.

[32] 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.

[33] Reniers GLL. Safety and security decisions in times of economic crisis: Establishing a competitive advantage. Chemical Engineering Transactions: Italian Association of Chemical Engineering - AIDIC; 2014. p. 1-6.

[34] Reniers GLL. Security within the chemical process industry: Survey results from flanders, belgium. Chemical Engineering Transactions: Italian Association of Chemical Engineering - AIDIC; 2012. p. 465-70.

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

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] Cox LAT, Babayev D, Huber W. Some limitations of qualitative risk rating systems. Risk Anal. 2005;25(3):651-62.

[42] Baybutt P. Designing risk matrices to avoid risk ranking reversal errors. Process Saf Prog. 2016;35(1):41-6.

[43] Von Neumann J, Morgenstern O. Theory of games and economic behavior: Princeton university press; 2007.

[44] Nash JF. Equilibrium points in n-person games. Proc Nat Acad Sci USA. 1950;36(1):48-9.

[45] Harsanyi JC. Games with incomplete information played by "Bayesian" players, i–iii: part i. the basic model&. Management science. 2004;50(12_supplement):1804-17.

[46] Lemke CE, Howson J, Joseph T. Equilibrium points of bimatrix games. Journal of the Society for Industrial and Applied Mathematics. 1964;12(2):413-23.

[47] Chen X, Deng X, editors. Settling the complexity of two-player Nash equilibrium. Foundations of Computer Science, 2006 FOCS'06 47th Annual IEEE Symposium on; 2006: IEEE.

[48] Nisan N, Roughgarden T, Tardos E, Vazirani VV. Algorithmic game theory: Cambridge University Press Cambridge; 2007.

[49] Rios J, Insua DR. Adversarial risk analysis for counterterrorism modeling. Risk Anal. 2012;32(5):894-915.

[50] Kelly A. Decision making using game theory: an introduction for managers: Cambridge University Press; 2003.

[51] Friedman M. Essays in positive economics: University of Chicago Press; 1953.

[52] Pita J, Jain M, Tambe M, Ordóñez F, Kraus S. Robust solutions to Stackelberg games: Addressing bounded rationality and limited observations in human cognition. Artificial Intelligence. 2010;174(15):1142-71.

[53] McKelvey RD, Palfrey TR. Quantal response equilibria for extensive form games. Experimental economics. 1998;1(1):9-41.

[54] McKelvey RD, Palfrey TR. Quantal response equilibria for normal form games. 1993.

[55] McLay L, Rothschild C, Guikema S. Robust adversarial risk analysis: A level-k approach. Decision Analysis. 2012;9(1):41-54.

[56] Rothschild C, McLay L, Guikema S. Adversarial risk analysis with incomplete information: A level-k approach. Risk Anal. 2012;32(7):1219-31.

[57] Nguyen TH, Yang R, Azaria A, Kraus S, Tambe M, editors. Analyzing the Effectiveness of Adversary Modeling in Security Games. AAAI; 2013.

[58] Nagel R. Unraveling in guessing games: An experimental study. The American Economic Review. 1995;85(5):1313-26.

[59] Skaperdas S. Contest success functions. Economic theory. 1996;7(2):283-90.

[60] Tambe M. Security and game theory: algorithms, deployed systems, lessons learned: Cambridge University Press; 2011.

[61] Nikoofal ME, Zhuang J. Robust allocation of a defensive budget considering an attacker's private information. Risk Anal. 2012;32(5):930-43.

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

[63] Cox Jr LAT. Some limitations of "Risk= Threat× Vulnerability× Consequence" for risk analysis of terrorist attacks. Risk Anal. 2008;28(6):1749-61.

[64] 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.
[65] Bier VM, Azaiez MN. Game theoretic risk analysis of security threats: Springer Science & Business Media; 2008.
[66] Guikema SD. Game theory models of intelligent actors in reliability analysis: An overview of the state of the art. Game theoretic risk analysis of security threats: Springer; 2009. p. 13-31.
[67] Zhang, Reniers. A Game-Theoretical Model to Improve Process Plant Protection from Terrorist Attacks. Risk Anal. 2016;36(12):2285-97.
[68] Feng Q, Cai H, Chen Z, Zhao X, Chen Y. Using game theory to optimize allocation of defensive resources to protect multiple chemical facilities in a city against terrorist attacks. J Loss Prev Process Ind. 2016;43:614-28.
[69] Hall Jr JR. The elephant in the room is called game theory. Risk Anal. 2009;29(8):1061-.

3

DAMS: A MODEL TO ASSESS DOMINO EFFECT BY USING AGENT-BASED MODELLING AND SIMULATION

Historical data analysis shows that escalation accidents, so-called domino effects, have an important role in disastrous accidents in the chemical and process industries. In this chapter, an agent-based modelling and simulation approach is proposed to study the propagation of domino effects in the chemical and process industries. Different to the analytical or Monte Carlo simulation approaches, which normally study the domino effect at probabilistic network levels, the agent-based modelling technique explains the domino effects from a bottom-up perspective. In this approach, the installations involved in a domino effect are modelled as agents while the interactions among the installations (e.g., by means of heat radiation) are modelled via the basic rules of the agents. Application of the developed model to several case studies demonstrates the ability of the model not only in modeling higher-level domino effects and synergistic effects but also in accounting for temporal dependencies. The model can readily be applied to large-scale complicated cases.

This chapter has been published as the following paper:

Zhang, L., Landucci, G., Reniers, G., Khakzad, N., & Zhou, J. (2017). DAMS: A Model to Assess Domino Effects by Using Agent - Based Modeling and Simulation. Risk analysis.

3.1 Introduction

Domino effects have been responsible for some catastrophic accidents which occurred in the chemical and petrochemical industries.(1-5) Although there are multiple definitions of domino effects in the chemical and process industries,(1, 6, 7) this type of accidents features a generic schematization with the following elements: 1) there is a "primary event", initiating the domino effect; 2) there is an escalation vector (e.g., fire impingement, heat radiation, explosion overpressure, etc.), facilitating the propagation of the domino effect; 3) one or more secondary accident events, involving one or more target equipment.(8, 9) In the second element, the influence of synergistic effects should be considered to account for the occurrence of multiple accident scenarios. Through synergistic effect, the escalation vectors of concurrent events are superimposed to identify the possibility of causing damage to other target equipment.

Domino effect occurred several times in the chemical and process industry, featuring high destructive potential.(10) Kourniotis et al.(11) examined 207 major chemical accidents and found that 114 of them involved a domino effect. The existence of domino effects make assets in process plants dependent on each other, resultant in a systemic risk.(12) The potential severity of such accident scenarios led to important efforts for the prevention of domino effects,(3, 13, 14) and also made relevant technical standards and legislation take into account measures to assess, control, and prevent domino effects.

Cozzani et al.(7, 8) developed a methodology for risk assessment based on the adoption of vulnerability models, which relied on simplified modeling and characterization of the escalation vectors. Khan and Abbasi(15, 16) synthesized the quantitative methodologies used in domino effects estimation, and developed a software named "DOMIFFECT" to support the domino effect estimation in complicate situations. Reniers and Dullaert(17) developed a software named "DomPrevPlanning" to support decision making on safety barriers to prevent/mitigate domino effects in complex chemical installations, which succeed in considering multiple domino scenarios. Abdolhamidzadeh et al.(9, 18) developed an algorithm named "FREEDOM" based on Monte Carlo Simulation to assess domino effects. Khakzad et al.(19-21) developed a methodology based on Bayesian network both to probabilistically simulate the propagation of domino effects and to identify the most likely sequence of events in a potential domino effect. In their methodology, both the possibility of higher-order domino effects and the influence of synergistic effects were taken into account.

Nevertheless, despite the relevant progress made in the framework of domino effect understanding and modeling, the time dimension and evolution of domino effects, which is critical for emergency preparedness and response,(22) is not systematically accounted for. Khakzad et al.(23) developed a dynamic Bayesian network (DBN) methodology to capture both spatial and temporal propagation of domino effects. However, in application of large-scale cases, the DBN model needs a combinatorial-increasing number of conditional probabilities. Furthermore, the DBN model uses a discrete time scheduling method, which has been proofed to be not efficient. To this end, the extension of DBN models feature a high level of complexity and demand relevant computational resources for the extension to realistic industrial cases, featuring the simultaneous analysis of dozens of units.

In the present work, an agent-based modelling and simulation model – DAMS – is proposed to support domino effect analysis in the chemical and process industries. The aim of this study is to provide a quick yet effective tool for the chemical and process industries to support the emergency response and mitigation strategies. The model is applied to i) a demonstration case study for verification purpose; ii) a real industrial setting for illustrating the implementation of the model; and iii) an intentionally constructed large scale case study to investigate computational issues. Furthermore, a discussion on computation resources and potential application is also addressed.

3.2 Agent-based modelling and simulation

Agent-based Modelling and Simulation (ABMS) is a bottom-up approach to study complex systems.(24) Given a system, instead of modelling the patterns, structures, and the system behaviors, the ABMS approach focuses on the basic units (namely, the "agents") of the system, including their attributes and interactions.(24) By performing computational experiments on the agent models, the response and behavior of the global system may be derived.(25) Several examples of ABMS applications in different disciplines and contexts are available in the literature.

Epstein and Axtell(26) proposed an agent-based social simulation model, named Sugarscape, in which multiple agents move, interact, and behave in order to get sources (i.e. sugar). In Epstein and Axtell's seminal book,(26) by defining simple but different rules for the agents, some complex social phenomena such as groups, war, trading, etc. emerged in the Sugarscape model. Since the Sugarscape, ABMS has been widely used in social science as well as complex system studies and prediction of pandemics,(27) economic crisis management,(28) and manufacturing.(29) In recent years, ABMS has also been used in the risk analysis domain; some examples among others are the analysis of hurricane evacuation procedures,(30) flood incident management,(31) and defensive resources allocation for spatially distributed networks.(32) However, the potential application in the framework of industrial safety, and especially in relation to domino effect assessment, is innovative and is discussed in this study.

In ABMS, an agent model normally consists of several static attributes and several simple rules. The rules and the interactions of agents should not be complex, since simple individual behaviors and interactions can already generate complex system behaviors.

In the case of the application to domino effect assessment, the accident propagation and evolution may be considered as a behavior of the system, resulting from the interactions (heat radiation propagation and/or overpressure following accidents) of items (target equipment, such as tanks, pipelines, etc.) in the industrial areas. For this purpose, the behavior and interactions of the items should be reproduced with simple rules, which is normally not the case for domino targets. In fact, several studies pointed out the complicating physical phenomena associated with equipment exposed to fire,(33, 34) overpressure,(35, 36) and missile projection.(37, 38) Specific advanced model tools such as Finite Elements Modeling(39) or computational fluid dynamics(40) may be required for a comprehensive detailed assessment. Nevertheless, previous research(33, 41) was dedicated to the development of simplified approaches aimed at reproducing the behavior of target equipment exposed to a given escalation vector. Such simple rules and models may be adopted in order to trace the behavior and interaction of single items. Even though the behavior of target equipment during domino effects can be simplified, the domino effect itself is still quite complicated due to several reasons: (i) the probabilistically propagation; (ii) the synergistic effects; (iii) the dynamic evolution. Therefore, ABMS features a suitable approach to support the analysis of complex domino effects.

3.3 Model description

3.3.1 Overview

Among others, the AnyLogic software group, which is one of the most successful groups in developing simulation platforms in the world, defines ABMS as: "from the viewpoint of practical applications agent based modelling can be defined as an essentially decentralized, individual-centric (as opposed to system level) approach to model design. When designing an agent based model the modeller identifies the active entities, the agents (which can be people, companies, projects, assets, vehicles, cities, animals, ships, products, etc.), defines their behaviour (main drivers, reactions, memory, states, ...), puts them in a certain environment, establishes connections, and runs the simulation."[42] Storage tanks are the most frequently involved items in domino effects in the chemical and process industries.[4, 43] Some global information such as weather, geography etc. also act as an important role in domino effects. To this end, active entities in the **D**omino effect assessment by **A**gent-based **M**odelling and **S**imulation (DAMS) model are the tanks and the environment, as shown in Figure 3.1. In the following sections, Figure 3.2 illustrates the static model of the tank agents; Figure 3.4 depicts the tank agent's behavior model; The environment model is given in Figure 3.5; Connections (or interactions) among agents and between the agent and the environment model, are given in Figure 3.4 and Figure 3.6 respectively; Finally, the DAMS model is implemented on 3 case studies, and Monte Carlo Simulation is employed to get the statistic results.

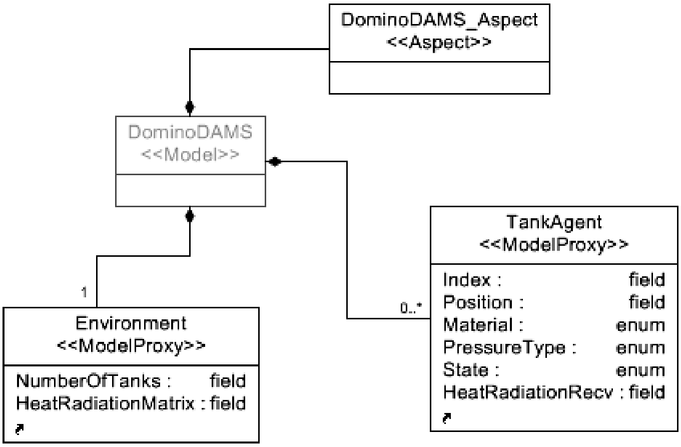


Figure 3. 1. Framework of the ABMS model for supporting Domino Effect assessment

In the following, the presented framework is explained in more details. For illustrative purposes, the analysis is devoted to domino effects triggered by fire, but the approach can be extended to consider domino effects triggered by overpressure.

3.3.2 Tank agent model

3.3.2.1 Static model

Storage tanks (tank agents) are the main agent type involved in the domino effect chain. Because modelling is an abstract form of reality, it cannot capture all the properties of the concerned object; thus, only the properties which are relevant to the modelling and simulation goal should be taken into consideration. Therefore, only the domino effect related attributes of a tank are considered in the tank agent models (and the same for other models in this study). All these attributes should be initialised at the beginning of the simulation, given the concerned process plant. The static model of the tank agent is shown in Figure 3.2, modelled in the software Generic Modelling Environment (GME).[44]

GME is a configurable toolkit for creating domain-specific modelling and program synthesis environments. It provides interfaces for secondary developers, enabling the users to define their own domain specific model libraries. In this study, all the static models are graphically shown in the GME while the implemented simulations are coded in C++. GME provides interfaces for the model developer to link their models coded in some programming languages (e.g., C++) and the graphic icons in the GME.

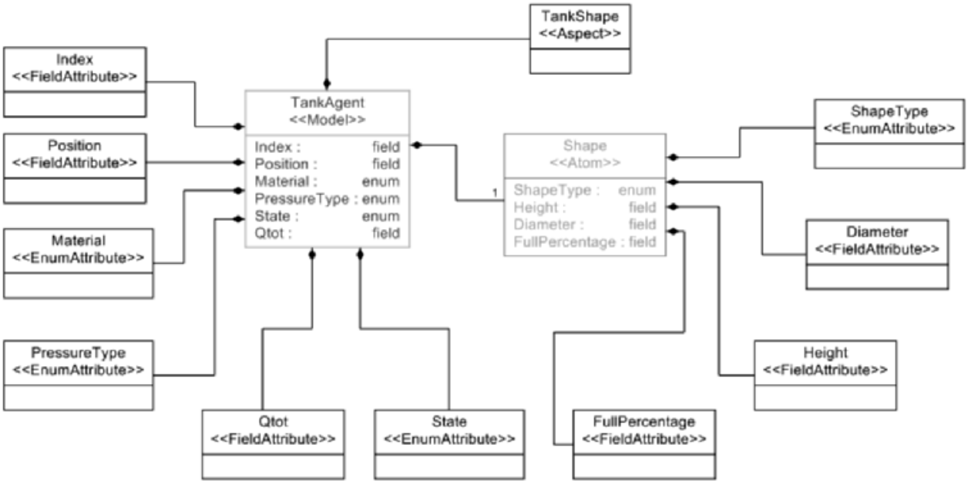


Figure 3. 2. Static model of Tank Agent

As shown in Figure 3.2, the Tank Agent model consists of several attributes:

- Index: unique identity of the tank;
- Position: coordinates of the tank;
- Material: nature of the chemical substance stored in the tank, such as Benzene, Gasoline, etc.;
- PressureType: indicating whether the tank is an “Atmospheric” or “Pressurized” tank;
- Q_{tot}: the accumulated heat radiation that the tank received from other tanks, more explanation of this attribute is given in the following paragraph;
- State: the state of the tank, as summarized in Table 3.1;
- Shape: shape information of the tank, consisting of:
  - ShapeType: indicating if the tanks is “Vertical Cylinder”, or “Horizontal Cylinder”, or “Spherical”;
  - Diameter: diameter of the tank;
  - Height: the height (length) of “Vertical (Horizontal) Cylinder” tank; this attribute does not apply to “Spherical” tank;

- FullPercentage: indicating how full (%) the tank is.

In case at least one tank within the area of interest (AOI)[1] is in failure mode, leading to a primary fire scenario such as a pool fire, one or more neighbouring tanks receive heat radiation. Heat radiation exposure may cause a structural damage on each tank. This in turn can result in secondary fires; as a result, possible synergistic effects between the primary and secondary fires should be taken into account. In fact, the resultant heat radiation received by a given target may be increased by the superimposition of the heat radiation of simultaneous fire. This is particularly relevant when a target tank does not receive enough heat radiation to reach failure conditions but, after the secondary failure of other equipment, the resultant heat radiation on the target is increased due to synergistic effects, increasing the possibility of failure.

Table 3. 1. **Definition of the variables which characterize the tank agent dynamic behaviour**

State	Description
Normal	The tank operates in normal conditions: initial state
Heat-up	The tank is not physically damaged, but due to heat radiation received from external fire the wall and the contains are heating up with consequent pressurization and incipient failure due to structural weakening
Leaking	The tank is physically damaged, thus hazardous materials are released. It is supposed that storage units are provided with a catch basin, which ensures the full containment of the released liquids
Fire	The tank is already on fire in the catch basin.

3.3.2.2 Dynamic model

The tank agent has four possible states during the evolution of a domino effect scenario, as summarized in Table 3.1.

In a domino effect, the tank agent can probabilistically transfer from one state to another state, as we will show later in this section. To this end, the formal modelling formalism named “probabilistic statechart” is employed to describe the tank agent’s dynamic model.

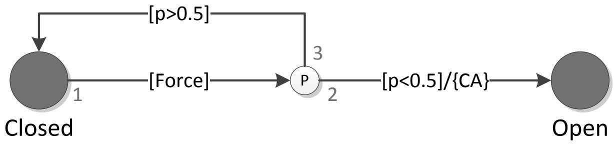
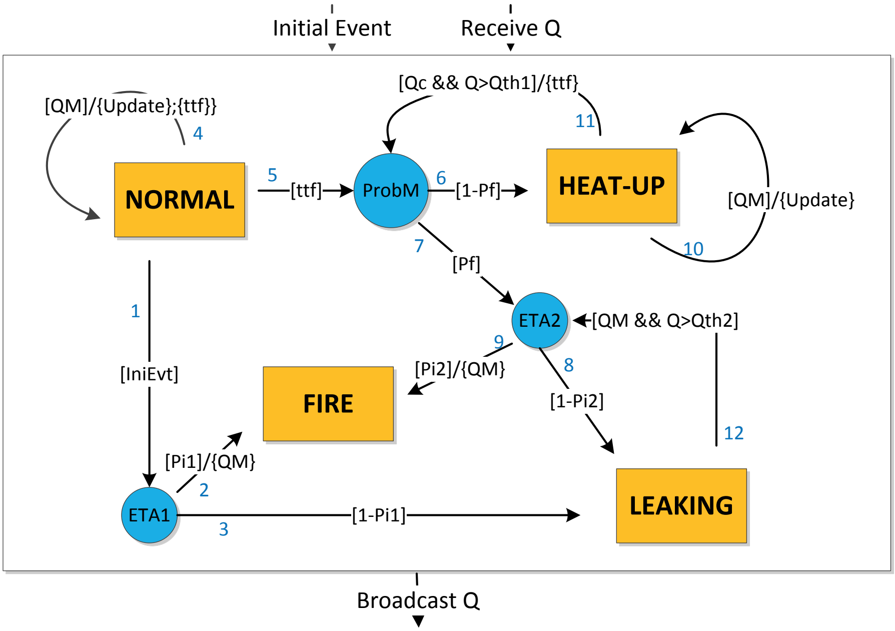


Figure 3. 3. An Illustrative p-statechart model showing door opening

A probabilistic statechart [(45)] is commonly adopted in order to show the probabilistic transition among the different states. A sample p-statechart showing the inner model of a door is depicted in Figure 3.3.

[1] Area of Interests (AOI), defined by the U.S. military, is the area of concern to the commander. Hereby we use it to describe the areas affected by the domino effects.



* “Broadcast” indicates that the ‘on Fire’ tank will transmit heat radiation to all other tanks within the AOI.

Figure 3. 4. Tank Agent – the dynamic model statechart

Table 3. 2. Description of agent behaviours and parameters depicted in Figure 3.4

Name	Definition	Details
QM	Heat radiation propagation message	The sender of this message is the tank in “FIRE” state (transitions 2 and 9); the receivers of this message are all the other tanks which are not in “FIRE” state (transitions 4, 10, and 12).
UPDATE	Update Q_{tot}	Updating Q_{tot} by adding the received heat radiation.
Qc	Q_{tot} changes	-
ttf	Time to failure	The time to failure is the time lapse between the start of the fire and the failure of a target vessel. It is computed through the simplified correlations reported in Table 3.3 and APPENDIX A.
P_f	Damage probability	This probability is evaluated according to the procedure summarized in Table 3.3.
IniEvt	Initial events	Internal process failure, due to corrosion, accidental pressurization, operator mistakes, etc.[46]
P_i	Probability of immediate ignition	P_{i1}: to be selected when there is no tank on fire within AOI. P_{i2}: to be selected when at least one tank is on fire within AOI.
Q_{th1},Q_{th2}	Threshold of heat radiation	$\mathrm{Q_{th1}}$: when the tank is already in a HEAT-UP state, the threshold of judging whether to use the Vulnerability Model. $\mathrm{Q_{th2}}$: when the tank is LEAKING, the threshold of judging whether to use ETA2. In this study, we set both Q_{th1} and Q_{th2}equal to a constant small number, as $Q_{th1} = 1kw/m^2$ and $Q_{th2} = 1kw/m^2$.

The arrows between states (modules) represent the possible transitions from the initial state to the end state. The content in the square bracket assigned to the transition represents the condition of the transition; the content in the braces represents the actions while the transition happens. For example, only when the condition of "Force" (i.e. someone is opening the door) and "p < 0.5" (i.e., he is on the right direction) are satisfied, the transitions "1" and "2" will happen, and if they happen, the door will Compute the opening Angel by "CA". Note that the door is only characterised by a "Closed" or "Open" state; the "P" module is just a judging point.

The states shown in Table 3.1 are implemented in the probabilistic statechart, adopted for the modelling and simulation of domino effects, as illustrated in Figure 3.4. Table 3.2 describes in detail all the conditions and actions in the model.

As shown in Figure 3.4, the tank agent is able to react either to an initial event or due to the heat radiation received from an external fire. In the first case, an accidental leak with consequent release of hazardous material is supposed to occur only due to internal causes, such as corrosion, erosion, and accidental pressurization.(46) If the tank is damaged, a secondary scenario may occur, which, in turn, may affect surrounding units in the AOI.

The **four states** in Figure 3.4 represent the tank possible states shown in Table 3.1. At a given time, the tank can only be in one of the states listed in Table 3.1. This implies that the states are mutually exclusive. Figure 3.4 also shows that there are two main modules (namely ProbM and ETA), which are used as judging points, so the tank never remains in these modules.

The **ProbM module** represents the vulnerability model, based on which the tank agent model can evaluate the probability of being damaged. The vulnerability models for heat radiation exposure based on Probit models developed by Landucci et al.(33) are adopted. Table 3.3 summarizes the vulnerability models, based on a simplified correlation for the estimation of time to failure, e.g., the time lapse before the eventual failure of a target tank since the start of an external fire. More details on the failure model of tanks are discussed in APPENDIX A.

Table 3. 3. Summary of vulnerability models (heat radiation effects) adopted for the assessment of vessel damage probability due to fire

Type of Tank	ttf correlations	Probit model
Atmospheric	$\ln(ttf) = -1.13 \cdot \ln(Q) - 2.67 \times 10^{-5} \cdot V + 9.9$ (3.1)	$Y = 9.25 - 1.85 \cdot \ln(\frac{ttf}{60.0})$ (3.3)
Pressurized	$\ln(ttf) = -0.95 \cdot \ln(Q) + 8.85 \cdot V^{0.032}$ (3.2)	

In Equations (3.1) to (3.3), ttf represents the time to failure (in seconds); Q denotes the total heat radiation received by the tank (in kW/m^2); V denotes the volume of the tank (in m^3); Y denotes the Probit value. In previous research,(47) a "threshold criteria" has been proposed, where if the received heat radiation was less than a threshold heat radiation, it could not make a credible damage and thus would be ignored. However, in this study, we do not perform a threshold criteria check, assuming that even if a target tank receives less heat radiation intensities, it still might be involved in the domino effect, especially due to the influence of synergistic effects in large scale cases.

The **ETA module** represents the post-release event tree (see APPENDIX B) to determine the occurrence probability of accidental scenarios (explosion, fire, dispersion, etc.).[1, 46] For illustrative purposes, we only considered pool fire as the possible accident scenario, assuming a null probability of delayed ignition (see APPENDIX B). Hence, explosions or flash fires are excluded from the analysis.

When the tank is in the "Normal" state (Figure 3.4):

1) it can react to an initial event (transition 1), such as flammable liquid leakage;
2) it can react to the heat radiation from other tanks (transition 4) and update the total heat radiation Q_{tot} it receives. After updating the Q_{tot}, the agent will compute the time to failure (ttf);
3) it can compute the probability of being damagedP_f, based on the vulnerability model, when the time reaches the ttf(transition 5).

When the tank is in the "HEAT-UP" state(Figure 3.4):

1) it can react to the heat radiation from other tanks, and update the total heat radiation Q_{tot}it receives (transition 10)
2) it can compute the ttf, if the updated Q_{tot} is greater than the threshold Q_{th1} (transition 11). Note that the threshold check here is different from the above-mentioned threshold criteria;[47] in the present study, when this threshold check is being executed, it means that the tank is already heated up. As shown in Table 3.2, Q_{th1} (and also Q_{th2}) is set to be a small value.

When the tank is in the "Leaking" state (Figure 3.4):

1) it can react to the heat radiation from other tanks if Q is greater than the threshold Q_{th2} (transition 12).

At the **ETA1 module**, if there is an ignition, a secondary fire will occur and the tank agent broadcasts heat radiation to all other tanks (transition 2). If there is no ignition, the tank will keep leaking out its content (transition 3). Only the primary tank agent will move to ETA1, thus we can set the ignition probability based on standard literature data, as $p_{i1} = 0.1$.[46]

At the **ProbM module**, the tank will compute the probability of being damaged (P_f) due to fire exposure. In case the tank does not fail, at probability $(1 - P_f)$, it will transfer to "HEAT-UP" state (transition 6, in Figure 3.4), which means the tank is not physically damaged, but its temperature and pressure are increasing due to the fire. In other words, a deterioration of the tank occurs without compromising its integrity; thus, with no release of hazardous materials. On the other hand, if the tank fails (with a probability P_f), it will transfer to ETA2 (transition 7, in Figure 3.4). This means that the tank is physically damaged and an event tree model is applied in order to trace the evolution of post-release scenario. It is worth mentioning that the domino targets move to ETA2 only as a consequence of fire exposure. Thus, due to the presence of heat radiation, the ignition probability for ETA2 was set higher than ETA1, for illustrative purpose, $p_{i2} = 0.6$.

At the **ETA2 module**, in case of ignition, a secondary fire scenario occurs ("FIRE" state) and the agent will broadcast heat radiation to all other tanks (transition 9, in Figure 3.4). If there is no ignition, the tank starts to/continues the release of hazardous material (transition 8).

3.3.3 Environment model

In the present work, the environment model has two functions. One function of the model, which considers the analysis of domino effect evolution, contains the information of the AOI, such as the geographic properties, the weather information etc. Another function is, on the side of the simulation run, that the environment model acts as an observer, monitoring and storing information about the states of each tank agent.

Figure 3.5 shows the static model of the environment model, modelled in GME, while the dynamic interaction of the environment model is presented in Figure 3.6.

As shown in Figure 3.5, the static model of the environment model consists of the global information of all the tanks, as well as some geographic information, weather information. The meanings of all the attributes can easily be understood according to their names, except the "HeatRadiationMatrix". HeatRadiationMatrix is a two-dimensional matrix, whose entry HRM_{ij} represents the heat radiation from tank i to tank j if tank i would be on fire.

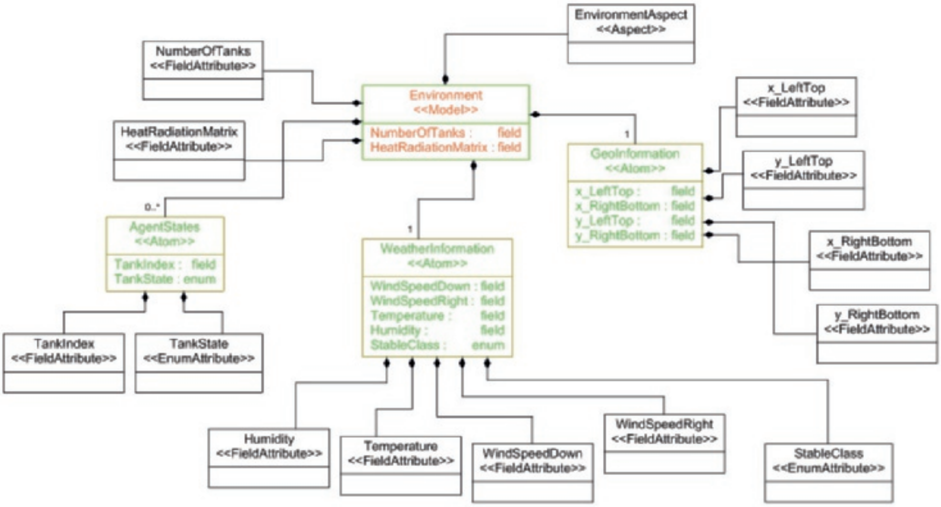


Figure 3. 5. Static model of the Environment model

Since the environment model manages all the global information, the tank agents need to Get the required Information (the GrI line in Figure 3.6) from it. For instance, when computing the heat radiation to other tanks, the tank on fire needs to know how many tanks are there in the AOI, and how many of them have already failed. When the tank agent changes state, it needs to Report the State change (the ReS line in Figure 3.6) to the environment model.

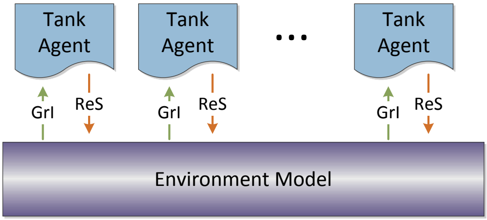


Figure 3. 6. Schematization of the dynamic interactions among agents and environmental model

3.4 Description of case studies

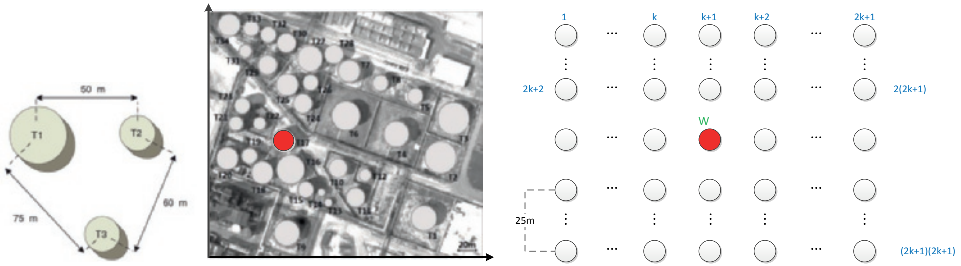
In this section, three case studies are used to demonstrate the application of DAMS model. Case study #1 is a simplified demonstration case study, aiming at interpreting the correctness of the DAMS model; case study #2 is a real scale case study composed of 34 chemical tanks, used to show the advantages of the DAMS model, compared to previous models;(23) case study #3 is an intentional constructed case study, aiming at illustrating the extensibility of the model to large scale cases possibly containing hundreds of tanks in industrial practice.

3.4.1 Case study #1: verification of the model

In order to test the validity of the model, a demonstration case study is firstly considered. The simplified example was defined in order to obtain the analytical solution of the problem , thus providing external validation data for the current model.

3.4.1.1 Description of the tank farm

The tank farm considered for the analysis of the simplified case study is represented in Figure 3.7(a) (modified version of the case used in Khakzad et al.(20)) and consists of three atmospheric storage tanks storing flammable liquids; the features of the tanks are summarized in Table 3.4. The same type of failure due to internal process causes was assumed to affect every tank, causing the release of the entire liquid content in the catch basin in ten minutes.(48)



(a) Case study #1

(b) Case study #2

(c) Case study #3

Figure 3. 7. Layout of the case studies considered in Section 3.4

Table 3. 4. Features of the tanks considered for the analysis of case study#1 and consequence assessment of the primary scenarios

ID	Coord. (x;y in m)	Subst.	D (m)	H (m)	V (m^3)	Initial event frequency (y^{-1})	Radiation on each target (kW/m^2)		
							T1	T2	T3
T1	0.0; 59.8	Hexane	20	10	3142	1×10^{-4}	-	23.8	15.7
T2	50.0; 59.8	Benzene	14	8	1232	1×10^{-4}	25.5	-	26.4
T3	45.3; 0.0	Benzene	14	8	1232	1×10^{-4}	9.54	12.5	-

The ALOHA software for consequence analysis(49) allowed estimating the heat radiation caused by the pool fire following the ignition of the flammable material. The following meteorological conditions were considered for the analysis of the case study: stability class D, wind at 5 m/s blowing from North, ambient temperature of 25°C and 50% relative humidity.

The results of the consequence assessment are also reported in Table 3.4. It is worth noting that the outputs of the ALOHA software (i.e., the heat radiation matrix) are used as the inputs of the DAMS model. By setting critical conditions for the ALOHA software, the output heat radiation would also be high, thus the domino risk assessed by the DAMS model are the worst result as well.

For illustrative purpose, T1 is assumed to be tagged with an initial event.

3.4.1.2 Analytic results of case study #1

Figure 3.8 shows the analytic results of the demonstration case, based on the Dynamic Event Tree Analysis,(50) considering not only the probabilistic dimension, but also the time dimension.

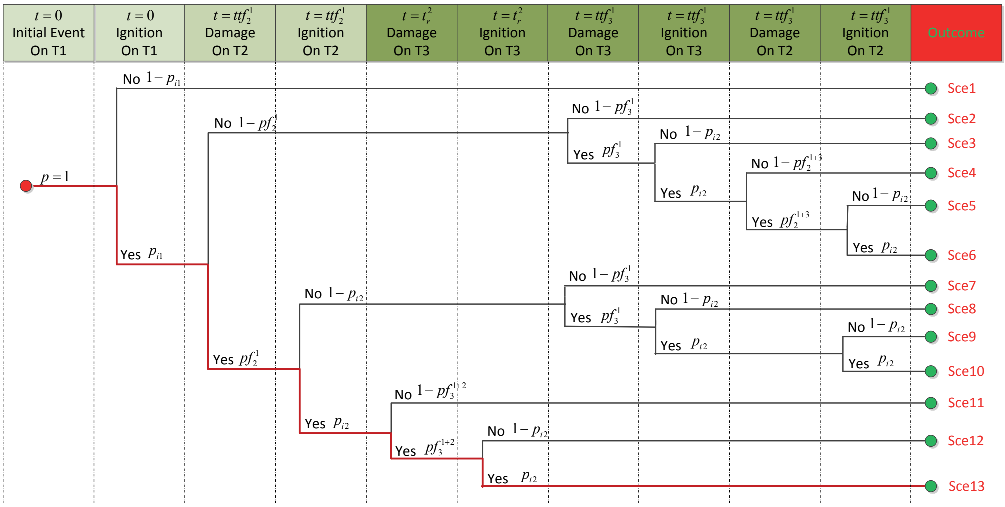


Figure 3. 8. Dynamic Event Tree Analysis: analytic result of the simplified case study

At $t=0$, there is a probability of p_{i1} that T1 would be ignited. If T1 is ignited, then T2 and T3 would be heated. By employing Equation (3.1) and data shown in Table 3.4, we have $ttf_2^1=536, and\ ttf_3^1=858$. At time $t=ttf_2^1$, T2 would be heated up, thus by employing the vulnerability model (i.e. Equation (3.3)), T2 would be physically damaged at probability $pf_2^1=0.5779$; furthermore, T2 would be ignited at that time by probability p_{i2}. If T2 is also on fire, T3 would receive heat radiation from both T1 and T2. By employing Equation (A.6) in APPENDIX A, we have $t_r^2=631$. At time $t=t_r^2$, T3 would be physically damaged and ignited, at probabilities pf_3^{1+2} and p_{i2}, respectively. Note that while computing pf_3^{1+2} by using Equation (3.1) and (3.3), the total heat radiation received (i.e. 15.7 kW/m^2 and 26.4 kW/m^2 from T1 and T2 respectively) by T3 should be used as input, obtaining $pf_3^{1+2}=0.9175$, as further explained in APPENDIX A.

The above paragraph explains the scenario 13, other scenarios can be explained in an analogous way. Table 3.5 summarizes all the 13 scenarios shown in Figure 3.8. Events in the description column are shown as (tank index, state, time), and they are listed according to the time of occurrence.

Table 3. 5. **Analytical solution of case study #1**

Scenario	Description (L: Leaking; H: Heat-up; F: Fire)	Probability
Sce 1	(1,L,0)	$1 - p_{i1}$
Sce 2	(1,F,0)→(2,H,ttf_2^1)→(3,H,ttf_3^1)	$p_{i1} \cdot (1 - pf_2^1) \cdot (1 - pf_3^1)$
Sce 3	(1,F,0)→(2,H,ttf_2^1)→(3,L,ttf_3^1)	$p_{i1} \cdot (1 - pf_2^1) \cdot pf_3^1 \cdot (1 - p_{i2})$
Sce 4	(1,F,0)→(2,H,ttf_2^1)→(3,F,ttf_3^1)→ (2,H,ttf_3^1)	$p_{i1} \cdot (1 - pf_2^1) \cdot pf_3^1 \cdot p_{i2} \cdot (1 - pf_2^{1+3})$
Sce 5	(1,F,0)→(2,H,ttf_2^1)→(3,F,ttf_3^1)→ (2,L,ttf_3^1)	$p_{i1} \cdot (1 - pf_2^1) \cdot pf_3^1 \cdot p_{i2} \cdot pf_2^{1+3} \cdot (1 - p_{i2})$
Sce 6	(1,F,0)→(2,H,ttf_2^1)→(3,F,ttf_3^1)→ (2,F,ttf_3^1)	$p_{i1} \cdot (1 - pf_2^1) \cdot pf_3^1 \cdot p_{i2} \cdot pf_2^{1+3} \cdot p_{i2}$
Sce 7	(1,F,0)→(2,L,ttf_2^1)→(3,H,ttf_3^1)	$p_{i1} \cdot pf_2^1 \cdot (1 - p_{i2}) \cdot (1 - pf_3^1)$
Sce 8	(1,F,0)→(2,L,ttf_2^1)→(3,L,ttf_3^1)	$p_{i1} \cdot pf_2^1 \cdot (1 - p_{i2}) \cdot pf_3^1 \cdot (1 - p_{i2})$
Sce 9	(1,F,0)→(2,L,ttf_2^1)→(3,F,ttf_3^1)→ (2,L,ttf_3^1)	$p_{i1} \cdot pf_2^1 \cdot (1 - p_{i2}) \cdot pf_3^1 \cdot p_{i2} \cdot (1 - p_{i2})$
Sce 10	(1,F,0)→(2,L,ttf_2^1)→(3,F,ttf_3^1)→ (2,F,ttf_3^1)	$p_{i1} \cdot pf_2^1 \cdot (1 - p_{i2}) \cdot pf_3^1 \cdot p_{i2} \cdot p_{i2}$
Sce 11	(1,F,0)→(2,F,ttf_2^1)→(3,H,t_r^2)	$p_{i1} \cdot pf_2^1 \cdot p_{i2} \cdot (1 - pf_3^{1+2})$
Sce 12	(1,F,0)→(2,F,ttf_2^1)→(3,L,t_r^2)	$p_{i1} \cdot pf_2^1 \cdot p_{i2} \cdot pf_3^{1+2} \cdot (1 - p_{i2})$
Sce 13	(1,F,0)→(2,F,ttf_2^1)→(3,F,t_r^2)	$p_{i1} \cdot pf_2^1 \cdot p_{i2} \cdot pf_3^{1+2} \cdot p_{i2}$

3.4.2 Case study #2: application of the model

In this section, we apply the DAMS model to a realistic chemical area containing 34 tanks, as shown in Figure 3.7(b). Further information of this case study is given in APPENDIX D.

Analytical methods such as dynamic event trees and Bayesian networks could be quite complex if they are implemented on this case study. However, the DAMS model proposed in this study can easily be implemented on this case study. Section 3.5.2 gives some computational results of this case study.

3.4.3 Case Study #3: computational complexity of the model

Although one advantage of DAMS model is that the number of replications will not be influenced by the number of tanks (see proofs in APPENDIX C), the computational time of each replication will increase when the number of tanks increases. In this section, we will show the computational time of cases with different numbers of tanks in case study # 3.

When the number of tanks increases, it becomes difficult to collect and input all the required data to the model. A typical tank farm is represented in Figure 3.7(c) in order to generate typical inputs for the computational time testing. As shown in Figure 3.7(c), the tank farm consists of $\mathrm{n} = (2\mathrm{k} + 1)^2$ tanks, and each tank is located on one grid in a $(2\mathrm{k} + 1) \times (2\mathrm{k} + 1)$ square.

The tank agents' index and position are given in the figure. We assume that all the tanks are atmospheric vertical cylindrical tanks $(\mathrm{d} = 14\mathrm{m}, \mathrm{h} = 8\mathrm{m})$ storing benzene. The same environment information used in the simplified case were adopted in the present assessment. The consequence assessment of the pool fire resulting from the failure of each tank were assessed using ALOHA.

We set the middle tank (the red one) as the primary unit W and run $N = 10^6$ replications for increasing the values of k.

3.5 Results

3.5.1 Analysis of case study #1

Based on the information given in Section 3.4.1, the tank agents and the environment model were initialized.

When tagged with an initial event, tank T1 has $(1 - p_{i1})$ probability of being in "LEAKING" state. In this case, the simulation would stop since the tank no longer contributes to the domino effect. Thus, such a case is excluded from the simulation. In other words, in the simulation, tank T1 is set on a "FIRE" state, and each simulation result is multiplied with the respective probability of immediate ignition (p_{i1}).

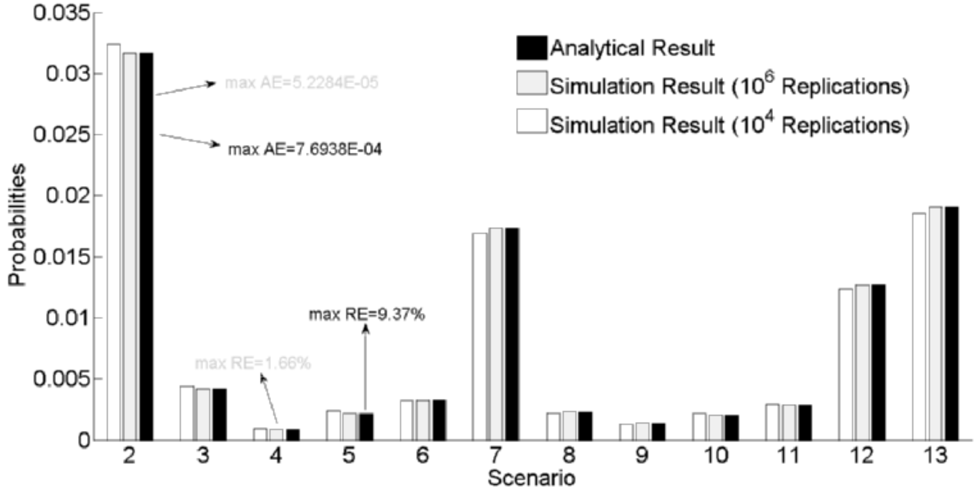


Figure 3. 9. Results of case study #1. Parity plot comparinganalytic and simulation results

Figure 3.9 shows the comparison of the results obtained via the application of the DAMS model against the analytical results described in Table 3.5. As shown in the figure, a good agreement is obtained, with a maximum relative error of 1.66% and a maximum absolute error of 5.2284×10^{-5} in case of running 10^6 replications. Therefore, the present approach is considered reliable and can be extended to the analysis of more complex cases.

3.5.2 Analysis of case study #2

Based on the information given in Section 3.4.2, we can initialize the tank agents and the environment model. T17 (indicated in red colour, in Figure 3.7(b)) is set to be the primary unit that is on "FIRE" (as explained above). To make sure that the probabilistic results are reliable on the thousandth, 10^6 replications were used, see APPENDIX C for further explanation.

Figure 3.10 shows the mean time μ $(\pm\sigma)$ of catching fire of each of the tanks. For example, T1 is on fire in500,668 replications among the 10^6 replications, and it might be on fire at different times in each of the replications.

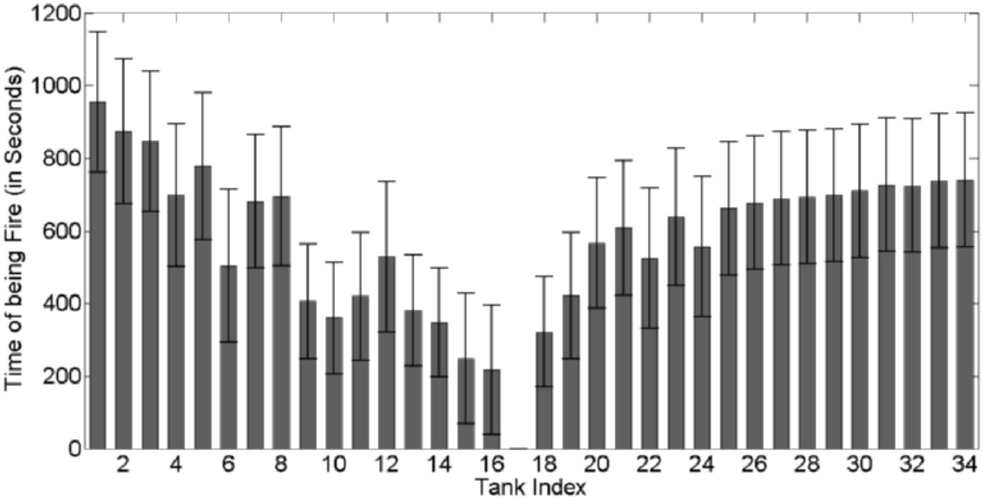
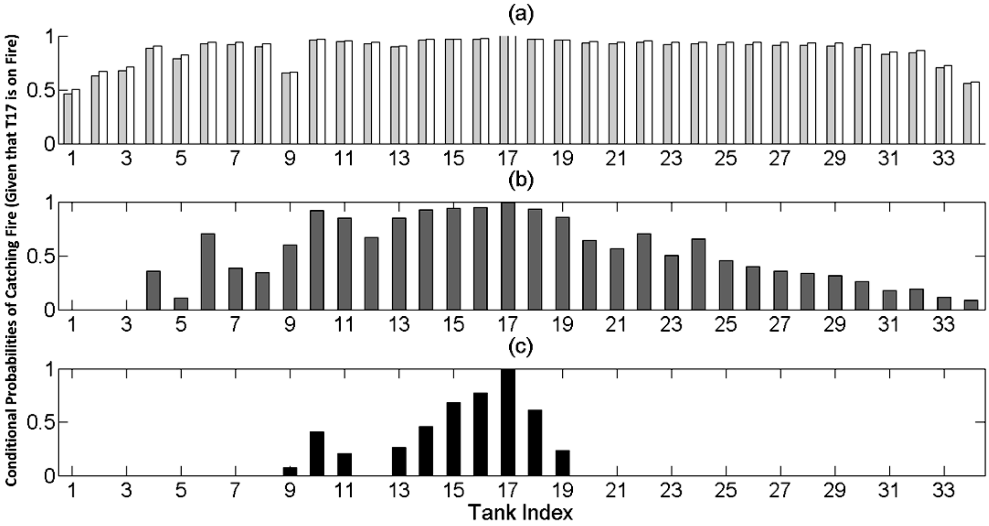


Figure 3. 10. Time distribution of catching fire of each tank



* (a):in case of no response at all (the white bar) and an emergency response time of 20 minutes (the grey bar); (b):in case of an emergency response time of 10 minutes; (c) in case of an emergency response time of 5 minutes

Figure 3. 11. Conditional Probabilities of catching fire of each tank, w.r.t. different emergency response times

As shown in Figure 3.10, since T17 is set as the primary unit, and is assumedon "Fire"initially, its time of being on fire is 0 sec. Generally, the nearer a tank to tank T17, the quicker it would be affected by a domino effect, resulting the roughly "V" shape of the time bars. However, due to the different materials stored in the tanks and different environmental conditions (e.g. the wind direction), the time of being affected is not strictly propotional to the distance from T17.

Figure 3.11 shows the probabilities of being on "Fire" of each tank, with respect to different response times. The white bars (in (a)) show the probabilities of each tank of being fire, under the condition that there is no emergency response at all (and also without heat radiation threshold check); the light grey bars (Figure 3.11(a)) show the probabilities of each tank being on fire up until 20 minutes (this knowledge is important as the emergency response teams are usually able to intervine and mitigate the fires within 20 minutes); the medium grey bars (Figure 3.11 (b)) show the results up until 10 minutes, while the dark grey bars (Figure 3.11 (c)) show the results up until 5 minutes.

Figure 3.11 shows that, when there is no emergency response (the white bars), the tanks from index 10 to 30 have similar conditional probabilities of being on fire ranging from 0.9 to 1.0. That is, if we only consider the probabilistic dimension of domino effect, these 20 tanks have similar risks. However, by considering the emergency response time (i.e., the grey bars in Figures 3.11(b) and (c)), it can be noted that tanks 9 to 11 and 13 to 19 have higher domino risks than the others.

It can be concluded that, if a leakage happens at T17, all the other tanks in this area will have a risk of being on fire at a probability ranging from around 0.05 to 0.1 (recall that with an initial event, T17 will be on "Fire" with $p_{i1} = 0.1$). However, if the emergency response team starts the intervention within 5 minutes after the first fire, the risk of the whole plant will decrease significantly.

3.5.3 Analysis of case study #3

The results of the computational times are summarized in Figure 3.12. The computational times in case of $n \geq 169$ are estimated based on 10^4 replications, and are drawn as mean time $\pm \sigma$.

As shown in Figure 3.12, the computational time increases exponentially with the number of tanks. In order to implement the model to a realistic case as given in Reniers et al.[(12)] (i.e., $\text{n} = 225$), the estimated time would be around 52.32 hours.

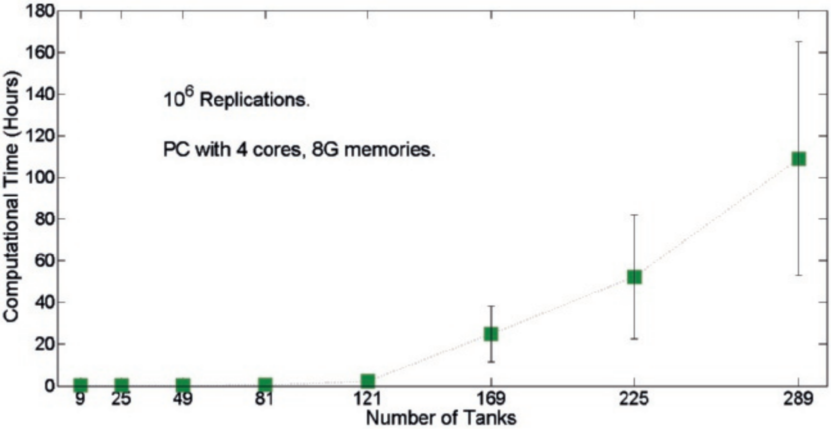


Figure 3. 12. Results of computational time analysis

Nevertheless, the DAMS model can still be applied to large scale cases because:

1) The computational time shown in Figure 3.12 is based on a personal computer with a limited computational and storage capacity. Since the different replications are independent, if

we employ distributed/parallel computing techniques, the computational time will reduce linearly to the computational capacity. For example, in case of $n = 225$, if we employ a work station with 50 cores (not very high requirement), then the computational time will be around 4.18 hours.

2) According to the Law of Moore, the computational capacity of computers will double every 24 months.[51] The last 50 years show the correctness of Moore's Law, however, the number of tanks in the chemical and process industries will not increase so quickly.

3) The proposed model can be used to explore possible scenarios, without knowing the probabilities, which otherwise would be very difficult if done by a human. In this case, it is not necessary to run the model so many times (see APPENDIX C). For example, if we just run the model 5000 times (thus the computational time will be $1/200$ of the time given in Figure 3.12), then a 99.3% reliability can be reached that any notable scenario (i.e., those with a probability higher than 0.001) will be recorded.

3.6 Discussion

The three case studies implemented in Sections 3.4 and 3.5 demonstrated that DAMS model is reliable in terms of the correctness of probabilistic, the advantages of capturing the time dimension, and the extensibility to large scale cases. Figure 3.8 and 3.9 demonstrates that DAMS correctly gets the probabilistic assessment of domino effects, taking into consideration of higher level domino effects (see, e.g., Sce 6, 10, and 13) and synergistic effects (see, e.g., Sce 10). Figure 3.11 illustrates how the time dimension of domino effect could change the risks, which however, is one of the main achievements of the developed model. Figure 3.12 shows the capability of the model of calculating domino risks of plants containing hundreds of tanks, whereas the current existing models are only able to deal with dozens of tanks.

Ccompared to the case studies, the realistic situation in industrial practice is more complicated, mainly due to three reasons: (i) the variety of domino agents (e.g. tanks, pipeline, mobile vehicles, etc.); (ii) the variety of accident scenarios (e.g., jet fire, VCE, etc.); (iii) the number of domino agents.

To deal with the variety of domino effects, we proposed the tank agent model, which with different static attributes can describe most kinds of tanks in a typical chemical and process industry (for instance, different shape attributes can describe different geometries, and then in the dynamic model, different types of tanks will use different parameters in the Vulnerability Model). For pipelines and mobile vehicles as well as some other kinds of domino agents, we need to develop agent-specific models. However, these models can be built in a similar way to the tank agent model.

To deal with the variety of accident scenarios, we only considered heat radiation escalation to represent pool fires jet fires, etc. In order to take other escalation vectors such as overpressure escalation and fragments into consideration, more chemical-related domain knowledge is needed, and thus stronger cooperation between simulation experts and chemical engineering experts.

To address the number of domino agents, we propose the "tank agent model", which is the basic unit in a domino effect; thus, regardless of the number of tanks, each physical tank is represented as a tank agent. In fact, the model proposed in this study has two important properties: (i) the tank agent's static and dynamic model will not be influenced by the number of tanks; (ii) the necessary replications of computational experiments will not be influenced by the number of tanks (see APPENDIX C).

To address the first and second reason, new models are needed thought they can be developed analogously. The model proposed in this study can be used in case of an increasing number of domino agents (i.e., the third reason).

3.7 Conclusions

In the present work, an agent-based modelling and simulation approach is proposed to estimate the potential domino risks in chemical plants. Tanks are modelled as agents who receive heat radiation from the tanks already on fire, run state transformation based on the received heat radiation, and if get on fire broadcast heat radiation to other tanks. The environment is modelled as an observer to manage global information. The proposed approach is able to capture not only the probabilistic dimension of domino effect, but also the time dimension which describes the timing when the domino effect may happen. Higher-level domino effect, as well as synergistic effects are also considered. The correctness, the advantages, and the extensibility of the model are illustrated by the computational experiments carried out on several case studies.

This work is a first attempt to employ an agent-based modelling and simulation (ABMS) approach to do domino risk assessment in chemical plants. By successfully modelling the dynamic procedure of domino effects, the outcome of this research can be used to support optimal and dynamic allocation of emergency resources. Furthermore, by tagging initial events on multiple targets, the model can support domino effect assessment triggered by a number of simultaneously failed tanks, a situation which may occur in case of a terrorist attack.

References

[1] Lees F. Lees' Loss prevention in the process industries: Hazard identification, assessment and control: Butterworth-Heinemann; 2012.

[2] Gledhill J, Lines I. Development of Methods to Access the Significance of Domino Effects from Major Hazard Sites: HSE Books; 1998.

[3] Bagster D, Pitblado R. Estimation of domino incident frequencies- an approach. Process Saf Environ Prot. 1991;69(4):195-9.

[4] Casal J, Darbra R-M. Analysis of past accidents and relevant case-histories. Domino Eff Process Ind Model Prev Manag. 2013:12-29.

[5] Cozzani V, Reniers G. Historical Background and State of the Art on Domino Effect Assessment. Domino Eff in the Process Industries: Model, Prev and Managing: Elsevier B.V.; 2013. p. 1-10.

[6] Necci A, Cozzani V, Spadoni G, Khan F. Assessment of domino effect: State of the art and research Needs. Reliability Engineering & System Safety. 2015;143:3-18.

[7] Antonioni G, Spadoni G, Cozzani V. Application of domino effect quantitative risk assessment to an extended industrial area. J Loss Prev Process Ind. 2009;22(5):614-24.

[8] Cozzani V, Gubinelli G, Antonioni G, Spadoni G, Zanelli S. The assessment of risk caused by domino effect in quantitative area risk analysis. J Hazard Mater. 2005;127(1-3):14-30.

[9] Rad A, Abdolhamidzadeh B, Abbasi T, Rashtchian D. FREEDOM II: An improved methodology to assess domino effect frequency using simulation techniques. Process Saf Environ Prot. 2014;92(6):714-22.

[10] Khakzad N, Reniers G. Risk-based design of process plants with regard to domino effects and land use planning. J Hazard Mater. 2015;299:289-97.

[11] Kourniotis S, Kiranoudis C, Markatos N. Statistical analysis of domino chemical accidents. J Hazard Mater. 2000;71(1):239-52.

[12] 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.

[13] Khan FI, Abbasi SA. The world's worst industrial accident of the 1990s: What happened and what might have been - A quantitative study. Process Saf Prog. 1999;18(3):135-45.

[14] Delvosalle C. A methodology for the identification and evaluation of domino effects. Belgian Ministery of Employment and Labour, Administration of Labour, Safety Chemical risks directorate. 1998.

[15] Khan FI, Abbasi S. Models for domino effect analysis in chemical process industries. Process Saf Prog. 1998;17(2):107-23.

[16] Khan FI, Abbasi SA. DOMIFFECT (DOMIno eFFECT): User-friendly software for domino effect analysis. Environmental Modelling and Software. 1998;13(2):163-77.

[17] Reniers GLL, Dullaert W. DomPrevPlanning©: User-friendly software for planning domino effects prevention. Safety Science. 2007;45(10):1060-81.

[18] Abdolhamidzadeh B, Abbasi T, Rashtchian D, Abbasi SA. A new method for assessing domino effect in chemical process industry. J Hazard Mater. 2010;182(1):416-26.

[19] Khakzad N, Reniers G, Abbassi R, Khan F. Vulnerability analysis of process plants subject to domino effects. Reliability Engineering & System Safety. 2016.

[20] Khakzad N, Khan F, Amyotte P, Cozzani V. Domino Effect Analysis Using Bayesian Networks. Risk Anal. 2013;33(2):292-306.

[21] Khakzad N, Reniers G. Using graph theory to analyze the vulnerability of process plants in the context of cascading effects. Reliab Eng Syst Saf. 2015;143:63-73.

[22] Landucci G, Argenti F, Tugnoli A, Cozzani V. Quantitative assessment of safety barrier performance in the prevention of domino scenarios triggered by fire. Reliab Eng Syst Saf. 2015;143:30-43.

[23] Khakzad N. Application of dynamic Bayesian network to risk analysis of domino effects in chemical infrastructures. Reliab Eng Syst Saf. 2015;138:236-72.

[24] Macal CM, North MJ. Tutorial on agent-based modelling and simulation. Journal of simulation. 2010;4(3):151-62.

[25] Holland JH. Emergence: From chaos to order: OUP Oxford; 2000.

[26] Epstein JM, Axtell R. Growing artificial societies: social science from the bottom up: Brookings Institution Press; 1996.

[27] Del Valle SY, Stroud PD, Smith JP, Mniszewski SM, Riese JM, Sydoriak SJ, et al. EpiSimS: epidemic simulation system. Los Alamos, NM: Los Alamos National Laboratory. 2006.

[28] Farmer JD, Foley D. The economy needs agent-based modelling. Nature. 2009;460(7256):685-6.

[29] Monostori L, Váncza J, Kumara SR. Agent-based systems for manufacturing. CIRP Annals-Manufacturing Technology. 2006;55(2):697-720.

[30] Chen X, Meaker JW, Zhan FB. Agent-based modeling and analysis of hurricane evacuation procedures for the Florida Keys. Natural Hazards. 2006;38(3):321-38.

[31] Dawson RJ, Peppe R, Wang M. An agent-based model for risk-based flood incident management. Natural Hazards. 2011;59(1):167-89.

[32] Kroshl WM, Sarkani S, Mazzuchi TA. Efficient allocation of resources for defense of spatially distributed networks using agent - based simulation. Risk Anal. 2015;35(9):1690-705.

[33] Landucci G, Gubinelli G, Antonioni G, Cozzani V. The assessment of the damage probability of storage tanks in domino events triggered by fire. Accident Analysis and Prevention. 2009;41(6):1206-15.

[34] Landucci G, Cozzani V, Birk M. Heat Radiation Effects. Domino Eff in the Process Industries: Model, Prev and Managing2013. p. 70-115.

[35] Salzano E, Hoorelbeke P, Khan F, Amyotte P. Overpressure Effects. Domino Eff in the Process Industries: Model, Prev and Managing2013. p. 43-69.

[36] Cozzani V, Salzano E. The quantitative assessment of domino effects caused by overpressure: Part I. Probit models. J Hazard Mater. 2004;107(3):67-80.

[37] Tugnoli A, Gubinelli G, Landucci G, Cozzani V. Assessment of fragment projection hazard: Probability distributions for the initial direction of fragments. J Hazard Mater. 2014;279:418-27.

[38] Tugnoli A, Milazzo MF, Landucci G, Cozzani V, Maschio G. Assessment of the hazard due to fragment projection: A case study. J Loss Prev Process Ind. 2014;28:36-46.

[39] Landucci G, Molag M, Cozzani V. Modeling the performance of coated LPG tanks engulfed in fires. J Hazard Mater. 2009;172(1):447-56.

[40] D'Aulisa A, Tugnoli A, Cozzani V, Landucci G, Birk AM. CFD modeling of LPG vessels under fire exposure conditions. AIChE Journal. 2014;60(12):4292-305.

[41] Reniers G, Cozzani V. Domino Effects in the Process Industries: Modelling, Prevention and Managing: Elsevier B.V.; 2013. 1-372 p.

[42] AnyLogic. Agent Based Modelling [cited 2017 29, March]. Available from: http://www.anylogic.com/agent-based-modeling.

[43] Gomez-Mares M, Zarate L, Casal J. Jet fires and the domino effect. Fire Safety Journal. 2008;43(8):583-8.

[44] Ledeczi A, Maroti M, Bakay A, Karsai G, Garrett J, Thomason C, et al., editors. The generic modeling environment. Workshop on Intelligent Signal Processing, Budapest, Hungary; 2001.

[45] Jansen DN, Hermanns H, Katoen J-P, editors. A probabilistic extension of UML statecharts. International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems; 2002: Springer.

[46] De Haag PU, Ale B. Guidelines for quantitative risk assessment (purple book). Committee for the Prevention of Disasters, The Hague (NL). 1999.

[47] Cozzani V, Gubinelli G, Salzano E. Escalation thresholds in the assessment of domino accidental events. J Hazard Mater. 2006;129(1-3):1-21.

[48] Safety CfE. Reference Manual Bevi Risk Assessments. the Netherlands: National Institute of Public Health and the Environment (RIVM), 2009.

[49] ALOHA. US Environmental Protection Agency, National Oceanic and Atmospheric Administration, ALOHA, Version 5.4.4 2013.

[50] Acosta C, Siu NO. Dynamic event tree analysis method (DETAM) for accident sequence analysis. Cambridge, Mass.: Dept. of Nuclear Engineering, Massachusetts Institute of Technology, 1991, 1991.

[51] Moore GE. Cramming more components onto integrated circuits. Electronics, 38 (8), April 1965. VLSI Technologies and Architectures. 2010.

4

SINGLE PLANT PROTECTION: A GAME-THEORETICAL MODEL FOR IMPROVING CHEMICAL PLANT PROTECTION

In this chapter, we introduce a game theoretic model for protecting a chemical plant from intelligent attackers. The model is named Chemical Plant Protection Game, abbreviated as "CPP Game". The CPP Game is developed based on the general intrusion detection approach in chemical plants. To this end, the general intrusion detection approach is firstly introduced. We develop and explain the CPP Game by modelling its players, strategies, and payoffs. Afterwards in section 4.4, different equilibrium concepts are used to predict the outcome of the CPP Game. An analysis of the inputs and outputs of the game is provided in section 4.5, from an industrial practice point of view. In section 4.6, a case study is employed, for demonstrating the model. Finally, conclusions are drawn at the end of this chapter.

This chapter is based on the following published papers:

Zhang, L., & Reniers, G. (2016). A Game-Theoretical Model to Improve Process Plant Protection from Terrorist Attacks. *Risk analysis*, *36*(12), 2285-2297.

Zhang, L., & Reniers, G. (2018). Applying a Bayesian Stackelberg game for securing a chemical plant. *Journal of Loss Prevention in the Process Industries*, *51*, 72-83.

Zhang, L., Reniers, G., Chen, B., & Qiu, X. (2018). Integrating the API SRA methodology and game theory for improving chemical plant protection. *Journal of Loss Prevention in the Process Industries*, *51*, 8-16.

4.1 Introduction

Since the 9/11 attacks in New York City, security research, and especially the protection of critical infrastructure, has gained even more attention from both academia and industry. Different to safety research that focuses on natural and randomly (non-strategic) hazards, security research has to face intelligent and strategic adversaries. Traditional methods or concepts used in safety science such as probabilistic risk assessments, historical data analysis and etc., no longer can be readily and easily used. When dealing with security problems, the adversaries' strategies should be taken into consideration instead of the incidents' probabilities.

Game theory, which originated in economic sciences, is a good choice to handle problems that contain intelligent players. Game theory has very rigor mathematical foundations, and if adequately used in the chemical security, we can obtain more accurate and more defensible quantitative results, besides the qualitative assessments and results used nowadays in chemical plants. In recent years, a lot of attention in academia has been laid on the combination of game theory and critical infrastructure protection. Cox(1) states that game theory and conventional (probabilistic) risk analysis techniques should be complementary to each approach, for advancing security risk research. Conventional risk analysis techniques (e.g., API SRA standard 780) can provide quantitative inputs for game theory models, while game theory models could process these inputs in an intelligent way, making best use of these data. Tambe(2) and his group used game theory to improve the security situation in airport patrolling, air marshals' allocation, and coast line protection. They developed several decision support systems based on their research, and these systems now work in reality. Bier(3) and her group researched the combination of game theory and security assessment methods from a theoretical viewpoint. They answered the questions why game theory has an important role in security research, and illustrated the advantages and disadvantages of using game theory in operational security. Talarico(27) presented a multi-modal security-transportation model to allocate security resources within a chemical supply chain which is characterized by the use of different transport modes, each having their own security features.

Although there are already some researches on using game theory to improve operational security, in fact in the process security, no research has been done as yet, to the best of the authors' knowledge. Security problems in the process industries are different to those in aviation or the electric power grid for example, although they are all critical infrastructures. We cannot readily apply game theoretical models now being used in aviation, within the process industries directly. Different security models are implemented in different types of industries. For instance, in Tambe's model(2), air marshals are allocated to defend an air plane, and therefore the players' strategies are limited to "protect" (that is, to allocate an air marshal on the plane) or not (that is, no air marshal on the plane), and "attack" or not. However, in case of security in the process industries, the model is more complex, the strategies may be at a different alert level (discrete model) or at a different investment level (continuous model). Moreover, specific of the process industry is the nature of the consequences: explosion fire and toxic spread. Actually, an attacker may want trying to get as much escalation as possible. This is by knowing what effects hazardous material releases will have. For a more detailed discussion of this, please see Reniers(26).

4.2 General intrusion detection approach in chemical plants

Though security risk assessment is a relatively new topic stimulated by the 9/11 attack, chemical industries have a long history with respect to separating their assets and facilities from citizens and nearby communities. The main purpose of this isolation is the existence of large amounts of dangerous materials, which, in case of a major accident, might lead to losses suffered by the plant as well as by the surrounding communities. The separation is achieved by distance (using perimeters), evidently, but also by intrusion control.

Figure 4.1 shows a typical illustrative layout of a chemical plant, indicating its general physical intrusion detection approach. As shown in the figure, the terms “PERIMETER”, “ENTRANCE”, “ZONE”, and “TARGET” are used. A perimeter concerns the boundary of an area related to the plant, and may consist of a fence, a wall, or even a geographical boundary such as a river bank. Whatever it is composed of, a perimeter is a closed area, and it should prevent illegal intrusion. Entrances are attached to perimeters. Authorized people are checked at entrances and afterwards they are allowed to pass the entrance. Zones are generated automatically due to the perimeters. As shown in Figure 4.1, zones are distinguished by different levels and each level may have several sub-zones. For instance, zone level 2 in Figure 4.1 has 3 sub-zones. Moreover, a higher level zone (e.g., level i) is contained in a lower level zone (i.e., the level $i-1$). Targets (illustrated as triangles in Figure 4.1) are the assets that may attract the potential attackers. Due to the possibility of being detected in each entrance and in each zone, an intruder could evidently easier reach a target situated in lower level zones. To this end, important infrastructures (based on for instance sensitivity, confidentiality, dangerousness, and what have you) of the chemical plant are usually situated in higher level zones.

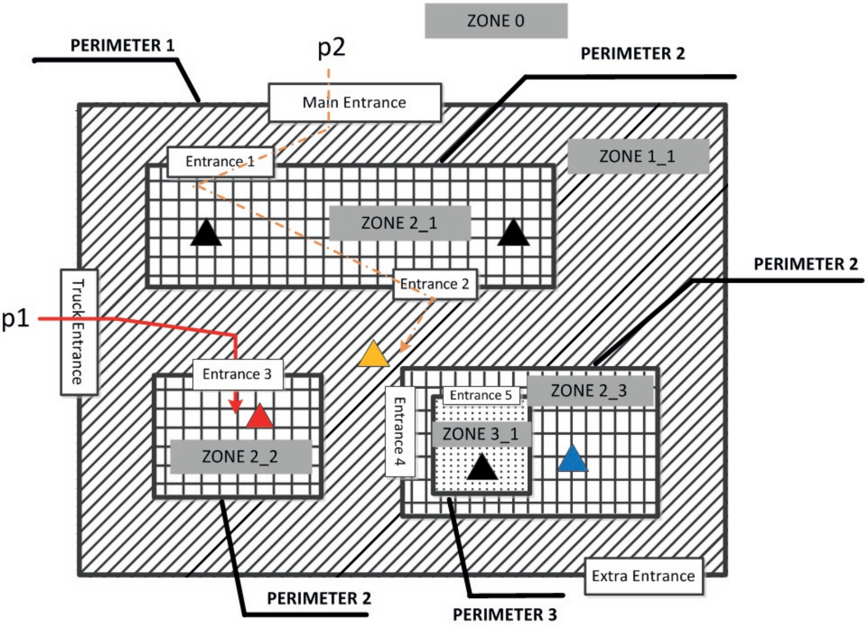


Figure 4. 1. General Physical Intrusion Detection Approach in Chemical Plants

With the general intrusion detection system, the industrial manager may allocate its security resources at each entrance or in each zone. For instance, the main entrance of a plant could be equipped with an employee card recognition machine, a security guard, a communication system (to the local police station or to the security centre of the plant), while a camera system (e.g. CCTV) and regular guard patrolling forms a typical detection scenario in zones. It is worth noting that security scenarios at each entrance or zone are not necessarily fixed, that is, the defender may have several different scenarios at one place, and she chooses a scenario according to the threat level. Different scenarios represent different security alert levels (SALs). If the plant evaluates that it has a higher threat level, either based on intelligence gathered or based on past security events, the plant will increase its SAL at some entrances or zones. Higher SALs need enhanced security scenarios. For instance, in case of a lower SAL, the security guards are not armed, while in a higher SAL, they could be armed (depending on the world region where the plant is located). In a higher SAL, the intruder will be more likely to be detected and halted, but the cost of a higher SAL is usually higher.

An intruder would have to, firstly, choose a target to attack, secondly, choose an attack scenario, and thirdly, choose a critical (easiest for him) path to reach the target. It is a complex decision problem, since there are multiple facilities and areas inside a plant, and the attacker would choose the target according to his purpose. A terrorist, for example, would prefer to cause some leakage or explosion at some storage tank(s), production facilities etc., while an environmental activist, might prefer to shut down the power station of a plant to stop the operation of the plant. Different types of attackers will use different attack scenarios, to different targets. The scenario of a terrorist attack with respect to a toxic tank can be related to the use of an explosive device, while the scenario with respect to a production facility can be linked to the switching off of an important safety valve. The attacker also needs to choose an intrusion path to reach the target. An intrusion path consists of several entrances where each entrance belongs to a different level of perimeter. For instance, the intrusion path p1 in Figure 4.1 consists of the truck entrance and entrance 3. Furthermore, in order to simplify the research and to reflect reality, we assume that the intruder would never step into the same zone level twice. This assumption excludes paths such as p2 in Figure 4.1, since if p2 is followed, the intruder would step into zone level 1 twice (after the main entrance and after entrance 2). This assumption is useful for simplifying the path analysis. Otherwise without this assumption, the intruder may have infinite numbers of intrusion paths. This assumption also reflects reality, since if the intruder steps into the same zone level twice, he would have to pass more entrances and zones, which will increase the likelihood of being detected.

Figure 4.2 is a plot from the intruder's viewpoint, illustrating the intrusion and attack procedure for the case of a chemical plant, as shown in Figure 4.1. The indexes of zones and perimeters are using the same name/label as used in Figure 4.1, while entrance A_{rj} denotes the j^{th} access of the perimeter r. Without loss of generality, we can map A_{11} in Figure 4.2 to the main entrance in Figure 4.1; A_{12} to the truck entrance; and so forth. Besides intruding through an entrance, the intruder may also step over the perimeter, which is also considered in Figure 4.2 as an access. The red dot line in Figure 4.2 represents the intrusion path p1 in Figure 4.1. As discussed earlier, the defender allocates security resources at entrances and in zones, while the intruder would have to pass the entrances and zones being situated on his intrusion path. We define a $P_i^z \in [0,1]$, denoting the probability of successfully passing zone level i, and we define a $P_r^p \in [0,1]$, denoting the probability of passing the perimeter r. Both P_i^z and P_r^p should be determined by the defender's security alert level at the entrance or the zone, and by the attacker's attack scenario. A more detailed discussion of P_i^z and P_r^p will be given in section 4.5.1. In theoretical research, the contest success function (CSF) (2) is often used to estimate these two parameters.

Since there might be multiple entrances on one perimeter (which is always the case in industrial practice), the probability of successfully passing the perimeter P_r^p will be equal to the probability of successfully passing the chosen entrance j, denoted as P_{rj}^p. For instance, in the intrusion path p1, we would have $P_1^p = P_{12}^p, P_2^p = P_{24}^p$ (see Figure 4.2). The security alert levels at different entrances of a perimeter can be different, thus the intruder has different success probabilities by choosing different entrances. However, it is not necessary that the intruder always chooses the easiest entrance to intrude. This is the result of the detection in the zones: an easier passing entrance might be situated further (in distance) to the target than a more difficult passing entrance, thus although the intruder could easily pass the entrance, the probability of him being detected in the zone will be higher.

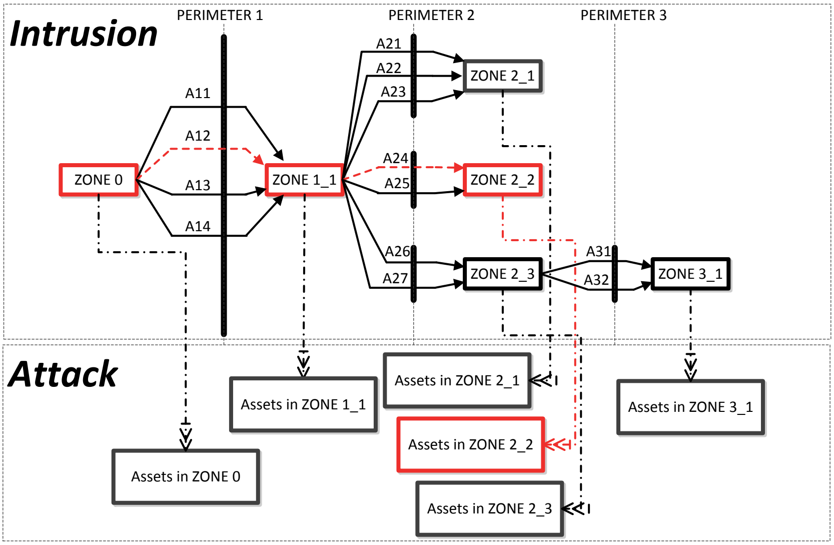


Figure 4. 2. The intrusion and attack procedure

Based on above analysis, if the intruder aims to attack a target situated in zone level 1, the probability that he will successfully reach the target can be calculated as shown in Formula (4.1).

$$P = \prod_{i=0}^{I} P_i^z \cdot \prod_{r=1}^{I} P_r^p. \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (4.1)$$

4.3 Game-theoretical modelling: the Chemical Plant Protection Game (CPP Game)

We introduce and explain the chemical plant protection game (CPP Game) from three modelling perspectives, namely, that of the players (section 4.3.1), the strategies (section 4.3.2), and the payoffs (section 4.3.3).

4.3.1 Players

The chemical plant protection game is played between 2 players: the defender and the attacker.

The ‘defender’ represents the security department of a chemical facility, who is responsible for the security management (amongst others carrying out security risk assessments) of the plant. The attacker can be various, for instance, terrorists, criminals, and environmental activists. The game is modelled as a two-player game implying that the collusion among different types of attackers are excluded from this research. In reality, different types of attackers may cooperate to implement an attack. A very straightforward cooperation can be, for instance, a disgruntled employee working together with criminals to cause damages and losses to the plant. If collaborative partnerships among attackers are taken into consideration, the model will be a multiple players game, and it will involve both a cooperative game (among different attackers) and a non-cooperative game (between the defender and the attackers). For legibility and simplicity reasons, in this research, we ignore the case of an alliance of attackers, and assume that different attackers are independent to each other.

The existence of different types of attackers can be modelled in a Bayesian approach. In a chemical plant protection game where multiple types of attackers are considered, a prior probability can be assigned to each type of attacker. Moreover, these prior probabilities can be calculated based on the threat level information, as shown in Formula (4.2). With the prior probability, the Bayesian chemical plant protection game can be interpreted as a defender-attacker two-player game in which the defender is facing threat t with probability p^t. By employing a Bayesian approach, not only the collusion among different types of attackers is not modelled, but also the simultaneous occurrence of more than 2 types of attacker is ignored. The probability p of a threat t can be expressed as:

$$p^t = \frac{ts^t}{\sum_{l \in TL} ts^l}, t \in TL, ts \in TS. \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (4.2)$$

In which p^t denotes the prior probability of threat t, ts^t represents the threat level of threat t, TL and TS are the threat list and the threat level set respectively. This threat information can be obtained by some conventional security risk assessment results, using the API SRA [(3)], for instance. Section 4.5.1 shows more details about obtaining the inputs for a chemical plant protection game.

The assumption of rationality can be justified for the defender, being the security department of a chemical plant. However, assuming rationality for the attacker is not so straightforward and requires more explanation and interpretation. As mentioned in Chapter 2, rationality in game theory is defined as the maximisation of a player's own payoff. Thus although from an ordinary person's point of view, some attacker behaviours (e.g., terrorist suicide attacks) are very emotion-based, intelligent attackers do have their own goal and they plan their attack to maximize, for instance, the defender's damage (in case of terrorism). Therefore, if the attacker's payoff is modelled according to the attack goal, then the rationality assumption can be defended. In Chapter 5, we extend the chemical plant protection game to deal with boundedly rational attackers.

Another interesting topic about modelling from the viewpoint of the player concerns whether the players have complete information of the game or not. Research has pointed out that terrorists need to collect certain information before they implement an attack. In fact, some research reveals that terrorists are able to obtain at least 80% (in some cases even 100%) of the needed information through public access [(4)]. According to this research, it is reasonable to assume that the attackers have complete information of the game. However, due to the lack of historic data, the defender hardly has any information about the attackers, and the required amount of information that the defender need to dispose of, might thus be few and very hard to obtain. For instance, the attacker can easily estimate the defender's defence costs for a security scenario (e.g., a combination of camera and patrolling), while it can be quite difficult for the defender to know the attacker's cost of obtaining an explosive device and/or to know the attacker's total available budget. To meet this modelling challenge, in Chapter 5, we propose a chemical plant protection game in which the defender only knows a distribution-free interval for the attacker's parameters. In the basic chemical plant protection game (in Chapter 4), we assume that both the defender and the attacker have complete information of the game.

In conclusion, the chemical plant protection game is a two-player game played by a defender and an attacker, and the types of attacker can be various. Both players have complete information of the game and both are assumed to behave rationally. The information related to the game is common knowledge to both players.

4.3.2 Strategies

To secure a chemical plant, the defender may set different security alert levels (SAL) at each entrance and in each zone of a chemical plant. Table 4.1 demonstrates an example of corresponding countermeasures of different security alert levels. A pure strategy of the defender can be defined as a combination of security alert levels at each entrance and in each (sub-) zone, as shown in Formula (4.3):

$$s_{di} = z^0 \times \prod_{r=1}^{Q}(A_1^r \times A_2^r \times \ldots \times A_{ent(r)}^r \times z_1^r \times z_2^r \times \ldots \times z_{sub(r)}^r) \cdots\cdots\cdots\cdots \tag{4.3}$$

In which z^0 denotes the SAL in zone level 0 (i.e., the outside zone); Q represents the total zone levels in the plant (for instance, the plant shown in Figure 4.1 has a $Q = 3$); A_j^r is the SAL at the j^{th} entrance of perimeter r; $ent(r)$ is the number of entrances of perimeter r (for instance, the plant in Figure 4.1 we have $ent(1) = 4$, $ent(2) = 7$, $ent(3) = 2$); z_i^r denotes the SAL in the i^{th} sub-zone of zone level r; $sub(r)$ denotes the number of sub-zones in zone level r (for instance, in Figure 4.1, we have $sub(1) = 1, sub(2) = 3, sub(3) = 1$); × and Π represent the Cartesian product.

For simplicity reason, we assume that the defender has the same number of SALs at each entrance point and in each subzone, say, k. For instance, a plant may set the security alert level at its main entrance as low/medium/high, thus we have $k = 3$. In other cases the security alert level can be white/blue/yellow/orange/red/, thus we have $k = 5$. The total number of the defender's pure strategies n can therefore be calculated by Formula (4.4):

$$n = k^{1+\sum_{r=1}^{Q}(ent(r)+sub(r))} \cdots\cdots\cdots\cdots \tag{4.4}$$

We further define the defender's pure strategy set as $S_d = \{\mathrm{s}_{\mathrm{d1}}, \mathrm{s}_{\mathrm{d2}}, \ldots, \mathrm{s}_{\mathrm{dn}}\}$.

Table 4. 1. Illustrative countermeasures corresponding to different security alert levels

SAL	Corresponding Measures	
	Entrances	(Sub-) Zones
1 (White)	➢ Access Control: wearing badge	➢ Light: minimum perimeter
2 (Yellow)	➢ Access Control: badge reader ➢ CCTV: security check	➢ Light: all the zone, for sight
3 (Orange)	➢ Access Control: badge reader, restricted list ➢ CCTV: security check, cyclical check at perimeter	➢ Light: all the zone, for video ➢ Dog detection: dog nearby
4 (Red)	➢ Access Control: badge reader, explosive search ➢ CCTV: security check, permanent check at perimeter	➢ Light: all the zone and towards outside, for video ➢ Dog detection: dog with microphone inside

An attacker's pure strategy is modelled as the combination of (i) which target to attack; (ii) with what attack scenario; and (iii) from which intrusion path to reach the target, as formulated in Formula (4.5):

$$s_{ai} = target \times \prod_{r=1}^{I} j_r \times e \cdots\cdots\cdots\cdots \tag{4.5}$$

In which $target$ denotes the facility that the attacker is going to attack; I represents the zone level that the target is situated in; e is the attack scenario, or the attack effort; j_r is the entrance that the attacker chooses to pass perimeter r, and $j_r \in \{1,2,\ldots,ent(r)\}$.

An example of the attacker's pure strategy is the following: if the attacker follows intrusion path p1 in Figure 4.1, and he wants to attack a target in zone 2_2 (assume the target has an index $\mathcal{L}$) with an explosive device, say a bomb, then this pure strategy can be expressed as: $s_{ai} = \mathcal{L} \times Truck\ Ent \times Ent3 \times Bomb$.

The total number of pure strategies of the attacker m can be calculated by Formula (4.6), and we define the attacker's pure strategy set as: $S_a = \{\mathrm{s_{a1}}, \mathrm{s_{a2}}, \ldots, \mathrm{s_{am}}\}$. In Formula (4.6), k_a^{tgt} is the number of attack scenarios that the attacker may use to the target tgt; Ast_0 denotes the set of targets in zone 0; $Ast_{r,i}$ denotes the set of targets that are situated in the i^{th} subzone in zone level r; $ent(j,i)$ represents the number of entrances on perimeter j that can lead to sub zone i. For instance, in Figure 4.1, we have $ent(2) = 7$ and $ent(2{,}2\_1) = 3$. It is worth noting that the defender can hardly enumerate all the available attack scenarios, thus the k_a can be difficult to know [(5)]. In the chemical plant protection game, we simply list all the possible attack scenarios according to the defender's knowledge and experiences, which aims to make the best use of the available data/knowledge.

$$m = \sum_{tgt \in Ast_0} k_a^{tgt} + \sum_{r=1}^{Q} \sum_{i=1}^{sub(r)} \left(\sum_{tgt \in Ast_{r,i}} k_a^{tgt} \cdot \prod_{j=1}^{r} ent(j,i) \right) \cdots\cdots\cdots\cdots\cdots\cdots (4.6)$$

According to Formulas (4.4) and (4.6), we notice that both the defender and the attacker will have a finite number of pure strategies (i.e., n and m respectively). The number of pure strategies of the defender will increase dramatically as the scale of the plant grows. However, this is not a problem in industrial practice, as the defender's pure strategy set can be cut according to her available budget, see, for instance, Talarico et al. [(6)].

The players' mixed strategy space can be defined as $Y = \{y \in R^{|S_d|} \mid \sum y_i = 1, y_i \in [0,1]\}$, and $X = \{x \in R^{|S_a|} \mid \sum x_i = 1, x_i \in [0,1]\}$, for the defender and for the attacker, respectively.

4.3.3 Payoffs

Payoffs are numbers representing the players' motivations. In the chemical plant protection game, the defender's payoff is defined as her expected loss from an attack minus her defence cost, while the attacker's payoff is defined as his expected gain from an attack minus his attack cost, as formulated in Formulas (4.7) and (4.8) respectively.

$$u_d(s_a, s_d) = -(P(s_a, s_d) \cdot P_y(s_a) \cdot L_y(s_a) + C_d(s_d)) \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (4.7)$$

$$u_a(s_a, s_d) = \tilde{P}(s_a, s_d) \cdot \tilde{P_y}(s_a) \cdot \tilde{L}_y(s_a) - C_a(s_a) \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (4.8)$$

In which: (s_a, s_d) denotes a given attacker and defender pure strategy pair; P is the probability that the attacker will successfully reach the target, and it is calculated by Formula (4.1); P_y is the probability that the attacker will successfully implement the attack in condition that he has reached the target; L_y is the estimated loss that if the attack is successfully implemented; $\tilde{P}$, $\tilde{P_y}$, and $\tilde{L}_y$ have the same meaning as P, P_y, and L_y, but they are the parameters estimated from the attacker's point of view; C_d and C_a are the defence and attack cost, respectively.

It is worth noting that for one and the same parameter, the defender and the attacker may have different perceptions and hence different estimations. For instance, for a thief, stealing a computer from a control room would be his true goal and lead to obtaining the hardware, but the defender may be most interested in the potential loss of important data (e.g., technique documents). Thus in this case, we may have $L_y \gg \tilde{L}_y$. Furthermore, for probabilities of successfully going through some entrances or zones, the defender and the attacker can also have quite different estimations: for a risk-seeking intruder, we may have $P \leq \tilde{P}$; for a risk-averse intruder, we may have $P \geq \tilde{P}$.

By implementing Formulas (4.7) and (4.8) for each defender and attacker pure strategy pair, we will obtain their payoff matrices, denoted as U_d and U_a respectively. According to the players' pure strategy numbers, we know that both payoff matrices are $m \times n$ matrices. If there are multiple types of attackers, we denote the defender and the attacker's payoff as U_d^t, U_a^t, for $\forall t \in TL$, respectively.

Formulas (4.7) and (4.8) also reveal that the chemical plant protection game is not necessarily a zero-sum game, though defenders and attackers always have opposite interests. The non-zero-sum property of the CPP game contains two aspects: (i) both the defence cost C_d and the attack cost C_a are involved in the payoff definitions, and no player will benefit from the other player's behaviour cost; (ii) the defender and the attacker might evaluate the same parameters with different values, including probabilities (e.g. , P and $\tilde{P}$) and consequences (i.e., L and $\tilde{L}$).

However, in some special conditions, the CPP game can be a strategically zero-sum game, as explained hereafter. Re-write the payoff Formulas (4.7) and (4.8) as Formulas (4.9) and (4.10).

$$u_d(s_a, s_d) = -f(s_a, s_d) - C_a(s_a) \tag{4.9}$$

$$u_a(s_a, s_d) = \tilde{f}(s_a, s_d) - C_d(s_d) \tag{4.10}$$

In which:

$$f(s_a, s_d) = P(s_a, s_d) \cdot P_y(s_a) \cdot L_y(s_a) - C_a(s_a) + C_d(s_d) \tag{4.11}$$

$$\tilde{f}(s_a, s_d) = \tilde{P}(s_a, s_d) \cdot \widetilde{P_y}(s_a) \cdot \tilde{L}_y(s_a) - C_a(s_a) + C_d(s_d) \tag{4.12}$$

Define a zero-sum defender-attacker game $(F, -F)$, of which the defender and the attacker's strategy sets are the same as the defender and the attacker's strategy sets in the CPP game, and payoff units of the new game are defined by Formulas (4.11) and (4.12).

If, in some cases, for all strategy tuples of the CPP game, the condition $\tilde{P} \cdot \tilde{P}_y \cdot \tilde{L} = P \cdot P_y \cdot L$ holds, then $(\bar{x}, \bar{y})$ is a NE of the CPP game if and only if $(\bar{x}, \bar{y})$ is a NE of the game $(F, -F)$.

Proof: Since $\tilde{P} \cdot \tilde{P}_y \cdot \tilde{L} = P \cdot P_y \cdot L$, we directly know that $\tilde{f} = f$. According to the definition of NE, $(\bar{x}, \bar{y})$ is a NE of the CPP game $\Leftrightarrow \begin{cases} \bar{x}^T \cdot U_a \cdot \bar{y} \geq x^T \cdot U_a \cdot \bar{y}, \ \forall x \in X \\ \bar{x}^T \cdot U_d \cdot \bar{y} \geq \bar{x}^T \cdot U_d \cdot y, \ \forall y \in Y \end{cases}$

$$\Leftrightarrow \begin{cases} \bar{x}^T \cdot F \cdot \bar{y} - \bar{x}^T \cdot C_D \cdot \bar{y} \geq x^T \cdot F \cdot \bar{y} - x^T \cdot C_D \cdot \bar{y}, \ \forall x \in X \\ -\bar{x}^T \cdot F \cdot \bar{y} - \bar{x}^T \cdot C_A \cdot \bar{y} \geq -\bar{x}^T \cdot F \cdot y - \bar{x}^T \cdot C_A \cdot y, \ \forall y \in Y \end{cases} \tag{4.13}$$

In the above formulas, C_D and C_A are the behave costs matrices, and their entries at the i^{th} row, j^{th} column are the defender's behave cost and the attacker's behave cost respectively, when the attacker plays a pure strategy s_i^a and the defender plays s_j^d. Since the attacker's strategy would not influence the defender's behave cost, C_D shows identical rows; and analogously, C_A has identical columns. Therefore, we have $\bar{x}^T \cdot C_D \cdot \bar{y} = \sum_{j \in N} C_{dj} \cdot \bar{y}_j = x^T \cdot C_D \cdot \bar{y}$, and $\bar{x}^T \cdot C_A \cdot \bar{y} = \sum_{i \in M} \bar{x}_i \cdot C_{ai} = \bar{x}^T \cdot C_A \cdot y$. Thus Formula (4.13) becomes:

$$\begin{cases} \bar{x}^T \cdot F \cdot \bar{y} \geq x^T \cdot F \cdot \bar{y},\ \forall x \in X \\ -\bar{x}^T \cdot F \cdot \bar{y} \geq -\bar{x}^T \cdot F \cdot y,\ \forall y \in Y \end{cases} \cdots\cdots (4.14)$$

Formula (4.14) represents that $(\bar{x}, \bar{y})$ is a NE of game $(F, -F)$.

The proof of the above observation implies that under the condition that $\tilde{P} \cdot \tilde{P}_y \cdot \tilde{L} = P \cdot P_y \cdot L$ holds for all strategy tuples, the CPP game is a strategically zero-sum game (7). In this case, the analysis of the CPP game becomes easier. For more information of the strategically zero-sum game, interested readers are referred to Moulin and Vial.(7).

Although the condition of $\tilde{P} \cdot \tilde{P}_y \cdot \tilde{L} = P \cdot P_y \cdot L$ is a strong condition for the CPP game, it might be the case in some real industrial practice situations. For instance, if the defender and the attacker evaluate the intrusion probabilities, the consequences of an attack etc. in the same way, we would have $\tilde{P} = P$, $\tilde{P}_y = P_y$, and $\tilde{L} = L$, and then the condition holds definitely.

4.4 Solutions for the CPP Game

A solution of a game is a pair of (mixed) strategies that the players would play. In this section, the Nash equilibrium (NE), the Stackelberg equilibrium (SE), the Bayesian Nash equilibrium (BNE), and the Bayesian Stackelberg equilibrium (BSE) are used to solve the chemical plant protection game. If the defender and the attacker in the game move simultaneously, an NE must be used, while if they move sequentially (defender moves first and attacker follows), an SE must be used. If only one type of attacker is considered, then an NE or an SE should be used, and if multiple types of attackers are considered, a BNE or a BSE should be used.

4.4.1 Nash equilibrium

In the CPP game, the defender implements her daily defence plan by setting security alert levels at each entrance or in each zone. The CPP game is played simultaneously in the case that when the attacker implements his attack, he does not know any information about the defender's defence. A Nash equilibrium can be employed to predict the outcome of a simultaneous CPP game.

A pure strategy Nash equilibrium (s_a^*, s_d^*) for the CPP game satisfies the condition in Formulas (4.15) and (4.16).

$$u_a(s_a^*, s_d^*) \geq u_a(s_a, s_d^*), \quad \forall s_a \in S_a \cdots\cdots (4.15)$$

and

$$u_d(s_a^*, s_d^*) \geq u_d(s_a^*, s_d), \quad \forall s_d \in S_d \cdots\cdots (4.16)$$

A mixed strategy Nash equilibrium (x^*, y^*) for the CPP game satisfies the condition in Formulas (4.17) and (4.18).

$$x^{*T} \cdot U_a \cdot y^* \geq x^T \cdot U_a \cdot y^*, \quad \forall x \in X \cdots\cdots (4.17)$$

and

$$x^{*T} \cdot U_d \cdot y^* \geq x^{*T} \cdot U_d \cdot y, \quad \forall y \in Y \cdots\cdots (4.18)$$

The CPP game is a finite game (with two players and each player has a finite number of pure strategies), thus at least one Nash equilibrium exists. However, in most cases, there is no pure strategy Nash equilibrium for the CPP game, since generally the defender and the attacker have opposite interests.

For calculating an NE for the CPP game, the Lemke and Howson algorithm [8] can be employed.

4.4.2 Stackelberg equilibrium

If the attacker would know the defender's defence plan when he attacks, the CPP game is a sequential game. The attacker can know the defender's plan by collecting information or by continued and thorough study and observation. Although currently some game-theoretic models in the security domain propose the use of simultaneous games (e.g. [9]), three reasons enforce modellers to prefer modelling the security game as a sequential game.

Firstly, a sequential game can reflect reality better. Literature has shown that adversaries may collect 80% (sometimes even 100%) of the needed information to execute a successful attack [4]. Based on such evidence, we can assume that the attacker has complete information of the security game. However, in industrial practice regarding critical infrastructure protection, the defender usually has to implement her countermeasures (strategy) first, and when the attacker plans the attack, he not only is able to collect the information of the target, but also information of the defender's defence strategies. Thus we may assume that the attacker has both complete and perfect information of the (sequential) game.

Secondly, a sequential game might bring higher payoff to the leader (which is the first-mover, hence the defender) than the simultaneous game. This principle is called "First-mover Advantage" [10]. As we already mentioned, the attacker would also collect information on the defender's strategies. If the attacker can fully observe the defender's executed strategy, the game is a "perfect information game", or a sequential move game; if he cannot observe the defender's strategy, the game is an "imperfect information game", or a simultaneous game (see also Chapter 2); if he can partly observe the defender's strategy, the case becomes more complicated. The problem is that, the defender, who moves first, does not know whether the attacker can fully observe her strategy or not, thus she does not know whether she is playing a simultaneous game or a sequential game or even a more complicated game. In these cases, Zhuang and Vicki [10] proved that if the attacker's best-response set is a singleton, then the defender's gain from the sequential game is at least the same as that from the simultaneous game (hence the existence of the principle of "First-mover Advantage"). Knowing the "First-mover Advantage", the defender could choose to make the strategy she implemented public, to enforce the game being a sequential game.

Thirdly, the equilibria selection problem can be avoided by playing a sequential game. The Nash Equilibrium (NE) [11] is the most extensively used concept in a simultaneous game to predict the outcome. However, as shown in section 2.2.2.2, a non-zero-sum game becomes unpredictable when there are multiple NEs. In the game illustrated in Figure 2.5, if the girl would move first, she commits to the boy that she will play the strategy 'O', enforcing the boy to also play 'O', thus the game becomes predictable. Since the security game is not necessarily zero-sum, and it is possible to have multiple NEs, playing the game sequentially can make the game predictable and controllable for the defender.

In a sequential move CPP game, the Stackelberg equilibrium can be employed. Formulas (4.19) and (4.20) define the Strong Stackelberg equilibrium $(\bar{s}_a, \bar{y})$ for the CPP game.

$$\bar{y} = \text{argmax}_{y \in Y} \, U_d(\bar{s}_a \, , :) \cdot y \cdots\cdots (4.19)$$

$$\bar{s}_a = \text{argmax}_{s_a \in S_a} \, U_a(s_a, :) \cdot y \cdots\cdots (4.20)$$

Formula (4.20) denotes the fact that knowing the defender's mixed strategy y, the attacker will choose a best response pure strategy (i.e., $\bar{s}_a$). Formula (4.19) shows that the defender can also work out the attacker's best response, and thus she can play accordingly.

It is worth noting that in Formula (4.19), the defender can play a mixed strategy, which means that the attacker is only able to know the defender's defence plan, without knowing the defender's exact defence when he attacks. For instance, after a long time observation or by getting the plant's security schedule document, the attacker could know that the plant will set the SAL at the main entrance at low (high) level with a probability of 60% (40%). However, the attacker is assumed not to know whether the defender sets the SAL at the main entrance as low or as high, at the day that he attacks. This is a reasonable assumption since preparing for an attack needs time. Nonetheless, if the attacker knows the exact defence when he attacks, a pure strategy Stackelberg equilibrium $(\bar{s}_a, \bar{s}_d)$ should be employed, as defined in Formulas (4.21) and (4.22).

$$\bar{s}_d = \text{argmax}_{s_d \in S_d} U_d(\bar{s}_a, s_d) \cdots\cdots (4.21)$$

$$\bar{s}_a = \text{argmax}_{s_a \in S_a} U_a(s_a, s_d) \cdots\cdots (4.22)$$

For calculating the strong Stackelberg equilibrium for the CPP game, the MultiLPs [12] algorithm can be employed.

4.4.3 Bayesian Nash equilibrium

A Bayesian Nash equilibrium (BNE) can be used for solving the chemical plant protection game if multiple types of attacker are involved in the game and when the attackers move, they do not have any information about the defender's defence plan. The BNE can capture the defender's uncertainties on the attacker's types, while the defender's uncertainties on the attacker's parameters/payoffs will be modelled in Chapter 5.

In the Bayesian CPP game, the defender's type is deterministic, and the attacker's type can vary. However, every type of attacker knows their own type and the defender knows a priori the probabilities of each type of attacker being involved. To this end, the ex-interim Bayesian Nash Equilibrium should be employed. For more discussion on ex-ante, ex-interim, and ex-post Bayesian Nash equilibrium, interested readers are referred to Shoham and Leyton-Brown [13] and Ceppi et al. [14]

The ex-interim Bayesian Nash equilibrium $(\dot{y}, \dot{x}_1, \dot{x}_2, \dots, \dot{x}_{|TL|})$ for the CPP game can be defined as shown in Formulas (4.23) and (4.24).

$$\dot{y} = \text{argmax}_{y \in Y} \left(\sum_{t \in TL} p^t \cdot \dot{x}_t^T \cdot U_d^t\right) \cdot y \cdots\cdots (4.23)$$

$$\dot{x}_t = \text{argmax}_{x_t \in X_t} x_t^T \cdot U_a^t \cdot \dot{y}, t \in TL \cdots\cdots (4.24)$$

In which U_a^t and U_d^t are the attacker and the defender's payoff matrix respectively, in case of an attacker type $t \in TL$; X_t is the mixed strategy space for attacker t; p^t is the prior probability of attacker t, as shown in Formula (4.2).

The most well-known approach for solving a Bayesian game is by using the Harsanyi transformation, which transfers an incomplete information game (i.e., in the CPP game, with the multiple types of attackers) to a complete but imperfect information game. However, this approach computes the ex-ante Bayesian Nash equilibrium. For the ex-interim Bayesian Nash equilibrium, new algorithms are needed. Ceppi et al. [14] developed three algorithms for computing the interim BNE for two-player strategic form games, namely the B-PNS (based on support enumeration), the B-LC (based on linear complementarity formulation), and the B-SGC (based on mixed integer linear programming). It is worth noting that all these three algorithms have a high computational complexity, thus before implementing them, the dominance checking should be carried out on the game first.

4.4.4 Bayesian Stackelberg equilibrium

A Bayesian Stackelberg equilibrium (BSE) can be employed to solve the chemical plant protection game, if multiple types of attackers are considered, and the attackers know the defender's defence plan when they attack.

The BSE $(\tilde{y}, \tilde{s}_a^1, \tilde{s}_a^2, \dots, \tilde{s}_a^{|TL|})$ for the CPP game can be defined as shown in Formulas (4.25) and (4.26).

$$\tilde{y} = \text{argmax}_{y \in Y} \sum\nolimits_{t \in TL} p^t \cdot U_d^t(\tilde{s}_a^t, :) \cdot y \cdots\cdots(4.25)$$

$$\tilde{s}_a^t = \text{argmax}_{s_a^t \in S_a^t} U_a^t(s_a^t, :) \cdot y,\ t \in TL \cdots\cdots(4.26)$$

A straightforward approach for computing the BSE for the CPP game is to solve the linear programming problem for each combination of the attacker's best responses. This is quite similar to the MultiLP algorithm, and in total $\prod_{t \in TL} m^l$ linear programming problems need to be solved. Paruchuri et al. [15] proposed a mixed integer linear programming (MILP) based algorithm for speeding up the computing of BSE for large scale Bayesian Stackelberg games, namely, the DOBSS algorithm. It is worth noting that the BSE calculated by the DOBSS algorithm is also a Strong Stackelberg Equilibrium, that is to say, the "breaking-tie" assumption (see section 2.2.2.3) is applied.

Table 4. 2. The MovLib Algorithm

```
Input:
        A sequential CPP game with multiple types of attackers $(U_a^t, U_d^t, p^t), t \in TL$;
        an BSE of the game $(\tilde{y}, \tilde{s}_a^1, \tilde{s}_a^2, \dots, \tilde{s}_a^{|TL|})$, outputted from the DOBSS algorithm;
        a small constant positive value $\varepsilon$.
Output:
        A Modified BSE $(\bar{\bar{y}}, \bar{\bar{s}}_a^1, \bar{\bar{s}}_a^2, \dots, \bar{\bar{s}}_a^{|TL|})$, or fail.
Solve the Linear Programming (LP):
$\begin{cases} Payoff = \max_{y \in Y} \sum_{t \in TL} p^t \cdot U_d^t(\tilde{s}_a^t, :) \cdot y \\ s.t.\ U_a^t(\tilde{s}_a^t, :) \cdot y \geq U_a^t(s_a^t, :) \cdot y + \varepsilon, \quad \forall s_a^t \in S_a^t - \{\tilde{s}_a^t\}, t \in TL \end{cases}$
If the LP feasible
    $\bar{\bar{y}} \leftarrow$ the optimal point $y$, and $\bar{\bar{s}}_a^t \leftarrow \tilde{s}_a^t$;
    Return the Modified BSE $(\bar{\bar{y}}, \bar{\bar{s}}_a^1, \bar{\bar{s}}_a^2, \dots, \bar{\bar{s}}_a^{|TL|})$.
If the LP infeasible
    Return failure.
```

Table 4.2 provides an algorithm named as 'MovLib' to slightly modify the BSE calculated by the DOBSS algorithm. The idea of the MovLib algorithm is that: based on the BSE, the defender moves a little bit from her BSE strategy, which is absolutely optimal for her but the "breaking-tie" assumption is required, to a strategy that is a bit less optimal to her but the "breaking-tie" assumption is no longer required.

Inputs for the MovLib are: payoff matrices for a sequential move CPP game with multiple types of attackers, i.e., U_a^t and U_d^t; the prior probability of the occurence of each type of attacker, i.e., p^t ; a Bayesian Stackelberg Equilibrium of the CPP game, i.e., $(\tilde{y}, \tilde{s}_a^1, \tilde{s}_a^2, \dots, \tilde{s}_a^{|TL|})$, which can be calculated by the DOBSS algorithm; a small constant positive value ε, which can be, for instance, 0.1. The output of the MovLib algorithm is a Modified BSE $(\bar{\bar{y}}, \bar{\bar{s}}_a^1, \bar{\bar{s}}_a^2, \dots, \bar{\bar{s}}_a^{|TL|})$. The main body of the algorithm is the Linear Programming problem. The cost function of the LP denotes that the defender is optimizing her payoff, knowing that the attacker t would play his strategy $\tilde{s}_a^t$. The constraint of the LP makes sure that the attacker t's payoff by playing strategy $\tilde{s}_a^t$ is at least ε more than his payoff by playing any other strategies. The constraint makes sure that $\tilde{s}_a^t$ is the unique best response strategy for the attacker and therefore there is no "breaking-tie" problem anymore.

The MovLib algorithm can be failure, if there is a $\tilde{s}_a^t$ that for any $y \in Y$, the constraint of the LP in the MovLib algorithm can not hold. For instance, if attacker t has a strategy s_a^t that has exactly the same payoff (no matter what the defender's strategy is) as strategy $\tilde{s}_a^t$, then the constraint in the algorithm would not hold, no matter how small the ε is.

4.5 CPP game from an industrial practice point of view

Developing, solving and using the chemical plant protection game needs massive quantitative inputs and the outputs of the CPP game are also quantitative results. In this section, discussions on how to obtain these inputs from conventional security risk assessment methodologies and on how to translate the game theoretic outputs to these conventional approaches are given.

The American Petroleum Institute (API) recommends to employ Security Risk Assessment (henceforth, the API SRA) methodologies in the petroleum and petrochemical industries. The API SRA indeed provides a systematic and practically implementable framework for security risk analysis in the process industries. Since first published in 2004, it was re-edited in 2013, and it has been extensively used in the industrial practice. In this section, we choose the API SRA as the baseline methodology, and show how to obtain inputs for the CPP game from the API SRA and to translate the game theoretic results back to the API SRA terminologies. For information about the API SRA methodology, please see section 2.1.4 in this book, or see the document [(3)].

4.5.1 Input analysis

The following inputs are needed for calculating the payoff matrices for the CPP game: (i) the prior probabilities of each type of attackers (i.e., p^t); (ii) the probabilities that the intruder can pass an entrance (i.e., p_{rj}^p) or a zone (i.e., p_i^z), under certain defence and certain attack scenario; (iii) conditional probabilities that an attack will be implemented if the attacker has successfully reached the target (i.e., P_y); (iv) estimated consequences of a successful attack (i.e., L); and (v) the defence and the attack costs, say, the C_d and C_a, respectively.

The prior probabilities of each type of attacker can be calculated according to the threat level linked to each attacker, as was discussed in section 4.3.1. In the API SRA methodology, the SRA team first decides what kind of threat the plant is faced with, and obtains the threat list TL. The team further estimates a threat level for each type of threat, according to the criteria shown in Table 4.3, and obtains the threat score list TS. Based on TL and TS, the prior probabilities of each attacker can be calculated based on Formula (4.2) (see section 4.3.1).

Table 4. 3. Threat ranking criteria (adopted from the API document)

API SRA Methodology	
Threat Level	Description [a]
1—Very low	Indicates little or no credible evidence of capability or intent and no history of actual or planned threats against the asset or similar assets (e.g. "no expected attack in the life of the facility's operation").
2—Low	Indicates that there is a low threat against the asset or similar assets and that few known adversaries would pose a threat to the asset (e.g. "≥ 1 event is possible in the life of the facility's operation").
3—Medium	indicates that there is a possible threat to the asset or similar assets based on the threat's desire to compromise similar assets, but no specific threat exists for the facility or asset (e.g. "≥ 1 event in 10 years of the facility's operation").
4—High	Indicates that a credible threat exists against the asset or similar assets based on knowledge of the threat's capability and intent to attack the asset or similar assets, and some indication exists of the threat specific to the company, facility, or asset (e.g. "≥ 1 event in 5 years of the facility's operation").
5—Very high	Indicates that a credible threat exists against the asset or similar assets; that the threat demonstrates the capability and intent to launch an attack; that the subject asset or similar assets are targeted or attacked on a frequently recurring basis; and that the frequency of an attack over the life of the asset is very high (e.g. "1 event/ event per year").

[a] User defined values should be applied.

The probability that an intruder can successfully reach the target depends on the defender's defence scenario and on the attacker's intrusion path, and thus, it is a function of both the defender and the attacker's actions. For instance, the probability would be quite low if the defender deploys an x-ray scanner at the main entrance, and the attacker chooses to intrude with an explosive device; while the probability could be higher if the attacker wants to implement an attack by switching off a key safety valve in the plant since the x-ray scanner does not work on this attack scenario. Table 4.4 shows how the API SRA methodology quantifies these probabilities.

The conditional probability of a successful attack and the estimated consequences depend only on the attacker's strategy, that is, which facility he wants to attack, and with what scenario. For instance, to cause losses from an explosion on an oil storage tank with an explosive device can be easier than to cause them by switching off a key safety valve, since the latter scenario needs more professional knowledge. Moreover, different attack scenarios will of course result in different consequences.

Table 4. 4. Vulnerability scores and corresponding quantitative data (adopted from the API document)

API SRA Methodology			
VL*	D	CPS	Description
1	Very low	[0.0, 0.2]	Indicates that multiple layers of effective security measures to deter, detect, delay, respond to, and recover from the threat exist, and the chance that the adversary would be readily able to succeed at the act is very low.
2	Low	(0.2, 0.4]	Indicates that there are effective security measures in place to deter, detect, delay, respond, and recover; however, at least one weakness exists that a threat would be able to exploit with some effort to evade or defeat the countermeasure.
3	Medium	(0.4, 0.6]	Indicates that although there are some effective security measures in place to deter, detect, delay, respond, and recover, but there is not a complete and effective application of these security strategies and so the asset or the existing countermeasures could still be compromised.
4	High	(0.6, 0.8]	Indicates there are some security measures to deter, detect, delay, respond, and recover, but there is not a complete or effective application of these security strategies and so the adversary could succeed at the act relatively easily.
5	Very high	(0.8, 1.0]	Indicates that there are very ineffective security measures currently in place to deter, detect, delay, respond, and recover, and so the adversary would easily be able to succeed.

*VL: Vulnerability Level; D: Descriptor; CPS: Conditional Probability of Success

In the API SRA methodology, there is no separated assessment of the intrusion probabilities and the conditional success probabilities. Table 4.4 may also be used for obtaining the conditional success probability. For the consequences, the API SRA uses a ranking method to measure the consequences of an event. However, the scores that an event will receive are based on quantitative descriptions, as shown in Table 4.5. The idea is to use the quantitative data from the left-hand column directly, instead of using the scores in the right-hand column. There are 5 different aspects of the quantitative data, namely, the casualties, the environment impacts, direct economic loss, business interruption, and reputation impacts. A set of coefficients are needed to utilize them into monetary numbers. For instance, to transfer a casualty as 5.8 million euro [16, 17].

The defence and the attack costs depend on the defender's defence plan and on the attacker's attack scenario, respectively. Generally speaking, higher security alert levels at each entrance and in each zone will secure the plant better, however, the defence cost will also be higher. A well trained attacker will always increase the success probability of an attack scenario as well as increase the consequence of the attack. In the API SRA, at the mitigation step (step 5.1 in the API SRA document), it is clearly mentioned that the costs of mitigation options should be considered. Thus we can get the C_d for the defender. For the attack costs, the API SRA does not point out how to estimate it. However, the SRA team could make an estimation of the attacker's cost based on an attack scenario. For instance, different types of bombs, different bombers (well-trained or not), different vehicles related to different attack scenarios, and a different scenario related to a different attack cost, can be conceptualized and mapped.

Table 4. 5. Consequence ranking and the corresponding quantitative data (adopted from the API document)

API SRA Methodology	
Description	Ranking
a) Possibility of minor injury on-site; no fatalities or injuries anticipated off site. b) No environmental impacts. c) Up to $X loss in property damage. d) Very short-term (up to X weeks) business interruption/expense. e) Very low or no impact or loss of reputation or business viability; mentioned in local press.	1
a) On-site injuries that are not widespread but only in the vicinity of the incident location; no fatalities or injuries anticipated off site. b) Minor environmental impacts to immediate incident site area only, less than X year(s) to recover. c) $X to $X loss in property damage. d) Short-term (>X week to Y months) business interruption/expense. e) Low loss of reputation or business viability; query by regulatory agency; significant local press coverage.	2
a) Possibility of widespread on-site serious injuries; no fatalities or injuries anticipated off site. b) Environmental impact on-site and/or minor off-site impact, Y year(s) to recover. c) Over $X to $X loss in property damage. d) Medium-term (Y to Z months) business interruption/expense. e) Medium loss of reputation or business viability; attention of regulatory agencies; national press coverage.	3
a) Possibility of X to Y on-site fatalities; possibility of off-site injuries. b) Very large environmental impact on-site and/or large off-site impact, between Y and Z years to recover. c) Over $X to $X loss in property damage. d) Long-term (X to Y years) business interruption/expense. e) High loss of reputation or business viability; prosecution by regulator; extensive national press coverage.	4
a) Possibility of any off-site fatalities from large-scale toxic or flammable release; possibility of multiple on-site fatalities. b) Major environmental impact on-site and/or off site (e.g. large-scale toxic contamination of public waterway), more than XX years/poor chance of recovery. c) Over $X loss in property damage. d) Very long-term (>X years) business interruption/expense; large-scale disruption to the national economy, public or private operations; loss of critical data. e) Very high loss of reputation or business viability; international press coverage.	5

Though the API SRA framework provides some data for the CPP game, some more work is needed. The API SRA focuses on estimating threats/probabilities/consequences from the defender's view point. However, it is not necessary that the attacker and the defender have the same preference, which was already shown in the payoff definitions (see, Formulas (4.7) and (4.8)) of the CPP game. For example, an important facility might not be an attractive target to an attacker, since the attacker has his own preference. The attacker would plan his attack according to his own preference and on his estimation of the defender's strategy. Moreover, the API SRA framework does not explicitly evaluate if the attacker would know her defence plan or not. In a game theoretic terminology, if the attacker knows the defender's strategy, the game is said to being played sequentially, and a (Bayesian) Stackelberg equilibrium should therefore be used. Otherwise, if the attacker would not know, the game is said to being played simultaneously, and a (Bayesian) Nash equilibrium therefore should be employed.

Instead of being obtained via security experts, the vulnerability related inputs (e.g., p_i^z) can also be calculated by a function named the "Contest Success Function (CSF) [(2)], for theoretical study purpose. The CSF investigates the success probability of each player in a multiple players contest. The success probability is determined both by the player and his opponents' contest effort and can be further formulated as:

$$p^i(e) = \alpha_i \cdot e_i^r \Big/ \sum_{j \in \aleph} \alpha_j \cdot e_j^r \cdots\cdots\cdots\cdots (4.27)$$

In which e denotes the players' effort; $r > 0$ and $\alpha > 0$ are constant real numbers which should be determined by the contest circumstance. For more information of the CSF, readers are referred to Clark and Rijs [(18)], Skaperdas [(2)], and Guan and Zhuang [(19)].

4.5.2 Output analysis

The Chemical Plant Protection game outputs an equilibrium strategy pair (s) $(\bar{s}_d, \bar{s}_a^t)$ (might also be a mixed strategy pair) and a corresponding equilibrium payoff (s) $(\bar{u}_d, \bar{u}_a^t)$. In this section we show how to translate these outputs to the API SRA terminologies.

The defender's strategy is modelled as setting security alert levels at each entrance and zone. In industrial practice, different security alert levels represent different combinations of security countermeasures (see Table 4.1). Therefore, the equilibrium strategy $\bar{\mathrm{s}}_\mathrm{d}$ can be mapped to the proposed countermeasures list CML in the API SRA methodology. If the equilibrium strategy is a mixed strategy, which denotes the probabilities of setting different security alert levels, then it can be mapped to the prioritization procedure of the countermeasures list. Furthermore, the countermeasures with a higher mixed strategy probability should be assigned a higher priority.

The attacker's equilibrium strategy $\bar{s}_a^t$ clearly indicates which target the attacker would attack and what attack scenario will be employed. It is anti-intuitive that knowing this, why shouldn't the defender enhance the protection of the attacker's target? This is a result of the intelligent interactions between the defender and the attacker: if the defender protects the equilibrium target better, then the attacker would also deviate from his current target to a new target.

The defender's equilibrium payoff $\bar{u}_d$ reflects the mitigated security risk, and in the API SRA methodology, it is denoted as R^2. The attacker's equilibrium payoff $\bar{u}_a^t$ reflects the attacker's attack motivation, and in the API SRA methodology, it is named "degree of interest".

It is worth noting that the attacker's equilibrium strategy can also be a mixed strategy. Denote the likelihood that target tgt would be attacked as: $\mathrm{lk}_\mathrm{tgt} = \sum_{i \in M(tgt)} x_i$, in which

$M(tgt) \subset M$ denotes all the attacker pure strategies that take tgt as the attack target. In the API SRA methodology, $\mathrm{lk}_{\mathrm{tgt}}$ is represented as "attractiveness", as shown in Table 4.6.

Table 4. 6. Attractiveness ranking level (adopted from the API document)

API SRA Methodology			
RL *	D	CPA	Threat Ranking
1	Very low	[0.0, 0.2]	Threat would have little to no level of interest in the asset.
2	Low	(0.2, 0.4]	Threat would have some degree of interest in the asset, but it is not likely to be of interest compared to other assets.
3	Medium	(0.4, 0.6]	Threat would have a moderate degree of interest in the asset relative to other assets.
4	High	(0.6, 0.8]	Threat would have a high degree of interest in the asset relative to other assets.
5	Very high	(0.8, 1.0]	Threat would have a very high degree of interest in the asset, and it is a preferred choice relative to other assets.

*RL: Ranking Level; D: Descriptor; CPA: Conditional Probability of the Act

4.6 Case study: applying the CPP game to a refinery

4.6.1 Case study setting

Figure 4.3 shows the layout of a refinery, which is also used as a case study in the API SRA document [(3)] and in Lee et al. [(20)].

The API SRA methodology concluded that the main gate (MG), the central control room (CCR), the co-gen unit and control room (CgCR), dock #1 (D1), and the tank farm (TF) are critical assets in this refinery, as described in the first page of form 1 in the API SRA document. We added the administration building (AdB), the electrical supply station (ESS), and the production facility (PF) to this list, since these assets also have important roles in the chemical plant defence in our opinion.

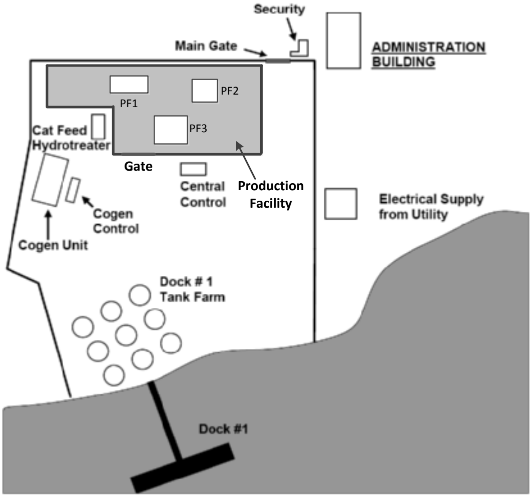


Figure 4. 3. Layout of a refinery (PF = Production Facility)

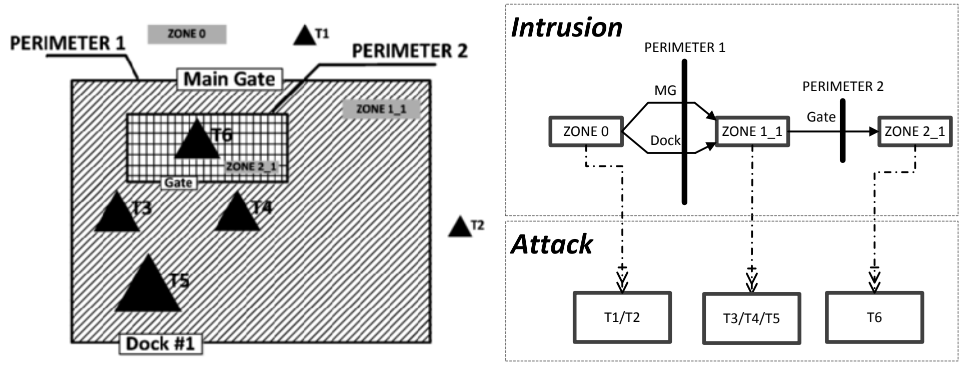
The main gate (MG) and Dock #1 (D1) are considered as critical assets since attackers may use these assets to intrude into the plant. Furthermore, D1 is located at the waterside and a tank farm (TF) is close to D1. If an attacker intrudes from D1 to attack the TF, the probability would be high and the consequence would be severe (especially the environmental damage). The MG is also vulnerable. Many people (the company's own employees as well as visitors of the plant) and vehicles are passing by the MG every day and an illegal intruder may try to act as one of the authorized people to intrude the plant. Countermeasures such as an access card system, CCTV, x-ray machine, and crash rated barrier etc. can therefore for instance be deployed at the MG and D1. The central control room (CCR) controls all production process in the plant, and manages security communications in the plant. If the CCR is damaged, there is a possibility for loss of lives, and a long recovering time is needed. The co-gen unit and control room (CgCR) steams production and generate electrical power for the plant. There is limited redundancy in the electrical system. If the CgCR would be attacked, the plant would suffer at least a two days business interruption. The tank farm (TF) stores crude, intermediate, waste and finished liquid hydrocarbons. An attack on the TF may result in direct economic loss, environmental damage, casualties, business interruption, and what have you. We learned the above asset information from Form 1 of Example 2 in the API SRA document [(3)].

Furthermore, as already mentioned, we consider the production facility (PF), the administration building (AdB) and the electrical supply station (ESS) also as critical assets. If the production facilities would be attacked, besides the direct economic loss and possible casualties, the plant needs a long repairing time, resulting in business interruption for a long time. The administration building, although less vulnerable, may store important technique documents. Technique thieves can be quite interested in intruding into the building. The electrical supply station (ESS) provides electricity to the plant, and if being attacked, direct economic loss and business interruption would exist.

Figure 4.4 (a) shows the conceptual description of the refinery, based on the above analysis. Table 4.7 shows the corresponding names of symbols used in Figure 4.3. Part of the perimeter 1 is the coastline, as we mentioned earlier in this chapter that a geographical border can also be a perimeter. There are multiple tanks in the tank farm and multiple production facilities (PF) in zone 2_1. However, on the one hand every tank or PF is similar in this illustrative example, and on the other hand, due to the possible existence of domino effects [(21)], if one tank or PF would be attacked, other tanks may also be damaged. Therefore, we simplify the whole tank farm and all the PFs as target 5 and target 6 respectively.

Figure 4.4 (b) demonstrates the intrusion and attack procedure. For each attack scenario, there are in total ten possible combinations of intrusion paths and targets, as shown in Table 4.8. Intrusion by stepping over the perimeter is ignored in this case study, for simplification reasons.

Three types of adversaries are mentioned in the API SRA result of this case study, namely, terrorists, disgruntled employees, and activists. The plant's general physical intrusion detection approach does not work for the disgruntled employees, thus this type of threat is excluded from this case study that we use further in this chapter. The terrorist mainly focuses on causing maximum damage to the plant as well as to the surrounded residents, while the activist is mostly concerning in shutting down the refinery operations, and these two threats have threat levels of 3 and 4 respectively (note that the threat level here only represents the likelihood (threat score) of the occurrence of the corresponding type of attacker, see also Figure 2.3 in Chapter 2).



(a) Abstract Description of the Plant

(b) Intrusion and Attack Procedure

Figure 4. 4. Formalized representation of the refinery

Table 4. 7. Symbols map between Figure 4.3 and Figure 4.4 (a)

Symbol in Figure 4.4 (a)	Symbol in Figure 4.3
ZONE0	Outdoor Area
ZONE1_1	Area within Enclosure
ZONE2_1	Production Facility area
PERIMETER 1	the boundary of the plant
PERIMETER 2	The boundary of the production facility
Main Gate	Main gate
Dock #1	Dock #1
Gate	The entrance to the production facility
T1	Administration Building
T2	Electrical Supply from Utility
T3	Cogen Unit/ Cogen Control/Cat Feed
T4	Central control
T5	Tank Farm
T6	Production facilities in production facility area

Table 4. 8. The attackers' intrusion and attack paths

No.	Path & Target	No.	Path & Target
PaT_1	Zone 0→T1	PaT_6	Zone 0→D1→Zone 1→T3
PaT_2	Zone 0→T2	PaT_7	Zone 0→D1→Zone 1→T4
PaT_3	Zone 0→MG→Zone 1→T3	PaT_8	Zone 0→D1→Zone 1→T5
PaT_4	Zone 0→MG→Zone 1→T4	PaT_9	Zone 0→MG→Zone 1→Gate→Zone 2→T6
PaT_5	Zone 0→MG→Zone 1→T5	PaT_{10}	Zone 0→D1→Zone 1→Gate→Zone 2→T6

4.6.2 Chemical Plant Protection game modelling

4.6.2.1 Players

Players in this case study are the company security management team (e.g., a API SRA team), who acts as the defender, and the terrorist as well as the activists, who act as the possible adversaries. Therefore, the game would be a Bayesian game and the attacker has two different types. According to Formula (4.2), the prior probabilities that the attacker would be a terrorist and an activist are $3/7$ and $4/7$, respectively.

4.6.2.2 Strategies

A defender's pure strategy is the combination of different security alert levels (SAL) at each entrance and zone. In this case study, we assume that three different SALs (denoting as Green(1)/Yellow(2)/Red(3)) are possible at each entrance or zone. There are in total three entrances and three (sub-) zones in the case study, and we further list them as (i) Zone 0; (ii) the MG; (iii) D1; (iv) Zone 1; (v) the Gate; (vi) Zone 2. Subsequently, we employ a cross product of six digital numbers, i.e., $s_d = d_{Z0} \times d_{MG} \times d_{D1} \times d_{Z1} \times d_{Gate} \times d_{Z2}$, to denote a defender's pure strategy, of which d_{Z0} denotes the security alert level at Zone 0, d_{MG} denotes the security alert level at the Main Entrance, and so forth. For instance, an $s_d = 2 \times 1 \times 3 \times 2 \times 3 \times 1$ represents that the SALs at Zone 0, the MG, D1, ZONE 1, the Gate, and Zone 2, are 2 (or Yellow), 1 (or Green), 3 (or Red), 2 (or Yellow), 3 (or Red), and 1 (or Green), respectively.

An attacker's pure strategy consists of a target, an intrusion path, and an attack scenario. The combinations of intrusion paths and targets are shown in Table 4.8. To simplify the case study, only one attack scenario is considered for each type of attacker. The terrorist is assumed to employ a vehicle-borne improvised explosive device (VBIED) as his attack scenario. We assume an environmental activist (EA) aiming to shut down the operation of the refinery as the scenario of the activist.

Table 4. 9. the terrorist's pure strategy list

Index	Strategy
s_{v1}	$T1 \times VBIED$
s_{v2}	$T2 \times VBIED$
s_{v3}	$T3 \times MG \times VBIED$
s_{v4}	$T4 \times MG \times VBIED$
s_{v5}	$T5 \times MG \times VBIED$
s_{v6}	$T6 \times MG \times Gate \times VBIED$

Table 4. 10. the activist's pure strategy list

Index	Strategy
s_{e1}	$T2 \times EA$
s_{e2}	$T3 \times MG \times EA$
s_{e3}	$T3 \times Dock \times EA$
s_{e4}	$T4 \times MG \times EA$
s_{e5}	$T4 \times Dock \times EA$
s_{e6}	$T6 \times MG \times Gate \times EA$
s_{e7}	$T6 \times Dock \times Gate \times EA$

In a VBIED attack, a car or a truck should be involved. Therefore, the terrorist would not be able to intrude from Dock #1 (D1). The environmental activist aims to shut down the plant, instead of causing damages to the plant, thus we assume that the administration building and the tank farm would not be his attack target. Tables 4.9 and 4.10 illustrate all the pure strategies for the terrorist and for the activist respectively.

4.6.2.3 Payoffs

Tables 4.11 and 4.12 show the probabilities (i.e., P_i^z and P_r^p) that the attacker would successfully pass an entrance or a zone, for the terrorist and for the activist respectively. The "Typical" column shows the entrance or the zone. The "p_d" column shows the defender's estimation of the probability. The "$\tilde{p}_a$" columns (i.e., $\tilde{p}_a^{min}$, $\tilde{p}_a^{max}$, and $\tilde{p}_a^{nominal}$) illustrate the attacker's values of the probabilities. The data is given separately for the defender and the attacker, since they can evaluate the same parameter in different results. In a sequential game setting, the $\tilde{p}_a$ columns are the defender's estimation of the attacker's data. The $\tilde{p}_a^{min}$, $\tilde{p}_a^{max}$, and $\tilde{p}_a^{nominal}$ columns thus denote the defender minimal, maximal, and nominal estimation of the parameters. In a simultaneous game setting, the $\tilde{p}_a^{nominal}$ column is the attacker's own estimation and it is assumed to be known by the defender (cfr. the 'common knowledge' assumption). The duration that the attacker stays in a zone affects his success probability. Generally speaking, in the same zone with the same security alert level (e.g., patrolling intensity), the longer the attacker stays in the zone, the more likely that he will be detected. Therefore, the "From" and "To" column are added to indicate different routes in each zone.

Table 4. 11. Basic probabilities of successful intrusion for the terrorist

Typical	From	To	p_d	$\tilde{p}_a^{min}$	$\tilde{p}_a^{max}$	$\tilde{p}_a^{nominal}$	Coe_2	Coe_3
zone0		T1 or T2	0.95	0.95	0.99	0.95	0.68	0.45
MG			0.3	0.3	0.5	0.3	0.65	0.38
zone1	MG	Gate	0.78	0.78	0.84	0.78	0.68	0.46
zone1	MG	T3	0.8	0.8	0.9	0.8	0.68	0.46
zone1	MG	T4	0.8	0.8	0.9	0.8	0.68	0.46
zone1	MG	T5	0.6	0.6	0.66	0.6	0.68	0.46
Gate			0.3	0.3	0.34	0.3	0.61	0.32
Zone2	Gate	T6	0.9	0.9	0.99	0.9	0.66	0.39

Probabilities in the "p_d" and "$\tilde{p}_a$" columns are estimated based on a lowest security alert level (e.g., a GREEN level or a level 1). Hereafter, a GREEN level is identical to level 1, a YELLOW level equals level 2, and a RED level is the same as level 3). If the "typical" has a higher SAL, such as a YELLOW (2) or a RED (3) level, extra tables should be provided by the security experts. In this study, for the sake of simplicity, we do not provide extra tables. Instead, the probabilities are assumed to decrease concavely [(10)]. The "Coe_2" and "Coe_3" columns show the decline coefficients, for a YELLOW level and for a RED level, respectively. Figure 4.5 demonstrates the concave property of these coefficients. Different lines denote different entrances or zones. Coefficients for the GREEN level are all set to be one, while coefficients for the YELLOW level and for the RED level are data from the "Coe_2" and "Coe_3" columns respectively. For example, in Table 4.11, $p_d(MG) = 0.3$ represents that, in a GREEN security alert level, the defender thinks that a terrorist with a VBIED scenario would have a probability of 0.3 to successfully pass the main entrance. While the $Coe_2(MG) = 0.65$ means that, if a YELLOW security alert level would be implemented at the main entrance, the defender thinks that a terrorist with a VBIED scenario would represent a probability of $0.3 \times 0.65 = 0.195$ to successfully pass the main entrance.

Table 4. 12. Basic probabilities of successful intrusion for the activist

Typical	From	To	p_d	$\tilde{p}_a^{min}$	$\tilde{p}_a^{max}$	$\tilde{p}_a^{nominal}$	Coe_2	Coe_3
zone0		T1 or T2	0.95	0.9	0.97	0.95	0.68	0.45
MG			0.3	0.2	0.32	0.3	0.65	0.38
Dock			0.28	0.26	0.29	0.28	0.53	0.3
zone1	MG	Gate	0.78	0.7	0.8	0.78	0.68	0.46
zone1	MG	T3	0.8	0.72	0.8	0.8	0.68	0.46
zone1	MG	T4	0.8	0.72	0.8	0.8	0.68	0.46
zone1	Dock	Gate	0.78	0.7	0.78	0.78	0.68	0.46
zone1	Dock	T3	0.8	0.74	0.8	0.8	0.68	0.46
zone1	Dock	T4	0.8	0.78	0.8	0.8	0.68	0.46
Gate			0.2	0.15	0.21	0.2	0.61	0.32
Zone2	Gate	T6	0.9	0.8	0.9	0.9	0.66	0.39

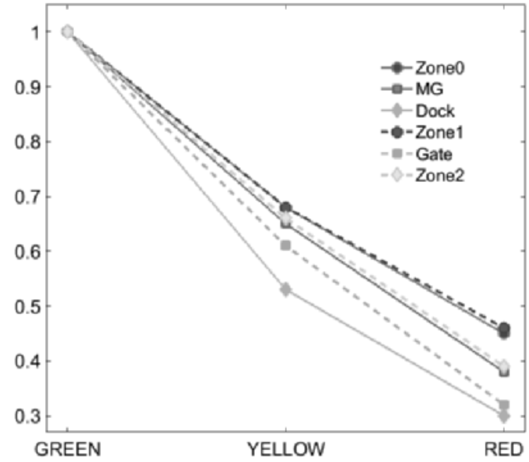


Figure 4. 5. The coefficients in Table 4.11 and 4.12

Tables 4.13 and 4.14 provide the estimations of conditional probabilities that an attack would be successfully executed and the accompanying estimated consequences/gains. The p_y column denotes the defender's estimation of the conditional probabilities that an attack would succeed under the condition that the attacker already reaches the target. The $\tilde{p}_y$ columns represent the attacker's estimation; the explanation of the three columns are similar to the columns in Tables 4.11 and 4.12. The L and $\tilde{L}$ columns denote the estimated losses and gains, for the defender and for the attacker respectively. Tables 4.15 and 4.16 further give the materialized defensive and attack costs respectively.

It is worth noting that data in Tables 4.11 to 4.16 are all illustrative data in this case study. If the CPP game would be used in industrial practice, they should be provided by security experts, for instance, by a company security team, as we discussed in section 4.5.

Table 4. 13. If already arrived at the target, probabilities of damage and consequences (k€), for terrorist

Target	p_y	$\widetilde{p}_y^{min}$	$\widetilde{p}_y^{max}$	$\widetilde{p}_y^{nominal}$	L	$\widetilde{L}^{min}$	$\widetilde{L}^{max}$	$\widetilde{L}^{nominal}$
T1	0.1	0.1	0.12	0.1	1000	1100	1300	1200
T2	0.9	0.9	0.95	0.9	100	120	140	130
T3	0.7	0.7	0.8	0.7	300	240	260	250
T4	0.6	0.6	0.8	0.6	800	880	920	900
T5	0.9	0.9	0.99	0.9	2000	3000	3300	3000
T6	0.99	0.99	1	0.99	10000	4900	5200	5000

Table 4. 14. If already arrived at the target, probabilities of damage and consequences (k€), for activist

Target	p_y	$\widetilde{p}_y^{min}$	$\widetilde{p}_y^{max}$	$\widetilde{p}_y^{nominal}$	L	$\widetilde{L}^{min}$	$\widetilde{L}^{max}$	$\widetilde{L}^{nominal}$
T2	0.7	0.68	0.72	0.7	200	105	115	110
T3	0.7	0.4	0.6	0.5	300	280	310	300
T4	0.7	0.56	0.64	0.6	850	880	910	900
T6	0.9	0.85	0.95	0.9	1000	1800	2200	2000

Table 4. 15. Materialized costs (k€) for defender

	Zone0	MG	Dock	Zone1	Gate	Zone2
SAL:GREEN	40	20	20	20	20	20
SAL:YELLOW	60	30	25	30	25	30
SAL:RED	100	50	40	50	40	50

Table 4. 16. Materialized costs (k€) for attackers

Terrorist			Activist		
C_a^{min}	C_a^{max}	$C_a^{nominal}$	C_a^{min}	C_a^{max}	$C_a^{nominal}$
5	15	10	0.2	2	1

4.6.3 CPP Game results

The game modelled for the case study is a 2-player game. The defender is determined, while the attacker can be either a terrorist or an activist. The prior probabilities of these two types of attackers are $p^t = (3/7,4/7)$. The defender has $n = 3^{1+\sum_{r=1}^{2}(ent(r)+sub(r))} = 729$ pure strategies. The terrorist has $m^{terr} = 6$ pure strategies and the activist has $m^{acti} = 7$ pure strategies, as shown in Tables 4.9 and 4.10. If Formulas (4.7) and (4.8), are filled in with data from tables shown in section 4.6.2, the defender's payoff matrices U_d^{terr} and U_d^{acti}, the attacker's nominal payoff matrices U_a^{terr} and U_a^{acti}, can be obtained.

The four solutions discussed in section 4.4 are calculated for the case study. For the sake of clarity, we first show the results of the CPP game of the case study by considering one type of attacker, in section 4.6.3.1. Subsequently, we discuss the results of the game by considering multiple types of attackers (i.e., both the terrorist and the activist) in section 4.6.3.2.

4.6.3.1 Single attacker type: the Activist

The Nash Equilibrium and the (Strong) Stackelberg Equilibrium are investigated for the case study by considering only the activist attacker. Results for the game only considering the terrorist attacker can be obtained analogously.

Nash equilibrium

In the case study (actually, hereafter in the entire section 4.6.3.1, the "case study" means the case study that only considers the activist and the defender), the defender and the activist are outguessing each other's pure strategy, and they have negatively correlated interests in the game. Therefore, there is no pure strategy Nash Equilibrium for the developed game. Table 4.17 illustrates the defender's best pure strategy responses (the second column) to each strategy of the activist (the first column), according to the defender's payoff matrix U_d^{acti}. The Activist's best response strategy to the defender's best response strategy is shown in the third column in the table. For instance, in the first row of Table 4.17, if the activist would play the strategy s_{e1}, then the defender's best response (a strategy that brings the player the highest payoff) is to play the strategy $2 \times 1 \times 1 \times 1 \times 1 \times 1$. Similarly, if the activist knows that the defender is going to play the strategy $2 \times 1 \times 1 \times 1 \times 1 \times 1$, then his best response is to play s_{e4}. Therefore, in Table 4.17, if in a row that the activist strategy (the first column) equals the activist's best response to the defender's best response (the third column), then it means that the activist's strategy and the defender's strategy in this row are mutual best responses to each other, and furthermore, the row shows a pure strategy NE for the CPP game. There is no row in Table 4.17 that the activist strategy column equals the third column, which means that no pair of the players' strategies satisfy the mutual best responses condition, thus no pure Nash Equilibrium exists in the game.

Table 4. 17. Players' pure strategy best responses to their opponent's strategies

Activist Strategy	Defender's best response	Activist's Best response to the defender's best response
s_{e1}	2×1×1×1×1×1	s_{e4}
s_{e2}	1×2×1×1×1×1	s_{e5}
s_{e3}	1×1×2×1×1×1	s_{e4}
s_{e4}	1×3×1×2×1×1	s_{e5}
s_{e5}	1×1×3×2×1×1	s_{e4}
s_{e6}	1×1×1×1×2×1	s_{e4}
s_{e7}	1×1×2×1×2×1	s_{e4}

By employing the Lemke-Howson algorithm (8) and taking U_a^{acti} and U_d^{acti} as inputs, we obtain one mixed strategy Nash Equilibrium for the game, as shown in Table 4.18.

Table 4. 18. Mixed strategy NE

Pure strategy: Defender	Probability	Pure strategy: Activist	Probability
2×1×1×2×1×1	0.6392	s_{e1}	0.3628
2×2×1×2×1×1	0.2249	s_{e4}	0.4555
2×2×2×2×1×1	0.1359	s_{e5}	0.1817

The Nash Equilibrium indicates that the defender should always set the security alert levels (SAL) at Zone 0 and Zone 1_1 as level 2 ("YELLOW") while at the Gate and Zone 2_1 as level 1 ("GREEN"). The SAL at the Main Entrance (MG) should be set as "GREEN" by a probability of 63.92% and as "YELLOW" by a probability of 22.49%+13.59% = 36.08%. The SAL at the Dock #1 should be set as "GREEN" by a probability of 63.92%+22.49% = 86.41% and as "YELLOW" by a probability of 13.59%. The activist would attack target T2 by a probability of 36.28% and would attack target T4 by a probability of 63.72% by passing perimeter 1 through the MG and through the Dock #1 by probabilities of 45.55% and of 18.17% respectively.

Table 4. 19. Probability that the attacker can reach the target sucesfully

Acti's Stg Def's Stg	s_{e1}	s_{e4}	s_{e5}
2×1×1×2×1×1	$P: 0.646, \tilde{P}: 0.646$	$P: 0.1054, \tilde{P}: 0.1054$	$P: 0.0984, \tilde{P}: 0.0984$
2×2×1×2×1×1	$P: 0.646, \tilde{P}: 0.646$	$P: 0.0685, \tilde{P}: 0.0685$	$P: 0.0984, \tilde{P}: 0.0984$
2×2×2×2×1×1	$P: 0.646, \tilde{P}: 0.646$	$P: 0.0685, \tilde{P}: 0.0685$	$P: 0.0522, \tilde{P}: 0.0522$

Table 4.19 further illustrates the probabilities that the activist would successfully reach the target, from the defender's and the attacker's point of view respectively. For instance, if the activist plays s_{e5}, then he has to pass Zone 0, Dock #1, and Zone 1_1, further if the defender plays a pure strategy $s_d = 2 \times 2 \times 2 \times 2 \times 1 \times 1$, then the SAL at Zone 0, the Dock #1, and Zone 1_1 are all "YELLOW". In this case, the probability that the activist will successfully reach target T4 can be calculated as $P = 0.95 * 0.68 * 0.28 * 0.53 * 0.8 * 0.68 = 0.0522$ (numbers are given in Table 4.12), resulting in the result shown in the bottom right cell in Table 4.19.

Table 4.20 further demonstrates defence costs and conditional expected losses and gains for each of the pure strategies that are played in the equilibrium. Note that the defence cost only depends on the defender's pure strategies while the conditional expected losses and gains only depend on the attacker's target. Hence, they can be given separately.

Table 4. 20. Defence cost and conditional expected losses and gains

Defender pure strategy	Defence cost (k€)	Activist Pure strategy	Conditional Expected Loss and Gain (k€)
2×1×1×2×1×1	170	s_{e1}	$PL: 140, \widetilde{PL}: 77$
2×2×1×2×1×1	180	s_{e4}	$PL: 595, \widetilde{PL}: 540$
2×2×2×2×1×1	185	s_{e5}	$PL: 595, \widetilde{PL}: 540$

According to the results in Tables 4.18 to 4.20, the defender obtains an equilibrium payoff of approximately [1] -242.0 k€ while the activist's equilibrium payoff equals 48.7 k€, as calculated by Formulas (7.1) and (7.2).

$$u_d^* = [0.6392, 0.2249, 0.1359] \times u_{d\_temp} \times \begin{bmatrix} 0.3628 \\ 0.4555 \\ 0.1817 \end{bmatrix} \approx -242.0 \cdots\cdots\cdots\cdots\cdots\cdots \text{(7.1)}$$

[1] In this Chapter, all the payoff values are rounded to the nearest tenth.

$$u_a^* = [0.6392, 0.2249, 0.1359] \times u_{a\_temp} \times \begin{bmatrix} 0.3628 \\ 0.4555 \\ 0.1817 \end{bmatrix} \approx 48.7 \cdots\cdots\cdots\cdots\cdots\cdots (7.2)$$

$$u_{d\_temp} = -\begin{bmatrix} 0.646 & 0.1054 & 0.0984 \\ 0.646 & 0.0685 & 0.0984 \\ 0.646 & 0.0685 & 0.0522 \end{bmatrix} .\times \begin{bmatrix} 140 & 595 & 595 \\ 140 & 595 & 595 \\ 140 & 595 & 595 \end{bmatrix} - \begin{bmatrix} 170 & 170 & 170 \\ 180 & 180 & 180 \\ 185 & 185 & 185 \end{bmatrix}$$

$$u_{a\_temp} = \begin{bmatrix} 0.646 & 0.1054 & 0.0984 \\ 0.646 & 0.0685 & 0.0984 \\ 0.646 & 0.0685 & 0.0522 \end{bmatrix} .\times \begin{bmatrix} 77 & 540 & 540 \\ 77 & 540 & 540 \\ 77 & 540 & 540 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Stackelberg equilibrium

The pure strategy Stackelberg Equilibrium of the game is that:$\bar{s}_d = 2 \times 1 \times 1 \times 2 \times 1 \times 1$ and $\bar{s}_a = s_{e4}$. Figure 4.6 shows the defender's payoffs and the activist's best responses to each of the defender's pure strategies. The x axis denotes the defender's pure strategies, and concerns in total a number of $n = 729$ points. The right-hand side y axis together with the 'o' (and brown) points represent the activist's best responses to the defender's pure strategies. The left-hand side y axis together with the '+' (and blue) points denote the defender's payoffs in case that the defender plays a pure strategy and the activist plays a best response. The square (and red) point on top of the figure denotes the highest payoff that the defender can obtain by playing pure strategies in a Stackelberg game. The square (and red) point's corresponding defender pure strategy (on the x axis) and activist's best response (on the right-hand side y axis) are the defender and the activist's strategies of a pure strategy Stackelberg Equilibrium. The corresponding strategies are $\bar{s}_d = 2 \times 1 \times 1 \times 2 \times 1 \times 1$ and $\bar{s}_a = s_{e4}$, and the corresponding defender's Stackelberg Equilibrium payoff is -232.7 k€.

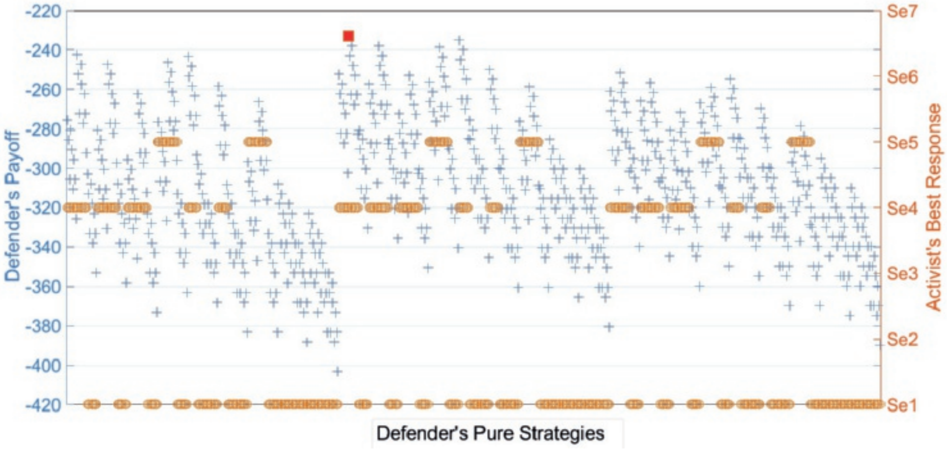


Figure 4. 6. Defender's payoff by responding with different strategies

Table 4.21 shows the mixed strategy Stackelberg Equilibrium for the game, obtained by employing the MultiLP algorithm [(12)] and taking U_a^{acti} and U_d^{acti} as input information. It is interesting to notice that the defender's Stackelberg Equilibrium strategy is identical to her Nash Equilibrium strategy (shown in Table 4.18). The fact that these equilibria are the same stems from the observation that when the defender plays her Nash (Stackelberg) Equilibrium strategy, the activist has the same payoffs by responding with strategies s_{e1}, s_{e4}, and s_{e5}, as shown in the second column of Table 4.22. Moreover, this payoff is higher than the payoffs obtained if responding with other strategies. Furthermore, the third column of Table 4.22 shows the defender's payoff when the activist responds different strategies. It reveals that though being indifferent among strategies s_{e1}, s_{e4}, and s_{e5}, the activist plays a strategy that benefits the defender, otherwise if the activist plays s_{e1}, the defender's payoff would significantly decrease to -264.7 k€. The activist (the game follower)'s behaviour of choosing a favour strategy for the defender (the game leader) is called the "breaking-tie" assumption, as we discussed in section 2.2.2.3.

Table 4. 21. SSE strategies of the game

Defender Pure Strategy	Probability
2×1×1×2×1×1	0.6392
2×2×1×2×1×1	0.2249
2×2×2×2×1×1	0.1359
Activist's pure strategy	**Probability**
s_{e4}	1

Table 4.22 further indicates that the defender's expected payoff resulting from the mixed strategy Stackelberg Equilibrium is -229.1 k€ and the activist's payoff equals 48.7 k€.

Table 4. 22. Players' payoffs when the activist play different strategies and the defender plays her SE strategy

Activist's strategy	Activist's payoff	Defender's payoff
s_{e1}	**48.7420**	**-264.7271**
s_{e2}	12.8172	-193.6312
s_{e3}	12.8172	-193.6312
s_{e4}	**48.7420**	**-229.0954**
s_{e5}	**48.7420**	**-229.0954**
s_{e6}	28.0991	-188.8367
s_{e7}	28.0991	-188.8367

4.6.3.2 Multiple attacker types: the Terrorist and the Activist

Results of the game involving multiple types of attackers, namely, the terrorist and the activist, are given. The Bayesian Nash Equilibrium and the Bayesian Stackelberg Equilibrium are calculated.

Bayesian Nash equilibrium

Table 4.23 shows the defender's and the attackers' Bayesian Nash Equilibrium (BNE) strategy. Table 4.24 shows the attackers' payoffs by responding with different pure strategies to the defender's BNE strategy. The best responses are underlined and put in bold. Recalling the attackers' BNE strategies given in Table 4.23, it is shown that both attackers are playing their best response strategies in their BNE. The terrorist is playing s_{v5} and the activist is playing s_{e1} and s_{e5}.

Table 4. 23. BNE strategies

Index	Defender Pure Strategy	Probability
$s_{d-BNE-1}$	2×2×1×2×1×1	0.8641
$s_{d-BNE-2}$	2×2×2×2×1×1	0.1359
Terrorist pure strategy		**Probability**
s_{v5}		1
Activist pure strategy		**Probability**
s_{e1}		0.6820
s_{e5}		0.3180

Table 4. 24. Attackers' payoff by responding with different strategies to the defender's BNE strategy

Terrorist Pure Strategy	Payoff	Activist pure strategy	Payoff
s_{v1}	67.5200	s_{e1}	**48.7420**
s_{v2}	65.5820	s_{e2}	9.2792
s_{v3}	1.9923	s_{e3}	12.8172
s_{v4}	27.0049	s_{e4}	36.0049
s_{v5}	**128.7686**	s_{e5}	**48.7420**
s_{v6}	79.2976	s_{e6}	20.6479
		s_{e7}	28.0991

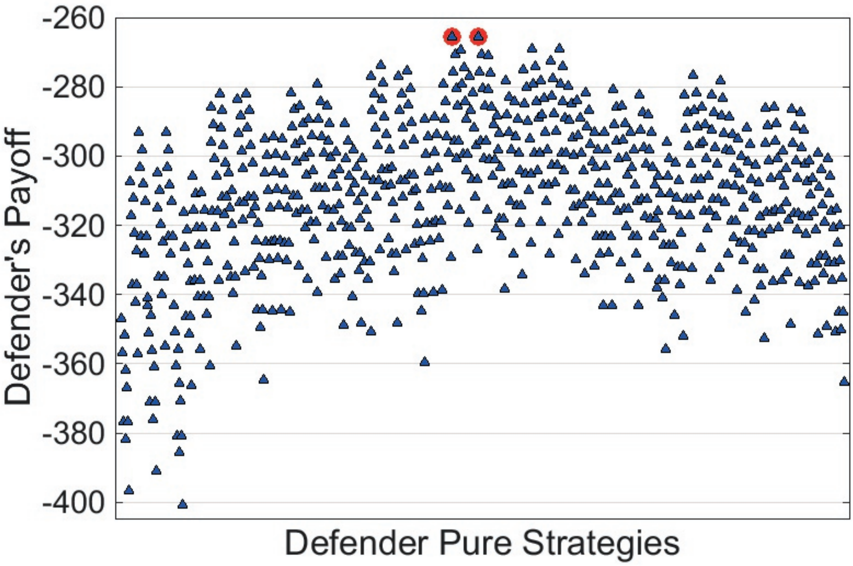


Figure 4. 7. Defender's payoffs by responding with pure strategies to the attackers' BNE strategies

Figure 4.7 illustrates the defender's payoff by responding with different pure strategies to the attackers' BNE strategy. The x axis denotes different defender pure strategies, while the blue triangles represent the defender's payoff (y axis) when she plays the corresponding pure strategy (x axis). Therefore, there are in total 729 triangles in the figure. The two triangles surrounded by red circles denote the best response strategies, and the corresponding defender pure strategies are 2×2×1×2×1×1 and 2×2×2×2×1×1 respectively, while the corresponding defender's payoff is -265.5 k€. Recalling the defender's BNE strategy given in Table 4.23, it is shown that the defender is also playing her best response strategies.

In conclusion, in the BNE, the defender and the attackers are all playing their best response strategies to their opponents' strategies.

Bayesian Stackelberg equilibrium

The defender's Bayesian Stackelberg Equilibrium strategy we obtained for the game is the same as her Bayesian Nash Equilibrium strategy as shown in Table 4.23. The defender's expected payoff from the BSE is -251.6 k€, being higher than her expected payoff from the BNE, which is -265.5 k€.

Table 4. 25. Slightly modified BSE strategies

Defender's BSE strategy			
Index	Defender Pure Strategy		Probability
$s_{d-BSE-1}$	2×2×1×2×1×1		0.8681
$s_{d-BSE-2}$	2×2×2×2×1×1		0.1319
Terrorist's BSE strategy		**Activist's BSE strategy**	
Index	Probability	Index	Probability
s_{v5}	1	s_{e5}	1

By playing the same strategy, the defender receives different expected payoffs from the BNE and the BSE. This is the result of the "breaking-tie" assumption (see section 2.2.2.3 for more information). As shown in Table 4.23, the activist has the same corresponding payoff (that is, 48.7 k€) by both responding strategies s_{e1} or s_{e5}. On the contrary, the defender's payoff would be -271.1 k€ and -235.5 k€ respectively, if the activist plays s_{e1} or s_{e5}. In the BNE solution, the activist is assumed not to be able to know the defender's strategy when he moves. Therefore, the activist randomly (but strategically) plays both strategy s_{e1} and s_{e5}. Conversely, in the BSE solution, the activist knows the defender's strategy when he moves. Therefore, the activist knows that both strategies s_{e1} and s_{e5} will bring himself the best payoff and the "breaking-tie" assumption requires the activist to choose s_{e1} or s_{e5} preferable for the defender, resulting that strategy s_{e5} is chosen as the activist's best response.

The MovLib algorithm is applied on the BSE to relax the "breaking-tie" assumption, by slightly modifying the players' strategies, as discussed in section 4.4.4. The ε is set as 0.1 for both the terrorist and the activist. The modified BSE is shown in Table 4.25. Table 4.26 further shows the attackers' payoffs by responding with different pure strategies to the defender's modified BSE strategy. It is shown that s_{v5} and s_{e5} are the unique best response to the terrorist and to the activist respectively. The defender's expected payoff from the modified BSE is only 0.0429 k€ less than the BSE payoff, being -251.7 k€,[2] while the "breaking-tie" assumption is no longer required.

[2] It is actually from -251.6466 k€ to -251.6895 k€.

Table 4. 26. Attackers' payoff by responding with different strategies to the defender's modified BSE strategy

Terrorist Pure Strategy	Terrorist Payoff	Defender Payoff	Activist pure strategy	Activist Payoff	Defender Payoff
s_{v1}	67.5200	-245.2595	s_{e1}	48.7420	-271.0995
s_{v2}	65.5820	-238.7995	s_{e2}	9.2792	-195.0503
s_{v3}	1.9923	-195.0503	s_{e3}	12.8450	-200.0422
s_{v4}	27.0049	-213.5528	s_{e4}	36.0049	-221.4335
s_{v5}	**128.7686**	-273.1719	s_{e5}	**48.8420**	-235.5772
s_{v6}	79.2976	-359.2546	s_{e6}	20.6479	-191.4834
			s_{e7}	28.1576	-195.2381

4.7 Conclusions

In this chapter, the most basic form of the chemical plant protection (CPP) game is proposed. The CPP game as put forward and explained in this chapter assumes complete information and rational players. Several typical game solutions are defined for the game, and the user of the game may decide which solution to use according to his/her belief of the threat. If only one threat is possible, and the threat is not able to know the defender's defence plan, then an NE should be used; if only one threat is possible, and the threat knows the defender's plan, then an SE should be used; if multiple threats exist, a Bayesian simultaneous CPP game or a Bayesian Stackelberg CPP game should be used, in case of that the attacker does not know and knows the defender's plant, respectively.

Inputs and outputs of the game are also discussed. The CPP game needs a lot of quantitative input data, and conventional security risk analysis approaches (e.g., the API SRA) are able to provide this required information. The output of the game can also be mapped to concepts used in conventional security risk analysis methodologies.

Drawbacks of the basic CPP game are obvious. Firstly, only the uncertainty of different types of attackers are modelled, while the uncertainties of the attacker's parameters/information and the uncertainties of the attacker's rationalities are not considered. Chapter 5 will address these 2 types of uncertainties respectively. Secondly, the CPP game works only for intrusion attacks. For remote attacks (i.e., through a network attack on the plant's control system) the CPP game cannot be employed since such attacks have totally different characteristics compared to intrusions. Hence, for cyber security new models are needed. Also, the exit procedure of an intrusion attack is not considered, which in case of a thief threat can be quite important. Thirdly, industrial security practice is more complicated than our theoretical description. For example, the mixed strategy for the defender is explained as the defender changing her security alert level day to day. However, in practice, the defender may not be able to change the SAL because of some location-fixed equipment (e.g., camera system).

References

[1] Cox Jr LAT. Game theory and risk analysis. Risk Anal. 2009;29(8):1062-8.

[2] Skaperdas S. Contest success functions. Economic theory. 1996;7(2):283-90.

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

[4] Brown G, Carlyle M, Salmerón J, Wood K. Defending critical infrastructure. Interfaces. 2006;36(6):530-44.

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

[6] Talarico L, Reniers G, Sörensen K, Springael J. MISTRAL: A game-theoretical model to allocate security measures in a multi-modal chemical transportation network with adaptive adversaries. Reliab Eng Syst Saf. 2015;138:105-14.

[7] Moulin H, Vial J-P. Strategically zero-sum games: the class of games whose completely mixed equilibria cannot be improved upon. International Journal of Game Theory. 1978;7(3-4):201-21.

[8] Lemke CE, Howson J, Joseph T. Equilibrium points of bimatrix games. Journal of the Society for Industrial and Applied Mathematics. 1964;12(2):413-23.

[9] Rao NS, Poole SW, Ma CY, He F, Zhuang J, Yau DK. Defense of Cyber Infrastructures Against Cyber-Physical Attacks Using Game-Theoretic Models. Risk Anal. 2015.

[10] Zhuang J, Bier VM. Balancing terrorism and natural disasters-defensive strategy with endogenous attacker effort. Operations Research. 2007;55(5):976-91.

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

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

[13] Shoham Y, Leyton-Brown K. Multiagent systems: Algorithmic, game-theoretic, and logical foundations: Cambridge University Press; 2008.

[14] Ceppi S, Gatti N, Basilico N, editors. Computing Bayes-Nash equilibria through support enumeration methods in Bayesian two-player strategic-form games. Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 02; 2009: IEEE Computer Society.

[15] Paruchuri P, Pearce JP, Marecki J, Tambe M, Ordonez F, Kraus S, editors. Playing games for security: an efficient exact algorithm for solving Bayesian Stackelberg games. Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2; 2008: International Foundation for Autonomous Agents and Multiagent Systems.

[16] IENM. Letter of 26 April 2013 to the Parliament. In: 2013/19920 MvIIB, editor. 2013.

[17] Reniers, Van Erp. Operational Safety Economics: A Practical Approach focused on the Chemical and Process Industries: John Wiley & Sons; 2016.

[18] Clark DJ, Riis C. Contest success functions: an extension. Economic Theory. 1998;11(1):201-4.

[19] Guan P, Zhuang J. Modeling Resources Allocation in Attacker-Defender Games with “Warm Up” CSF. Risk Anal. 2016;36(4):776-91.

[20] Lee Y, Kim J, Kim J, Kim J, Moon I. Development of a risk assessment program for chemical terrorism. Korean Journal of Chemical Engineering. 2010;27(2):399-408.

[21] Reniers G, Cozzani V. Domino Effects in the Process Industries: Modelling, Prevention and Managing: Elsevier B.V.; 2013. 1-372 p.

5

SINGLE PLANT PROTECTION: PLAYING CHEMICAL PLANT PROTECTION GAME WITH UNCERTAINTIES

A common criticism on game theoretic risk analysis of security threats is that it requires quantitative parameters of both the defender and the attacker, whereby the parameters of the attackers especially are difficult to estimate. Moreover, current literature on chemical plant protection games assume a 'rational' attacker. This chapter therefore extends the Chemical Plant Protection game to be able to deal with the defender's distribution-free uncertainties on the attacker's parameters (section 5.2) and to be able to take bounded rational attackers into account (section 5.3). Algorithms for solving these games with uncertainties are proposed. In section 5.4, a case study, which has the same basic settings as the case used in Chapter 4, is employed to illustrate the models and algorithms.

This chapter is based on the following papers:

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

Zhang, L., Reniers, G., Chen, Bin., & Qiu, X. (2018). A Chemical Plant Protection Game Incorporating Bounded Rational Attackers and Distribution-free Uncertainties. Submitted to Risk Analysis.

5.1 Introduction

Malicious attacks can be a serious threat to chemical facilities. However, in current literature, modelling intelligent adversaries is still in its infancy, and the existence of intelligent adversaries is one of the key challenges of security research.(1) Cox (2) suggested a combination of game theory and probabilistic risk analysis for adequately carrying out a security risk analysis. Some game theoretic research has been published for protecting chemical industries. Zhang and Reniers (3, 4) proposed a chemical plant protection (CPP) game based on the general intrusion detection approach in chemical plants. Feng et al. (5, 6) studied a resource allocation game among multiple chemical sites in a city. Reniers and his co-authors (7-9) conducted game theoretic research on stimulating security investments in chemical parks. Talarico et al. (10) proposed a game theoretic model named "MISTRAL" for protecting multi-modal chemical transportation networks.

However, all these mentioned game theoretic models for protecting chemical plants ask for quantitative inputs, such as the probabilities of intrusion, consequences of an attack etc. However, these quantitative inputs are difficult to obtain in industrial practice, making these models difficult to be used in realistic cases. Though the exact numerical inputs are difficult to obtain, the intervals of them are relatively easier to estimate. In this chapter, the previously proposed chemical plant protection (CPP) game (11, 12) is extended to deal with distribution-free/interval inputs. Distribution-free/interval input means that given a parameter, if denoted as σ, the defender neither knows the exactly number of σ, nor knows the distribution of σ, but she does know the maximal and minimal values of σ.

Moreover, current literature employing game theory for protecting chemical facilities assumes rational players in all games, that is, players aim at maximizing their benefits. However, human beings are not always rational players, and maybe especially terrorist attackers. There are multiple approaches on modelling bounded rational decision makers. The simplest one is the epsilon-optimal player model.(13) An epsilon-optimal player is a player who is not definitely playing his best choice strategy, but playing any strategies which have a close payoff (i.e., a difference less than a defined small constant number: epsilon) to the best choice strategy. The quantal response model assumes that a player would play any strategies, and the probability that a strategy is played should satisfy interiority, continuity, responsiveness, and monotonicity.(14) By far, the most common specification for calculating the quantal response probabilities is the logit form.(15) In the $level-k$ thinking model, a level 0 player is a player who plays randomly, while a level k $(k=1,2,\dots)$ player is a player who plays optimally assuming his opponent is a level $k-1$ player. The $level-k$ thinking model can be supported by a famous experiment named "the beauty contest game".(16) All these 3 approaches have been well studied in academia and applied in reality. However, a common drawback of them is their requirement on quantitative, user-defined parameters: in the epsilon-optimal model, a tolerance ε should be defined; in the logit form of the quantal response model, a parameter λ is needed; in the level-k thinking model, a k is required. These parameters are used to describe players' behaviour, and in case of an attacker player in the security game, these parameters are quite difficult to obtain. Jiang et al. (17) proposed a Monotonic Maximin solution for security games to include bounded rational attackers. In the Monotonic Maximin solution, attackers are assumed to play higher payoff strategies with higher probabilities, and no further constraints for the attacker's behaviours are needed.

5.2 Playing the Chemical Plant Protection game with distribution-free uncertainties

In this sub-section, the Chemical Plant Protection game is extended to deal with input parameters with distribution-free uncertainties [(18)]. The so-called interval CPP game is defined. Two algorithms, namely, the interval bi-matrix game solver (IBGS) and the interval CPP game solver (ICGS), are proposed.

5.2.1 Motivation

The defender plans her defence according to her guess about the attacker's behaviour, while the attacker also plans and implements his attack based on his information on the defender's defence. The CPP game models these intelligent interactions between the company defender and the potential attackers. Uncertainties on multiple types of attackers may also be modelled by using the Bayesian Nash/Stackelberg equilibrium. Inputs of the game can be obtained by using some conventional security risk assessment methods such as the API SRA, and the outputs of the game can be translated to the conventional security risk analysis terminologies.

However, in industrial practice, as well as in most conventional security risk assessment methods, it is quite difficult to obtain exact numbers (point values) for most parameters that the CPP game needs. For instance, in Table 4.5 (see section 4.5.1), the SRA team may only provide an interval of the consequences, such as a property loss from 1k euro to 2k euro. Furthermore, the team would not be able to know how the consequences are distributed on the interval, and whether the consequences are uniformly distributed between $[1000, 2000]$ euro, or whether they follow a triangle shape between this interval, among other possibilities. The estimation on vulnerabilities (i.e., intrusion probabilities and conditional probabilities of a successful attack) is even more difficult than the estimation of consequences.

There are three approaches to deal with interval inputs with distribution-free uncertainties. The first approach is to use a figure representing the interval, such as using the median or an average number. By using this approach, we obtain the inputs for the CPP game directly, and then solve the game. The result of this approach is not robust. The CPP game is finally solved by (Mixed Integer) Linear Programming (LP), no matter which equilibrium concept is employed. It is well-known that the optimal value of a Linear Programming problem is always situated on the boundary of the feasible area [(19)]. Thus a small error on the input parameters would make the LP result very bad [(20)], and in the CPP game, the "real exact" number (which we do not know) would always have an (at least small) error to the representative number that we use. The second approach is to add assumptions on the parameters' distribution on the interval, and then solve the game with continuous uncertainties. For instance, assume that the consequence is uniformly distributed between $[1000{,}2000]$. With these assumptions, several algorithms can be employed to solve the game, see for instance, a comprehensive investigation in Kiekinveld et al. [(21)]. An obvious drawback of this approach is the assumption on the distribution. Different distribution assumption may have different results and in practice it is difficult to decide which distributions to use. This approach is also quite computationally time-consuming. The third approach is by employing robust optimization techniques, and works directly on the distribution-free uncertainties, see for instance, Kiekintveld et al. [(22)] and Nikoofal and Zhuang [(23)]. In this approach, no extra assumption is needed. The defender knows the intervals that the attacker's parameters will be situated in, and she plays the game conservatively thinking that the attacker's parameters are located at the worst point (but still within the interval) for her. In this chapter, the third approach is employed and explained.

As discussed in section 4.4.2, a sequential CPP game may bring the defender a “First-Mover advantage”, and it does not have the Equilibrium choice problem. Moreover, a sequential CPP game also reflects the industrial reality better. Therefore, in this chapter, we limit our research to the sequential CPP game.

5.2.2 Interval CPP game definition

Recalling the payoff Formula (4.8) (see section 4.3.3) and the intrusion probability calculation Formula (4.1) (see section 4.2), the attacker’s payoff can be re-written as Formula (5.1).

$$u_a(s_a, s_d) = \prod_{i=0}^{I} \tilde{P}_i^z \cdot \prod_{j=1}^{I} \tilde{P}_j^p \cdot \tilde{P}_y(s_a) \cdot \tilde{L}_y(s_a) - C_a(s_a) \cdots\cdots\cdots\cdots (5.1)$$

For each attacker parameter (i.e., the $\tilde{P},\ \tilde{P}_y,\ \tilde{L}_y,\ C_a$), for the sake of convenience, denoting it as σ, assume that the defender does not know the exact value of σ and she knows that $\sigma \in [\sigma^{min}, \sigma^{max}]$. Since $\tilde{P}_i^z, \tilde{P}_j^p, \tilde{P}_y, \tilde{L}_y \geq 0$, thus we can easily know that:

$$u_a^{min} = \prod_{i=0}^{I} \tilde{P}_i^{z^{min}} \cdot \prod_{j=1}^{I} \tilde{P}_j^{p^{min}} \cdot \tilde{P}_y^{\ min} \cdot \tilde{L}_y^{\ min} - C_a^{\ max} \cdots\cdots\cdots\cdots (5.2)$$

$$u_a^{max} = \prod_{i=0}^{I} \tilde{P}_i^{z^{max}} \cdot \prod_{j=1}^{I} \tilde{P}_j^{p^{max}} \cdot \tilde{P}_y^{\ max} \cdot \tilde{L}_y^{\ max} - C_a^{\ min} \cdots\cdots\cdots\cdots (5.3)$$

In this research, the defender is assumed to know the exact numbers of her own parameters. Therefore, in the Interval CPP game, the defender’s payoff matrix is the same as in the CPP game, while the attacker’s payoff matrix consists of an upper bound matrix and a lower bound matrix, as defined in Formulas (5.2) and (5.3). We denote the Interval CPP game as $ICG = \{U_d, U_a^{min}, U_a^{max}\}$.

5.2.3 Interval Bi-Matrix Game Solver (IBGS)

In the Interval CPP game, if the defender commits to a mixed strategy $y \in Y$, she would not be able to work out the attacker’s best response, due to the existence of uncertainties. Contrary to Formula (4.20) (see section 4.4.2), the defender in an interval game only knows an interval of the attacker’s payoffs related to responding to a pure strategy, denoted as $u_a^i \in \left[u_a^{i-min}, u_a^{i-max}\right]$, and we have $u_a^{i-min} = U_a^{min}(i,:) \cdot y$ and $u_a^{i-max} = U_a^{max}(i,:) \cdot y$.

Knowing the range of the attacker’s payoffs, the defender can work out the attacker’s maximal lower bound of payoffs among all the attacker’s pure strategies, i.e., $R = \max_{i \in M} u_a^{i-min}$, in which $M = \{1,2,\ldots,m\}$ and $i \in M$ means for each attacker pure strategies (i.e., $s_a \in S_a$). The defender beliefs that a rational attacker would not play a strategy whose upper bound payoff is less than R. Formula (5.4) illustrates the reason for this judgement, in which ml is the strategy who has the maximal lower bound payoff, k is the strategy whose upper bound payoff is less than R. The formula shows that the attacker would always have a higher payoff by responding strategy ml instead of by responding strategy k, to the defender’s strategy y.

$$U_a(k,:) \cdot y \leq u_a^{k-max} < R = u_a^{ml-min} \leq U_a(ml,:) \cdot y \cdots\cdots\cdots\cdots (5.4)$$

Based on the above analysis, an algorithm for solving the Chemical Plant Protection game with distribution-free uncertainties is proposed, as shown in Formula (5.5).

$$\max_{q,h,\gamma,y,R} \sum_{t\in TL} p^t\gamma^t$$

$$s.t.\begin{cases} c1.\ \ 0 \le R^t - \underline{U}_a^t(i,:)\cdot y \le (1-h_i^t)\cdot\Gamma,\ \ \forall i\in M^t \\ c2.\ \ (q_i^t-1)\cdot\Gamma \le \overline{U}_a^t(i,:)\cdot y - R^t \le q_i^t\cdot\Gamma,\ \ \forall i\in M^t \\ c3.\ \ \Gamma\cdot(1-q_i^t)+U_d^t(i,:)\cdot y \ge \gamma^t,\ \ \forall i\in M^t \\ c4.\ \ q_i^t \ge h^t{}_i,\ \ \forall i\in M^t \\ c5.\ \ q_i^t, h_i^t \in \{0,1\} \\ c6.\ \ \sum h^t = 1 \\ c7.\ \ \sum y = 1, y_i\in[0,1] \\ c8.\ \ R^t, \gamma^t \in R \end{cases} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots(5.5)$$

In the algorithm, t, TL, p^t denote a threat, the threat list, and the prior probability of threat, respectively, as already defined in previous chapters; R^t is the maximal value of the lower bound payoffs for threat t; $\underline{U}_a^t$ and $\overline{U}_a^t$ are the lower bound and upper bound payoff matrices for threat t respectively; $M^t = \{1,2,\dots,m^t\}$ is the pure strategy index set; h_i^t and q_i^t are binary variables; U_d^t denotes the defender's payoff matrix in case of a threat t; Γ is a constant large real number.

To understand the algorithm, notice that in constraint $c1$, if $h_i^t = 1$, we obtain $R^t = \underline{U}_a^t(i,:)\cdot y$, otherwise if $h_i^t = 0$, we obtain $R^t \ge \underline{U}_a^t(i,:)\cdot y$. Thus the binary variable h_i^t represents that whether the i^{th} pure strategy of threat t has the maximal lower bound payoff, and if yes, $h_i^t = 1$, if no, $h_i^t = 0$. In constraint $c2$, if $q_i^t = 1$, we get $R^t \le \overline{U}_a^t(i,:)\cdot y$, otherwise if $q_i^t = 0$, we have $R^t \ge \overline{U}_a^t(i,:)\cdot y$. To this end, the binary variable q_i^t denotes that whether the i^{th} pure strategy of threat t has a higher upper bound payoff than R^t, and if yes, $q_i^t = 1$, if no, $q_i^t = 0$. Constraint $c3$ is activated only when $q_i^t = 1$, which means that the i^{th} strategy is a possible choice for the attacker t. $c3$ together with the cost function also indicate that among all the possible attacker strategies, the defender conservatively aims to optimize the worst case γ^t. Constraint $c4$ means that the strategy who has the highest lower bound payoff must be a possible strategy for the attacker. To understand $c4$, one must notice that i) in $c2$, if $R^t = \overline{U}_a^t(i,:)\cdot y$, which means that the attacker's upper bound payoff by playing strategy i exactly equals R^t, then q_i^t can be either 0 or 1; ii) if q_i^t can be either 0 or 1, then in some cases, constraint $c3$ and the cost function together will lead to a result that $q_i^t = 0$; iii) in some special situations, for instance, in a situation that the interval radius equals 0, which means that $\underline{U}_a^t = \overline{U}_a^t$, constraint $c4$ is needed to make sure that the strategy which will bring the attacker the highest lower bound payoff will be a possible best response strategy.

We may notice that Formula (5.5) does not use any special characteristics of the CPP game. Indeed, it is applicable to any bi-matrix games with distribution-free uncertainties. Therefore, we name it as Interval Bi-Matrix Game Solver (IBGS). It is also worth noting that this algorithm can be used for solving games with bounded rational attackers who are $\epsilon - optimal$ players, more details will be given in section 5.3.2 of this book. The IBGS is mainly derived from the BRASS algorithm developed by Pita et al. (13).

5.2.4 Parameter coupling

The IBGS, being general enough, loses some model properties of the CPP game. Recalling Formulas (5.2) and (5.3), the uncertainties on the parameters result in uncertainties on the payoffs. However, different attacker strategies may share some parameters, and the shared parameters cannot reach their maximal values in one strategy while reaching their minimal values in another.

An illustrative example can be useful to make the above statement clear. Assume that attacker strategies s_{a1} and s_{a2} aim to attack different targets with the same attack scenario, and both targets are situated in zone level 0 (i.e., outside of the plant, for simplicity reasons). In this case, the attacker's payoff can be simplified as $u_a(s_a, s_d) = \tilde{P}_0^z(s_a, s_d) \cdot \widetilde{P}_y(s_a) \cdot \tilde{L}_y(s_a) - C_a(s_a)$. Further assume that the defender plays a pure strategy s_d resulting in the parameters as shown in Table 5.1. The only difference between these two strategies is the target, thus we can see that in Table 5.1, the $\tilde{P}_0^z(s_{a1}, s_d)$ and $\tilde{P}_0^z(s_{a2}, s_d)$, the $C_a(s_{a1})$ and $C_a(s_{a2})$, have the same range. The target related parameters $\widetilde{P}_y$ and $\tilde{L}_y$, on the other hand, have different ranges.

Table 5. 1. Illustrative parameters

Strategy s_{a1}			Strategy s_{a2}		
Para	min	max	Para	min	max
$\tilde{P}_0^z$	0.8	0.9	$\tilde{P}_0^z$	0.8	0.9
C_a	10	12	C_a	10	12
$\tilde{P}_y$	0.9	0.92	$\tilde{P}_y$	0.8	0.84
$\tilde{L}_y$	100	102	$\tilde{L}_y$	130	140

Based on Formulas (5.2) and (5.3), we have:

$$u_a^{min}(s_{a1}, s_d) = 0.8 \cdot 100 \cdot 0.9 - 12 = 60 \qquad (5.6)$$

$$u_a^{max}(s_{a1}, s_d) = 0.9 \cdot 102 \cdot 0.92 - 10 = 74.456 \qquad (5.7)$$

$$u_a^{min}(s_{a2}, s_d) = 0.8 \cdot 130 \cdot 0.8 - 12 = 71.2 \qquad (5.8)$$

$$u_a^{max}(s_{a2}, s_d) = 0.9 \cdot 140 \cdot 0.84 - 10 = 95.84 \qquad (5.9)$$

According to algorithm IBGS, we have $u_a^{min}(s_{a2}, s_d) = R \leq u_a^{max}(s_{a1}, s_d)$. Therefore, the defender may not be able to know whether the attacker would play strategy s_{a1} or not. However, strategies s_{a1} and s_{a2} have the same parameters $\tilde{P}_0^z$ and C_a. We substitute other parameters (i.e., $\widetilde{P}_y$ and $\tilde{L}_y$) into Formulas (5.2) and (5.3), and remain the shared parameters (i.e., $\tilde{P}_0^z$ and C_a), we obtain:

$$u_a^{min}(s_{a1}, s_d) = 90 \cdot \tilde{P}_0^z - C_a \qquad (5.10)$$

$$u_a^{max}(s_{a1}, s_d) = 93.84 \cdot \tilde{P}_0^z - C_a \qquad (5.11)$$

$$u_a^{min}(s_{a2}, s_d) = 104 \cdot \tilde{P}_0^z - C_a \qquad (5.12)$$

$$u_a^{max}(s_{a2}, s_d) = 117.6 \cdot \tilde{P}_0^z - C_a \qquad (5.13)$$

Although the defender only knows the intervals where the parameters $\tilde{P}_0^z$ and C_a are located in, she knows that $93.84 \cdot \tilde{P}_0^z - C_a = u_a^{max}(s_{a1}, s_d) \leq u_a^{min}(s_{a2}, s_d) = 104 \cdot \tilde{P}_0^z - C_a$. Based on this information, the defender can conclude that, for the attacker, strategy s_{a2} is always a better strategy than strategy s_{a1}. However, algorithm IBGS cannot support the defender to draw this conclusion. The reason is that, in the IBGS, when calculating the attacker's lower bound payoff by playing strategy $\mathrm{s_{a2}}$ (see Formula (5.8)), $\widetilde{\mathrm{P}}_0^{\mathrm{z}}$ is substituted by 0.8 and $\mathrm{C_a}$ is substituted by 12 (see Table 5.1, illustrative figure), while when calculating the attacker's upper bound payoff by playing strategy $\mathrm{s_{a1}}$ (see Formula (5.7)), $\widetilde{\mathrm{P}}_0^{\mathrm{z}}$ is substituted by 0.9 and $\mathrm{C_a}$ is substituted by 10 (see Table 5.1, illustrative figure). However, strategies $\mathrm{s_{a1}}$ and $\mathrm{s_{a2}}$ use the same attack scenario and their paths in zone 0 are the same. Therefore, the attack cost (i.e., $\mathrm{C_a}$) of these two strategies, and the probabilities of being detected in zone 0 of these two strategies (i.e., $\widetilde{\mathrm{P}}_0^{\mathrm{z}}$), should be the same, although the defender does not know the exact numbers of these parameters. Hence, when calculating the lower bound payoff, the lower bound values of the shared parameters (i.e., $\widetilde{\mathrm{P}}_0^{\mathrm{z}}$ and $\mathrm{C_a}$) are used, while when calculating the upper bound payoff, the upper bound values are used, and the fact is that the shared parameters should not reach their lower bound in one strategy and reach their upper bound in another strategy.

To formulate the parameters coupling problem as illustrated above, the so-called attacker payoff differences of two attacker pure strategies should be defined, as shown in Formula (5.14).

$$\Delta_{kl} = U_a(k,:) \cdot y - U_a(l,:) \cdot y, \quad \forall k,l \in M \cdots\cdots\cdots\cdots (5.14)$$

Further define Tp_k as the set of sub-zones and entrances that attacker pure strategy k would pass, and define Tp_l analogously. Define $Tp_{k\cdot l} = Tp_k \cap Tp_l$, denoting zones or entrances that the two strategies both pass by. For instance, if the attacker would follow an intrusion path as p1 in Figure 4.1 (see section 4.2), then we have:

$Tp_1 = \{zone0, truck\ entrance, zone\ 1\_1, entrance\ 3, zone\ 2\_2\}$,

and if the p2 in Figure 4.1 would be followed (only for illustrating Tp purpose), we obtain:

$Tp_2 = \{zone0, main\ entrance, zone\ 1\_1, entrance\ 1, zone\ 2\_1, entrance\ 2\}$.

And,

$Tp_{1\cdot 2} = \{zone\ 0, zone\ 1\_1\}$.

Substitute payoff Formula (5.1) into Formula (5.14), resulting in:

$$\Delta_{kl} = \sum_{j\in N}\left(\prod_{i\in Tp_k} \tilde{p}_{ij} \cdot \widetilde{PL}_k - C_k\right) \cdot y_j - \sum_{j\in N}\left(\prod_{i\in Tp_l} \tilde{p}_{ij} \cdot \widetilde{PL}_l - C_t\right) \cdot y_j \cdots\cdots\cdots (5.15)$$

In which $N = \{1,2,\dots,n\}$ denotes the defender's pure strategy index space; $\tilde{p}_{ij}$ denotes the success probability of passing an entrance i or a sub-zone i (which must be on the attacker's intrusion path) if the defender plays a pure strategy j; $\widetilde{PL}_k$ represents the conditional expected loss in case the attacker already arrived the target, i.e., $\widetilde{PL}_k = \tilde{P}_k \cdot \tilde{L}_k$; C_k is the cost of attacker strategy k; y_j is the probability that the defender plays pure strategy j.

Formula (5.15) can be further organized as:

$$\Delta_{kl} = \sum_{j\in N}\left[\left(\prod_{i\in Tp_{k\cdot l}} \tilde{p}_{ij}\right) \cdot \left(\prod_{i\in Tp_k - Tp_{k\cdot l}} \tilde{p}_{ij} \cdot \widetilde{PL}_k - \prod_{i\in Tp_l - Tp_{k\cdot l}} \tilde{p}_{ij} \cdot \widetilde{PL}_l\right)\right] \cdot y_j + C_l - C_k \cdots\cdots\cdots (5.16)$$

Which separates the shared parameters (i.e., $i \in Tp_{kl}$), and un-shared parameters (i.e., $i \in Tp_k - Tp_{k\cdot l}$ and $i \in Tp_l - Tp_{k\cdot l}$).

We hereafter determine and analyse Δ_{kl} from four cases, depending on whether strategy k and l use the same attack scenario (i.e., $e_k = e_l$) and whether they attacks the same target (i.e., $target_k = target_l$).

Case 1: $\boldsymbol{e_k = e_l, target_k = target_l}$.

The two strategies have the same attack scenario in case 1, which implies that their attack cost should be the same, i.e., $C_l = C_k$. Moreover, the attack targets are the same. Thus we have also the same conditional expected loss, i.e., $\widetilde{PL}_k = \widetilde{PL}_l$. Taking these analysis observations into consideration, Formula (5.16) can be simplified as:

$$\Delta_{kl}= \sum_{j\in N}[(\prod_{i\in Tp_{kl}} \tilde{p}_{ij}) \cdot \widetilde{PL}_k \cdot (\prod_{i\in Tp_k - Tp_{kl}} \tilde{p}_{ij} - \prod_{i\in Tp_l - Tp_{kl}} \tilde{p}_{ij})] \cdot y_j \cdots\cdots\cdots(5.17)$$

Recalling that only the unshared parameters are independent, which means that they can be equal to their maximal value in one strategy and be equal to their minimal value in another, we have:

$$\Delta_{kl}\geq \sum_{j\in N}\left[\left(\prod_{i\in Tp_{kl}} \tilde{p}_{ij}\right) \cdot \widetilde{PL}_k \cdot \left(\prod_{i\in Tp_k - Tp_{kl}} \tilde{p}_{ij}{}^{min} - \prod_{i\in Tp_l - Tp_{kl}} \tilde{p}_{ij}{}^{max}\right)\right] \cdot y_j \cdot\cdot(5.18)$$

Define $\xi_{klj}^{min} = \left(\prod_{i\in Tp_k - Tp_{kl}} \tilde{p}_{ij}{}^{min} - \prod_{i\in Tp_l - Tp_{kl}} \tilde{p}_{ij}{}^{max}\right)$. For $\forall j \in N$ in Formula (5.18), inequality (5.19) would hold if $\xi_{klj}^{min} \leq 0$, and vice versa.

$$\left[\left(\prod_{i\in Tp_{kl}} \tilde{p}_{ij}\right) \cdot \widetilde{PL}_k \cdot \xi_{klj}^{min}\right] \cdot y_j \geq \left[\left(\prod_{i\in Tp_{kl}} \tilde{p}_{ij}^{max}\right) \cdot \widetilde{PL}_k^{max} \cdot \xi_{klj}^{min}\right] \cdot y_j \cdots\cdots\cdots(5.19)$$

Noting that Formula (5.18) is a polynomial, thus we may derive:

$$\Delta_{kl}\geq \sum_{j\in N}\left[\left(\prod_{i\in Tp_{kl}} \tilde{p}_{ij}^{\varphi}\right) \cdot \widetilde{PL}_k^{\varphi} \cdot \xi_{klj}^{min}\right] \cdot y_j = \Delta_{kl}^{min} \cdots\cdots\cdots(5.20)$$

In which:

$$\varphi = \begin{cases} max, & if\ \xi_{klj}^{min} \leq 0 \\ min, & otherwise \end{cases}$$

Analogously, we have:

$$\Delta_{kl}^{max}= \sum_{j\in N}\left[\left(\prod_{i\in Tp_{kl}} \tilde{p}_{ij}^{\varphi}\right) \cdot \widetilde{PL}_k^{\varphi} \cdot \xi_{klj}^{max}\right] \cdot y_j \cdots\cdots\cdots(5.21)$$

In which:

$$\xi_{ktj}^{max} = \left(\prod_{i\in Tp_k - Tp_{lk}} \tilde{p}_{ij}{}^{max} - \prod_{i\in Tp_l - Tp_{lk}} \tilde{p}_{ij}{}^{min}\right)$$

$$\varphi = \begin{cases} max, & if\ \xi_{klj}^{max} \geq 0 \\ min, & otherwise \end{cases}$$

Case 2: $\boldsymbol{e_k = e_l, target_k \neq target_l}$.

In this second case, the two strategies have the same attack scenario, hence their attack costs are identical, i.e., $C_l = C_k$. The attack targets are however different. Thus we have different conditional expected losses, i.e., $\widetilde{PL}_k \neq \widetilde{PL}_l$. In this case 2, Formula (5.16) can thus be reformulated as:

$$\Delta_{kl}= \sum_{j\in N}[(\prod_{i\in Tp_{kl}} \tilde{p}_{ij}) \cdot (\prod_{i\in Tp_k - Tp_{kl}} \tilde{p}_{ij} \cdot \widetilde{PL}_k - \prod_{i\in Tp_l - Tp_{kl}} \tilde{p}_{ij} \cdot \widetilde{PL}_l)] \cdot y_j \cdots\cdots(5.22)$$

Recalling that only the unshared parameters are independent, which means that they can be equal to their maximal value in one strategy and be equal to their minimal value in another, we have:

$$\Delta_{kl}\geq \sum_{j\in N}\left[\left(\prod_{i\in Tp_{kl}} \tilde{p}_{ij}\right) \cdot \left(\prod_{i\in Tp_k - Tp_{kl}} \tilde{p}_{ij}{}^{min} \cdot \widetilde{PL}_k^{min} - \prod_{i\in Tp_l - Tp_{kl}} \tilde{p}_{ij}{}^{max} \cdot \widetilde{PL}_l^{max}\right)\right] \cdot y_j \cdots\cdots\cdots(5.23)$$

Define $\xi_{klj}^{min} = \left(\prod_{i\in Tp_k - Tp_{kl}} \tilde{p}_{ij}{}^{min} \cdot \widetilde{PL}_k^{min} - \prod_{i\in Tp_l - Tp_{kl}} \tilde{p}_{ij}{}^{max} \cdot \widetilde{PL}_l^{max}\right)$. For $\forall j \in N$ in Formula (5.23), inequality (5.24) would hold if $\xi_{klj}^{min} \leq 0$, and vice versa.

$$\left[\left(\prod_{i\in Tp_{kl}} \tilde{p}_{ij}\right) \cdot \xi_{klj}^{min}\right] \cdot y_j \geq \left[\left(\prod_{i\in Tp_{kl}} \tilde{p}_{ij}^{max}\right) \cdot \xi_{klj}^{min}\right] \cdot y_j \cdots\cdots\cdots(5.24)$$

Noting that Formula (5.23) is a polynomial, we obtain:

$$\Delta_{kl} \geq \sum_{j \in N}\left[\left(\prod_{i \in Tp_{kl}} \tilde{p}_{ij}^{\varphi}\right) \cdot \xi_{klj}^{min}\right] \cdot y_j = \Delta_{kl}^{min} \cdots\cdots \tag{5.25}$$

In which:

$$\varphi = \begin{cases} max, & if\ \xi_{klj}^{min} \leq 0 \\ min, & otherwise \end{cases}$$

Analogously, we have:

$$\Delta_{kl}^{max} = \sum_{j \in N}\left[\left(\prod_{i \in Tp_{kl}} \tilde{p}_{ij}^{\varphi}\right) \cdot \xi_{klj}^{max}\right] \cdot y_j \cdots\cdots \tag{5.26}$$

In which:

$$\xi_{klj}^{max} = \left(\prod_{i \in Tp_k - Tp_{lk}} \tilde{p}_{ij}{}^{max} \cdot \widetilde{PL}_k^{max} - \prod_{i \in Tp_l - Tp_{lk}} \tilde{p}_{ij}{}^{min} \cdot \widetilde{PL}_l^{min}\right)$$

$$\varphi = \begin{cases} max, & if\ \xi_{klj}^{max} \geq 0 \\ min, & otherwise \end{cases}$$

Case 3: $\boldsymbol{e_k \neq e_l, target_k = target_l}$, and Case 4: $\boldsymbol{e_k \neq e_l, target_k \neq target_l}$.

The third and fourth cases are analysed together. The attacker uses different attack scenarios, thus the cost related to the attack will be different, i.e., $C_l \neq C_k$. Furthermore, the conditional expected loss would also be different since the attack scenarios are different (even for the same target, different attack scenario may result in different consequences). Thus we have $\widetilde{PL}_k \neq \widetilde{PL}_l$. Analogously to the previous two cases, we directly discuss the result of case 3 and case 4 hereafter.

$$\Delta_{kl}^{min} = \sum_{j \in N}\left[\left(\prod_{i \in Tp_{kl}} \tilde{p}_{ij}^{\varphi}\right) \cdot \xi_{klj}^{min}\right] \cdot y_j + C_l^{min} - C_k^{max} \cdots\cdots \tag{5.27}$$

In which:

$$\xi_{klj}^{min} = \left(\prod_{i \in Tp_k - Tp_{kl}} \tilde{p}_{ij}{}^{min} \cdot \widetilde{PL}_k^{min} - \prod_{i \in Tp_l - Tp_{kl}} \tilde{p}_{ij}{}^{max} \cdot \widetilde{PL}_l^{max}\right)$$

$$\varphi = \begin{cases} max, & if\ \xi_{klj}^{min} \leq 0 \\ min, & otherwise \end{cases}$$

And,

$$\Delta_{kl}^{max} = \sum_{j \in N}\left[\left(\prod_{i \in Tp_{kl}} \tilde{p}_{ij}^{\varphi}\right) \cdot \xi_{klj}^{max}\right] \cdot y_j + C_l^{max} - C_k^{min} \cdots\cdots \tag{5.28}$$

In which:

$$\xi_{klj}^{max} = \left(\prod_{i \in Tp_k - Tp_{kl}} \tilde{p}_{ij}{}^{max} \cdot \widetilde{PL}_k^{max} - \prod_{i \in Tp_l - Tp_{kl}} \tilde{p}_{ij}{}^{min} \cdot \widetilde{PL}_l^{min}\right)$$

$$\varphi = \begin{cases} max, & if\ \xi_{klj}^{max} \geq 0 \\ min, & otherwise \end{cases}$$

Proposition 4.1. Δ_{kl}^{min} and Δ_{kl}^{max} are the maximal and the minimal bounded of Δ_{kl}.

Proposition 4.2. $\Delta_{kl}^{min} = -\Delta_{lk}^{max}$.

We can remark that this second proposition can be proved considering the fact that for each case of the above mentioned four cases, we have $\xi_{klj}^{min} = -\xi_{lkj}^{max}$. Hereafter, only the Δ_{kl}^{min} for each attacker strategy pairs will be investigated.

Proposition 4.3. If $\Delta_{kl}^{min} > 0$, strategy k is always a better response than l, in case of the defender's committed mixed strategy y.

We can remark here that this proposition can be straightforwardly proved since $0 < \Delta_{kl}^{min} \leq \Delta_{kl} = U_a(k,:) \cdot y - U_a(l,:) \cdot y$.

Aside from the $c2$ in the IBGS, proposition 4.3 shows a new criterium to calculate the attacker's possible best responses, that is, for any attacker pure strategy l, if there exists a strategy $k \in M$ that satisfies $\Delta_{kl}^{min} > 0$, then strategy l must not be the attacker's best response to the defender's strategy y.

Δ_{kl}^{min} is a linear polynomial of y. The coefficient of y_i can be calculated as long as the interval parameters are given. Firstly, decide which case (i.e., the above mentioned four cases) the pure strategy pair (k,l) belongs to; secondly, employ the corresponding Formulas to calculate the coefficient. For instance, in case 1, Formulas (5.19) and (5.20) must be used. Define a 3-dimension coefficient matrix $\Omega^t(M^t, M^t, N)$, whose unit $\Omega^t(k,l,:)$ is the coefficient vector of y in Δ_{kl}^{min}.

5.2.5 Interval CPP Game Solver (ICGS)

Taking into account the parameter coupling problem in the interval CPP game, the game would be played as:

1) The defender commits a mixed strategy $y \in Y$;
2) Each type of attacker t observes y, and plays his best response $BR^t \in M^t$ to y;
3) The defender also tries to work out each type of attacker's best response. However, due to the existence of the distribution-free uncertainties on the attacker's payoffs, she is not able to work out the BR^t;
4) Instead, the defender commits a $y \in \bar{Y} \subset Y$ which results that strategy k^t has the highest low bound payoff for the attacker type t;
5) For any other attacker strategies, i.e., $\forall l \in M^t$, if $\Delta_{k^t l}^{min} > 0$, then l would definitely not be the attacker's best response, while if $\Delta_{k^t l}^{min} \leq 0$, then l would possibly be the attacker's best response. Define $PBR^t = \{l \in M^t | \Delta_{k^t l}^{min} \leq 0\}$ which denotes the attacker's possible best response set;
6) The defender only knows that the attacker will play a strategy from the PBR^t, and she does not know which strategy exactly that the attacker would play;
7) The defender thus conservatively assume that the attacker would play a strategy from the PBR^t which would minimize the defender's payoff.
8) The defender plays her optimal action.

Formula (5.29) shows a Mixed Integer Linear Programming (MILP) based algorithm, for calculating the equilibrium for the Bayesian Stackelberg CPP game with distribution-free uncertainties. Being different to the IBGS algorithm, which is general for solving any bi-matrix games with distribution-free uncertainties, the algorithm shown in Formula (5.29) considers the parameter coupling problem in the CPP game, and therefore it is only applicable for solving the CPP game with distribution-free uncertainties. For this reason, the algorithm shown in Formula (5.29) is named as Interval CPP Game Solver (ICGS).

In the ICGS algorithm as shown in Formula (5.29), Constraint $c9$ denotes that k^t is assumed to have the highest lower bound payoff for attacker type t (step 4) and this constraint limits the defender to play a subset of her mixed strategy (i.e., $\bar{Y} \in Y$ that satisfies constraint $c9$). Constraint $c10$ picks out the attacker's possible best response strategies. In constraint $c10$, if $\Omega^t(k^t,i,:) \cdot y > 0$, then $q_i^t = 0$; if $\Omega^t(k^t,i,:) \cdot y < 0$, then $q_i^t = 1$; otherwise q_i^t can be either 0 or 1. Comparing to step 5 in the above text, q_i^t therefore indicates whether strategy i belongs to the PBR^t (i.e., $q_i^t = 1$) or not (i.e., $q_i^t = 0$). Constraint $c11$ together with the cost function denotes that among all the attacker's possible best responses (i.e., PBR^t), the defender conservatively thinks that the attacker strategy being worst to the defender is the actual best response strategy for the attacker (step 7 and 8). In the cost function, the defender is maximizing γ while in constraint $c11$, γ is required to be less than any $U_d^t(i,:)\cdot y$ (which is the defender's payoff in case that the attacker plays strategy i as a response to the defender's committed strategy y) if $q_i^t = 1$ (which means that strategy i is in the attacker's possible best response strategy set). Notice that in constraint $c11$, if $q_i^t = 0$, then the constraint is not activated (Γ is a big constant number and therefore the left-hand side of the inequality is always greater than the right-hand side). Constraints $c12$ and $c13$ express that q and y are binary variables and a mixed strategy, respectively. $c12$ also mentions $q_{k^t}^t = 1$. The reason is that: (i) obviously we have that $\Omega^t(k^t,k^t,:)\cdot y = 0$; (ii) according to constraint $c10$, $q_{k^t}^t$ can be either 0 or 1, which means that strategy k^t can be either in the attacker's best response strategy set or not; (iii) however, k^t is the strategy that has the highest attacker lower bound payoff (i.e., $R^t = \underline{U}_a^t(k^t,:)\cdot y = \max_{i\in M^t}\{\underline{U}_a^t(i,:)\cdot y\}$), and the attacker's upper bound payoff by playing strategy k^t would be no less than the lower bound payoff (i.e., $\bar{U}_a^t(k^t,:)\cdot y \geq \underline{U}_a^t(k^t,:)\cdot y = R^t$), thus strategy k^t should be included in the PBR^t; (iv) we manually set $q_{k^t}^t = 1$.

$$
\max_{q,\gamma,y} \sum_{t\in TL} \rho^t \gamma^t
$$
$$
s.t.\begin{cases}
\text{c}9.\;\; \underline{U}_a^t(i,:)\cdot y \leq \underline{U}_a^t(k^t,:)\cdot y, \;\; \forall i\in M^t \\
\text{c}10.\;\; -q_i^t\cdot\Gamma \leq \Omega^t(k^t,i,:)\cdot y \leq (1-q_i^t)\cdot\Gamma, \;\; \forall i\in M^t \\
c11.\;\; \Gamma\cdot(1-q_i^t)+U_d^t(i,:)\cdot y \geq \gamma^t, \;\; \forall i\in M^t \\
c12.\;\; q_i^t \in\{0,1\}, q_{k^t}^t = 1 \\
c13.\;\; \sum y = 1, y_i \in[0,1]
\end{cases} \quad \cdots\cdots\cdots\cdots\cdots\cdots(5.29)
$$

The algorithm should be implemented to each combination of $k^t \in M^t$, obtaining the optimal payoff for the defender $H(k^1,k^2,\ldots,k^{|TL|})$ and the corresponding optimal strategy for the defender $\tilde{y}(k^1,k^2,\ldots,k^{|TL|})$. Finally, choose the maximal H and its corresponding $\tilde{y}$ as the defender's optimal payoff and optimal solution.

Proposition 4.4. Defender's equilibrium payoff from the ICGS is higher than or equal to her equilibrium payoff from the IBGS.

Proof: ∀y ∈ Y, without loss of generality, assume that $\underline{U}_a^t(\pi^t,:)\cdot y \geq \underline{U}_a^t(i,:)\cdot \text{y}$, for all $i \in M^t$.

In the IBGS, from $c1$ we have $R^t = \underline{U}_a^t(\pi^t,:)\cdot y$. From c2 we know that $\forall i \in M^t$, if $\overline{U}_a^t(i,:)\cdot y < R^t$, then $q_i^t = 0$, which means that the strategy i will definitely not be the attacker's best response. Define $E_B^t = \left\{i \in M^t - \{\pi^t\} \middle| \overline{U}_a^t(i,:)\cdot y < R^t\right\}$. According to $c3$, we have $\gamma_B^t = min_{i\in M^t - E_B^t}\{U_d^t(i,:)\cdot y\}$.

In the ICGS, from $c9$ we have $k^t = \pi^t$. From $c10$ we know that $\forall i \in M^t$, if $\Omega^t(k^t,i,:)\cdot y > 0$, then $q_i^t = 0$, which means that strategy k^t is always a better response than strategy i, or, strategy i will definitely not be the attacker's best response. Define

$E_C^t = \{i \in M^t - \{\pi^t\} | \Omega^t(k^t, i, :) \cdot y > 0\}$. According to $c11$, we have that $\gamma_C^t = min_{i \in M^t - E_C^t}\{U_a^t(i, :) \cdot y\}$.

We prove that $E_B^t \subseteq E_C^t$. $\forall i \in E_B^t$, we have $0 < \underline{U}_a^t(\pi^t, :) \cdot y - \overline{U}_a^t(i, :) \cdot \mathrm{y} \leq min\{U_a^t(\pi^t, :) \cdot y - U_a^t(i, :) \cdot y\} = \Omega^t(k^t, i, :) \cdot \mathrm{y}$, thereby $i \in E_C^t$.

Since $E_B^t \subseteq E_C^t$, thus we have $\gamma_B^t \leq \gamma_C^t$, thus $\sum_{t \in TL} \rho^t \gamma_B^t \leq \sum_{t \in TL} \rho^t \gamma_C^t$.□

5.3 Playing the Chemical Plant Protection Game involving attackers with bounded rationality

In this sub-section, attackers with bounded rationality in the Chemical Plant Protection game are modelled. Three different behaviour models of attacker are investigated, namely, the epsilon-optimal attacker, the monotonic-optimal attacker, and the MiniMax attacker. All these three attacker models are integrated to the Stackelberg CPP game, which means that the defender moves first, and the attackers follow. Furthermore, the monotonic-optimal attacker is investigated in the Interval CPP game with only one type of attacker, and a game solution named Monotoic MaxiMin Solution for the Interval CPP game (MoSICP) is defined (24). The MoSICP solution incorporates both boundly rational attackers and distribution-free uncertainties into the CPP game. The epsilon-optimal attacker model, being correlated to the defender's distribution-free uncertainties, and the MiniMax attacker model, being the most conservative model, are therefore investigated in the Bayesian Stackelberg CPP game framework, instead of in the Interval CPP game framework. The defender, being the model of the industrial defenders, is still assumed to behave rationally to maximize her payoff.

5.3.1 Motivation

The chemical plant protection (CPP) game, proposed in Chapter 4, is able to model intelligent interactions between the industrial defender and potential so-called adversaries (or in other words, human attackers). Moreover, by extending the CPP game to a Bayesian game, multiple types of attackers can be modelled in the game. The interval CPP game, proposed in section 5.2, fills the gap between the difficult requirement of a lot of quantitative data for the CPP game and the difficulties of obtaining these input numbers. Nonetheless, all previous models assume rational players in the game.

However, the players of the CPP game (i.e., the defender and the adversaries) are not always rational players, on the contrary, especially terrorist attackers. Guikema (25) points out that the rationality assumption brings modelling convenience to the modellers and that it is also a common assumption in a wide variety of fields. However, Guikema (25) further indicates that spontaneous attackers are not behaving to maximize their subjective expected utilities while in case of a premediated attacker, it is difficult to quantify the attacker's emotional dimension (e.g., honor). Therefore, game theoretic models must be extended to be able to deal with such 'bounded-rational' attackers.

Many researches study games with bounded-rational players. Among others, in the security game domain, several representative attacker models are:

1) the epsilon-optimal attacker (13), as also explained in section 5.3.2.

2) the quantal response equilibrium (QRE) (15, 26), which is only defined for games with discrete strategies and the probabilities that each pure strategy would be played can be calculated by Formula (5.30), in which x_i denotes the probability that pure strategy i would

be played, $EXP(\cdot)$ represents the exponential function, λ is a constant real number, A is the payoff matrix and $e_i \cdot A \cdot y$ is the player's payoff by responding strategy i to his opponent's strategy y, m is the number of pure strategies.

$$x_i = \frac{EXP(\lambda \cdot e_i \cdot A \cdot y)}{\sum_{k=1,2,\dots,m} EXP(\lambda \cdot e_k \cdot A \cdot y)} \cdots\cdots (5.30)$$

3) the $level - k$ thinking model [27], which assumes that a level k $(k = 1, 2, \dots)$ player is playing the game optimally presuming his opponent is a level $(k - 1)$ player and the level 0 player is a non-strategic player who randomly chooses his strategies.

4) the monotonic optimal attacker [17, 28], as explained in section 5.3.3 in this book.

5) the MiniMax attacker, as explained in section 5.3.4 in this book.

The quantal response equilibrium and the $level - k$ thinking model strictly rely on their modelling parameters, namely the λ for the QRE and the k for the $level - k$ model. These parameters are difficult, if not impossible, to be obtained. Tambe and his co-authors employed both simulation games and machine learning techniques for estimating the λ for the QRE [29]. There are also real experiments, for instance, the "beauty contest games" [16], for evaluating how people behave (i.e., what the value of k should be) in the $level - k$ thinking framework. However, on the one hand these parameters can vary between people and on the other hand a terrorist may be thinking completely different from ordinary people, and thus, the QRE and the $level - k$ thinking model are not investigated in this book for the Chemical Plant Protection game.

5.3.2 Epsilon-optimal attacker

5.3.2.1 Definition of an 'Epsilon-optimal attacker'

Definition 5.1: An attacker is called an epsilon-optimal attacker if he would play any strategies from a possible strategy set PS as defined in Formula (5.31), under a condition that the defender commits a mixed strategy y.

$$PS = \{s_a \in S_a | BRP - U_a(s_a, :) \cdot y \leq \varepsilon\} \cdots\cdots (5.31)$$

In Formula (5.31), BRP represents the attacker's best response payoff to the defender's strategy y, i.e., $BRP = \max_{s_a \in S_a} U_a(s_a, :) \cdot y$; ε is a user-defined constant real number.

The definition implies (and reveals) that an epsilon-optimal attacker would not always play his best response strategy, instead, he may deviate to any strategies that have close payoffs to his best response payoff.

The 'epsilon-optimal attacker' concept is investigated in this chapter because, on the one hand, such concept represents a typical human behaviour rule. Human beings are not machines and when they make decisions or do calculations, small errors are generally ignored. In daily lives, people do even not carry out numerical calculations, and they make decisions depending just on a fuzzy intuitive. For instance, in a plant with multiple dangerous oil tanks, the defender and a professional (knowledgeable) attacker can perform certain calculations and, by doing so, know which tank will lead to the most severe consequences, and determine a priority list of tanks to attack. However, a non-professional adversary not carrying out any calculations, may have identical interests with respect to the different oil tanks, since all these oil tanks have a certain dangerousness level according to intuition.

On the other hand, the epsilon-optimal attacker model may also reflect the defender's uncertainties on the attacker's payoff. The attacker's best response payoff BRP is calculated based on his payoff matrix. In reality, the defender may not know the exact numbers of the attacker's payoff matrix, thus there might be errors on the defender's estimation on the attacker's BRP. Therefore, the defender sets a tolerance (i.e., ε) to increase the robustness of her decision. In section 5.3.2.3, we will see that the algorithm for solving the chemical plant protection game played by an epsilon-optimal attacker is actually similar to the algorithm IBGS that we developed for solving general bi-matrix games with distribution-free uncertainties.

5.3.2.2 Game modelling of the 'Epsilon-optimal attacker'

In the definition of the Stackelberg equilibrium for the CPP game (see Formulas (4.19) and (4.20), in section 4.4.2), the defender first commits a mixed strategy y, then the attacker plays his best response, and the defender can also work out the attacker's best response, thus the defender plays accordingly. In the CPP game played by a rational defender and an epsilon-optimal attacker, the procedure can be illustrated as follows:

1) the defender commits a mixed strategy y;
2) the attacker calculates his best response strategy and the corresponding best response payoff (BRP);
3) the defender can also work out the attacker's BRP, and she furthermore calculates the attacker's possible strategies set PS, according to Formula (5.31);
4) the defender conservatively thinks that the attacker would play a strategy $s_a \in PS$ which minimizes her payoff;
5) based on procedures 1) to 4), the defender plays optimally.

Steps 3 and 4 are different to the procedure of calculating the Stackelberg equilibrium. In step 3, the attacker is assumed bounded-rational, thus the defender can only work out a possible strategies set, instead of knowing that the attacker will definitely play his best response strategy. In step 4, the defender does not know which strategy the attacker would use, thus she chooses to play conservatively, presuming that among all possible strategies in the PS, the worst one to her will be the attacker's choice.

5.3.2.3 Solving the CPP game with 'Epsilon-optimal attackers'

As we discussed in 5.2.1 that an epsilon-optimal attacker model can also reflect the defender's uncertainties on the attacker's payoff. Therefore, an algorithm as shown in Formula (5.32) is proposed to solve the CPP game with epsilon-optimal attackers. The algorithm is deviated from the IBGS, which is developed in section 5.2.3 for solving games with distribution-free uncertainties. Only the constraints $c1$ and $c2$ in Formula (5.32) are different from the constraints in the IBGS, while other constraints as well as the cost functions and the variables are the same. For this reason, we only explain constraints $c1$ and $c2$ in this algorithm and the explanation of other elements of the algorithm are the same as to the explanation in the IBGS algorithm, in section 5.2.3.

$$\max_{q,h,\gamma,y,R} \sum_{t\in TL} p^t \gamma^t$$

$$s.t.\begin{cases} c1.\ \ 0 \le R^t - U_a^t(i,:)\cdot y \le (1-h_i^t)\cdot\Gamma, \ \ \forall i \in M^t \\ c2.\ \ (q_i^t - 1)\cdot\Gamma \le U_a^t(i,:)\cdot y - R^t + \varepsilon^t \le q_i^t\cdot\Gamma, \ \ \forall i \in M^t \\ c3.\ \ \Gamma\cdot(1-q_i^t) + U_d^t(i,:)\cdot y \ge \gamma^t, \ \ \forall i \in M^t \\ c4.\ \ q_i^t \ge h^t{}_i, \ \ \forall i \in M^t \\ c5.\ \ q_i^t, h_i^t \in \{0,1\} \\ c6.\ \ \sum h^t = 1 \\ c7.\ \ \sum y = 1, y_i \in [0,1] \\ c8.\ \ R^t, \gamma^t \in R \end{cases} \qquad \cdots\cdots(5.32)$$

In Formula (5.32), $U_a^t(i,:)$ is the i^{th} row of the attacker t's payoff matrix; ε^t is the user-defined tolerance of attacker t. Other notations are the same as explained in section 5.2.3 below the IBGS algorithm. Constraint $c1$ calculates the attacker's best response payoff R^t. Note that in $c1$, if $h_i^t = 1$, then $R^t = U_a^t(i,:)\cdot y$, while if $h_i^t = 0$, then $R^t \ge U_a^t(i,:)\cdot y$. Therefore, $R^t = \max_{i\in M^t} U_a^t(i,:)\cdot y$ and $h_i^t = 1$ means that strategy i is the attacker t's best response strategy to the defender's committed strategy y. Constraint $c2$ defines the attacker's possible response strategy set PS. Note that in $c2$, if $R^t - U_a^t(i,:) < \varepsilon^t$, then $q_i^t = 1$, if $R^t - U_a^t(i,:) > \varepsilon^t$, then $q_i^t = 0$, while if $R^t - U_a^t(i,:) = \varepsilon^t$, then q_i^t can be either 0 or 1. Therefore, $q_i^t = 1$ indicates that strategy i belongs to the PS, for attacker type t (see the definition of PS in Formula (5.31)).

Similar to the IBGS algorithm, the above algorithm calculates the defender's conservative payoff, knowing that she is playing the game with epsilon-optimal attackers.

5.3.3 Monotonic optimal attacker

5.3.3.1 Definition of a 'Monotonic optimal attacker'

Definition 5.2: An attacker is called a monotonic optimal attacker if he plays a mixed strategy $x \in Q(y)$ as defined in Formula (5.33), in a condition that the defender commits a mixed strategy y.

$Q(y) = \{x \in X | \forall(i,j) \in E(y), x_i \ge x_j\}$ ··················(5.33)

In which,

$E(y) = \{(i,j) | U_a(i,:)\cdot y \ge U_a(j,:)\cdot y, \forall i,j = 1,2,\dots,m;\ i \ne j\}$ ··················(5.34)

for a CPP game or a general bi-matrix game, and,

$E(y) = \{(i,j) | \Delta_{ij}^{min} \ge 0, \forall i,j = 1,2,\dots,m;\ i \ne j\}$ ··················(5.35)

for an interval CPP game.

$E(y)$ is a set of attacker's pure strategy pairs of which the former strategy would bring higher payoff to the attacker than the latter strategy. $Q(y)$ is a subset of the attacker's mixed strategy space, satisfying that if a pure strategy pair $(i,j) \in E(y)$, then the probability that the attacker would play strategy i will be higher than the probability that he plays strategy j.

The definitions of $Q(y)$ and $E(y)$ reveal the key property of the monotonic optimal attacker model, that is, *for a committed defender's strategy y, the attacker is assumed to be* *more likely to respond a pure strategy with higher payoff than to respond a pure strategy that has a lower payoff*. For instance, knowing the defender's strategy y, if the attacker would have a payoff u_1 by playing strategy s_{a1} and a payoff u_2 by playing strategy s_{a2}, then if $u_1 \geq u_2$, the attacker would be more likely to play strategy s_{a1} than to play s_{a2} (but s_{a2} still has a probability of being played), and vice versa.

Comparing to the epsilon-optimal attacker model, the quantal response attacker model, and the $level - k$ thinking attacker model, the assumption of the monotonic optimal attacker is quite relaxed. In the epsilon-optimal attacker model, a user-defined tolerance (i.e., ε) is required. However, this tolerance can be difficult to estimate. In the quantal response attacker model, a model-specific parameter λ is required, and the attacker's behaviour critically depends on that parameter, see Formula (5.30). In the $level - k$ thinking model, it is difficult to decide which level that the attacker is situated in, thus the defender would not be able to operate optimally. The monotonic optimal attacker model, however, does not depend on any extra parameters.

5.3.3.2 Game modelling of the 'Monotonic optimal attacker'

A **Mo**notonic MaxiMin **S**olution for the **I**nterval **CP**P game (MoSICP) is

$arg \max_{y \in Y} \min_{x \in Q(y)} x^T \cdot U_d \cdot y$ ··(5.36)

In which $Q(y)$ is as defined in Formulas (5.33) and (5.35).

The definition of the MoSICP is different to the Monotonic MaxiMin Solution (MMS) defined in Jiang et al. [(17)], of which the $Q(y)$ is defined by Formulas (5.33) and (5.34). Figure 5.1 illustrates the differences of the definition of $E(y)$ in the MoSICP and in the MMS. The adversary's pure strategies are shown on the X axis, while the Y axis represents the defender's estimation of the attacker's payoff. The defender is assumed to play a mixed strategy, and the attacker has seven different pure strategies to respond. The attacker's payoff by responding different strategies, without considering distribution-free uncertainties, are represented by red dots in the figure. The defender's estimation of the range of the attacker's payoffs, when there are distribution-free uncertainties, are denoted by the vertical lines. Furthermore, in this illustrative figure, the parameter coupling problem descript in section 5.2.4 is excluded. Therefore, Δ_{kl}^{min} would be equal to the lowest point of vertical line k minus the highest point of vertical line l.

Formula (5.37) shows the $E(y)$ for the MMS according to Formulas (5.33) and (5.34). For example, Figure 5.1 shows that the attacker's payoff by responding strategy 1 is higher than by responding strategy 2, thus $(1,2) \in E(y)$ in Formula (5.37). Other strategy pairs can be explained analogously. Formula (5.38) demonstrates the $E(y)$ for the MoSICP, according to Formulas (5.33) and (5.35). For example, $(3,4) \in E(y)$ in Formula (5.38), since in Figure 5.1, the lowest point of the third vertical line is higher than the highest point of the fourth vertical line.

For the MMS, $E(y) = \left\{ \begin{matrix} (1,2),(1,3),(1,4),(1,5),(1,6),(1,7),(3,2), \\ (3,4),(3,6),(3,7),(4,2),(5,2),(5,3),(5,4), \\ (5,6),(5,7),(6,2),(6,4),(7,2),(7,4),(7,6) \end{matrix} \right\}$. ···············(5.37)

For the $MoSCIP$, $E(y) = \{(1,2),(1,4),(1,6),(1,7),(3,2),(3,4),(5,2),(5,4),(5,6)\}$.
···(5.38)

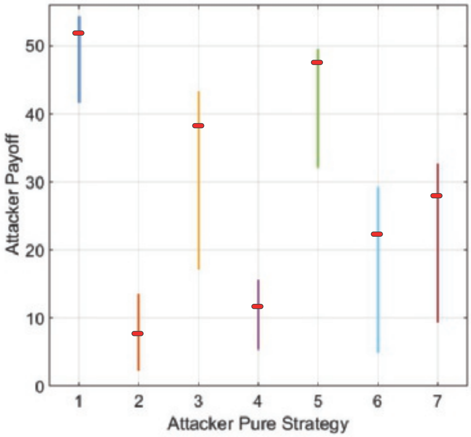


Figure 5. 1. Attacker's payoff by responding different pure strategies to y

5.3.3.3 Calculating the MoSICP

The definition of the MoSICP contains both the maximal and the minimal optimization problem, and the inner minimal problem (i.e., $\min_{x \in Q(y)} x^T \cdot U_d \cdot y$) involves a $Q(y)$, thus it is not a standard linear minimal problem. Jiang et al. [(17)] proposed an approach for transforming this problem into a standard linear problem by using duality. In this research, we mainly follow the idea from Jiang et al. [(17)].

For a given y, the inner optimization problem (i.e., $\min_{x \in Q(y)} x^T \cdot U_d \cdot y$) in the MoSICP can be written as Formula (5.39), of which the constraints demonstrates the definition of the $Q(y)$.

$$\min_x x^T \cdot (U_d \cdot y)$$
$$s.t. \begin{cases} x_i \ge x_j, \forall (i,j) \in E(y) \\ x_i \ge 0 \\ \sum x = 1 \end{cases} \qquad \cdots\cdots (5.39)$$

For any attacker strategy pair $(i,j) \in E(y)$, the first constraint in Formula (5.39) can be written as $(e_i^m - e_j^m) \cdot x \ge 0$, while for any strategy pair $(i,j) \notin E(y)$, it can be written as $0_{1\times m} \cdot x \ge 0$. e_i^m is a row vector with a length of m. The i^{th} entry of e_i^m is 1, and other entries are 0. For instance, $e_3^4 = [0\ 0\ 1\ 0]$. $0_{1\times m}$ is a row zero vector with a length of m. With this definition, Formula (5.39) can further be formulated as:

$$\min_x x^T \cdot (U_d \cdot y)$$
$$s.t. \begin{cases} W \cdot x \ge 0 \\ x_i \ge 0 \\ \sum x = 1 \end{cases} \qquad \cdots\cdots (5.40)$$

In which W is an $m(m-1) \times m$ matrix, whose ij^{th} row is $e_i^m - e_j^m$ if $(i,j) \in E(y)$ and is $0_{1\times m}$ otherwise.

With a given y, the $E(y)$ is determined, thus Formula (5.40) is a standard linear programming problem. By applying the duality theory on Formula (5.40), we obtain:

$$\max_{\theta,t} t$$
$$s.t.\begin{cases}\theta_i \geq 0, i=1,2,\ldots,M \\ W^T \cdot \theta + t \leq U_d \cdot y\end{cases} \cdots\cdots (5.41)$$

In which θ and t are the dual variables and $M = m(m-1)$. Note that Formula (5.41) works only for a determined $E(y)$, and different y may result in different $E(y)$. Define a binary variable $z_{ij} \in \{0,1\}$, setting $z_{ij} = 1$ if the attacker strategy pair $(i,j) \in E(y)$ and $z_{ij} = 0$ otherwise. Combining the definition of $E(y)$ (Formula (5.35)), the definition of z_{ij} can also be expressed as:

$$\Delta_{ij}^{min} + \Gamma \cdot (1 - z_{ij}) \geq 0, \cdots\cdots (5.42)$$

And,

$$\Delta_{ij}^{min} - \Gamma \cdot z_{ij} < 0. \cdots\cdots (5.43)$$

In these 2 inequalities, $\Delta_{ij}^{min} \geq 0 \Leftrightarrow z_{ij} = 1$, and $\Delta_{ij}^{min} < 0 \Leftrightarrow z_{ij} = 0$.

With the support of z_{ij}, we may rewrite the ij^{th} row of matrix W as $z_{ij} \cdot (e_i^m - e_j^m)$. Moreover, the second constraint in Formula (5.41) can be rewritten as:

$$\sum_{i=1,j=1,i\neq j}^{i=m,j=m} \bar{w}_{ij,k} \cdot \theta_{ij} \cdot z_{ij} + \mathrm{t} \leq U_d(k,:) \cdot y,\ k = 1,2,\ldots,m \cdots\cdots (5.44)$$

In which $\bar{w}_{ij,k}$ is the (ij,k) entry of a matrix whose ij^{th} row is $e_i^m - e_j^m$.

Formula (5.44) is not a standard linear constraint either, due to the existence of the multiplies of variables (i.e., $\theta_{ij} \cdot z_{ij}$). We further define $\omega_{ij} = \theta_{ij} \cdot z_{ij}$, such that we obtain a mixed integer linear programming (MILP) based algorithm, for calculating the MoSICP, as shown in Formula (5.45).

$$\max_{y,\omega,t,z} t$$
$$s.t.\begin{cases} c1. & \sum_{i=1,j=1,i\neq j}^{i=m,j=m} \bar{w}_{ij,k} \cdot \omega_{ij} + t \leq U_d(k,:) \cdot \mathrm{y}, \quad \forall k \in M \\ c2. & 0 \leq \omega_{ij} \leq N \cdot z_{ij}, \quad \forall i,j \in M, i \neq j \\ c3. & \Delta_{ij}^{\min} + N \cdot (1 - z_{ij}) \geq 0, \quad \forall i,j \in M, i \neq j \\ c4. & \Delta_{ij}^{\min} - N \cdot z_{ij} < 0, \quad \forall i,j \in M, i \neq j \\ \mathrm{c}5. & (1 - z_{ij}) + (1 - z_{jh}) + z_{ih} \geq 1, \quad \forall i,j,h \in M, i \neq j \neq h \\ c6. & z_{ij} \in \{0,1\}, y \in Y, \quad \forall i,j \in M, i \neq j \end{cases} \cdots\cdots (5.45)$$

In the algorithm, constraint $c1$ directly refers to Formula (5.44) (and thus it refers to the second constraint in Formula (5.41)); in constraint $c2$, if $z_{ij} = 0$, then $\omega_{ij} = \theta_{ij} \cdot 0 = 0$, if $z_{ij} = 1$, then $\omega_{ij} = \theta_{ij} \cdot 1 \geq 0$, therefore, $c2$ refers to the first constraint in Formula (5.41); constraints $c3, c4$ reflect the definition of z_{ij}, as also shown in Formulas (5.42) and (5.43); constraint $c5$ represents that if (i,j) and (j,h) belong to $E(y)$ then (i,h) must also belong to $E(y)$.

Proposition 6.1. According to the definition of MoSICP, if (i, j) and (j, h) belong to $E(y)$ then (i, h) must also belong to $E(y)$, that is to say, $\forall i, j, h \in M, i \neq j \neq h$, then $z_{ij} = 1$ and $z_{jh} = 1$ imply that $z_{ih} = 1$.

Proof: If (i, j) and (j, h) belong to $E(y)$, then we have that $\Delta_{ij}^{min} \geq 0$ and $\Delta_{jh}^{min} \geq 0$ (see Formula (5.35)). Furthermore, we have that $0 \leq \Delta_{ij}^{min} + \Delta_{jh}^{min} = \min\{U_a(i,:) \cdot y - U_a(j,:) \cdot y\} + \min\{U_a(j,:) \cdot y - U_a(h,:) \cdot y\} \leq \min\{U_a(i,:) \cdot y - U_a(j,:) \cdot y + U_a(j,:) \cdot y - U_a(h,:) \cdot y\} = \Delta_{ih}^{min}$. Therefore, $\Delta_{ih}^{min} \geq 0$ and thus (i, h) also belongs to $E(y)$.

Constraint $c4$ is a strict inequality, which cannot be processed by typical MILP solvers. When we implement the algorithm, $c4$ is typed in as $\Delta_{ij}^{min} - \Gamma \cdot z_{ij} + \sigma \leq 0$ in which σ is a small constant real number. The usage of σ may reduce the defender's expect optimal payoff by excluding a sub-set of Y. Furthermore, the sub-set can be worked out as $\tilde{Y} = \{y \in Y | -\sigma < \Delta_{ij}^{min} < 0, \forall i, j\}$. By fixing σ sufficiently small, $\tilde{Y}$ can be reasonably bounded.

When the defender's optimal strategy y is obtained, according to Formulas (5.33) and (5.35), sets $E(y), Q(y)$ can be constructed. By solving the linear program $\min_{x \in Q(y)} x^T \cdot (U_d \cdot y)$, the attacker's response strategy can be calculated.

5.3.4 MiniMax attacker

5.3.4.1 Definition of a 'MiniMax attacker'

Definition 5.3: a MiniMax attacker is an attacker who plays totally converse to the defender's interest. That is, knowing a defender's committed strategy y, a MiniMax attacker would play a strategy $k = arg \min_{s_a \in S_a} U_d(s_a,:) \cdot y$.

MiniMax attackers can be treated as totally irrational players, since they do not care about their own payoffs at all, instead, they look at minimizing the defender's payoff.

5.3.4.2 Game modelling of the 'MiniMax attacker'

In a MiniMax attacker case, the adversary's payoff is obviously no longer needed for the game modelling. The defender knows that whatever strategy she plays, the attacker aims to minimize her payoff. Therefore, the defender can play optimally as:

$$\tilde{y} = \text{argmax}_{y \in Y}\{\min_{x \in X} x^T \cdot U_d \cdot y\} \cdots\cdots\cdots (5.46)$$

5.3.4.3 Solving the CPP game with 'MiniMax attackers'

Solving the CPP game with 'MiniMax attackers' is equivalent to solving a zero-sum game. Formula (5.47) shows an algorithm for calculating the defender's optimal strategy and the corresponding payoff.

$$\max_{y,\gamma} \sum\nolimits_{t \in TL} p^t \cdot \gamma^t$$
$$s.t. \begin{cases} c1: U_d^t(i,:) \cdot \mathrm{y} \geq \gamma^t, \forall i \in M^t, t \in TL \\ c2: y \in Y \end{cases} \cdots\cdots\cdots (5.47)$$

The first constraint shows that if the defender plays a mixed strategy y, then the attacker would play a pure strategy $i \in M$ that minimizes the defender's payoff. The cost function aims to maximize the defender's minimal payoff (result in the attacker's choice). This algorithm is developed based on an important property of the attacker's choice: knowing y, the attacker's choice would be a pure strategy, instead of being a mixed strategy.

5.4 Case study – CPP game applied to a refinery considering uncertainties

5.4.1 Case study setting

This section is a follow-up study of the case study in section 4.6. Therefore, all the settings descript in section 4.6 will be inherited by this section. Furthermore, results from section 4.6 are also used in this section for comparison.

5.4.2 Results

The game modelled for the case study is a 2-player game. The defender is determined, while the attacker can be either a terrorist or an activist. The prior probabilities of these two types of attackers are $p^t = (3/7, 4/7)$. The defender has $n = 3^{1+\sum_{r=1}^{2}(ent(r)+sub(r))} = 729$ pure strategies. The terrorist has $m^{terr} = 6$ pure strategies and the activist has $m^{acti} = 7$ pure strategies, as shown in Tables 4.9 and 4.10, in Chapter 4. If Formulas (4.7), (4.8), (5.2), (5.3), (5.20), (5.25), and (5.27) are filled in with data from tables shown in section 4.6.2, the attacker's lower bound payoff matrices $\underline{U}_a^{terr}$ and $\underline{U}_a^{acti}$, the attacker's upper bound payoff matrices $\overline{U}_a^{terr}$ and $\overline{U}_a^{acti}$, and the coefficients matrices Ω^{terr} and Ω^{acti}, can be obtained.

The Interval CPP game solution, the robust solution considering epsilon-optimal attackers, the MoSICP solution, and the MiniMax solution are calculated for the case study. For the sake of clarity, we first show the results of the CPP game of the case study by considering one type of attacker, in section 5.4.2.1. Subsequently, we discuss the results of the game by considering multiple types of attackers (i.e., both the terrorist and the activist) in section 5.4.2.2. (Inter-) comparison of these different solutions are also shown.

5.4.2.1 Single attacker type: the Activist

Interval CPP Game solution

Tables 5.2 and 5.3 give the defender's optimal solution of the game from the IBGS algorithm and from the ICGS algorithm respectively, in case that she has distribution-free uncertainties on the activist's parameters. U_d^{acti}, U_a^{acti}, $\underline{U}_a^{acti}$, $\overline{U}_a^{acti}$, and Ω^{acti} are used as input information for these two algorithms.

The activist's payoffs by responding with different strategies to the defender's IBGS optimal strategy and to the defender's ICGS optimal strategy are shown on the left-hand side panel and on the right-hand panel of Figure 5.2, respectively. The x axis of Figure 5.2 denotes the activist's pure strategies, the y axis denotes the activist's payoff. The bold red dots denote the activist's payoff by responding with different pure strategies, without considering distribution-free uncertainties. Conversely, if the distribution-free uncertainties are considered, then the activist's payoffs by responding with different strategies to the defender's optimal strategy cannot be obtained directly and only the ranges can be calculated, as shown in the figure by vertical lines. The horizontal lines represents the activist's maximal lower bound payoffs.

The left-hand side panel of Figure 5.2 shows that if the defender plays her IBGS optimal strategy and if there would be no uncertainties, then the activist's best response would be either s_{e4} or s_{e5}. When uncertainties are considered, then the defender knows that strategy s_{e5} can deliver the activist the highest lower bound payoff, being 54.4 k€. Furthermore, the upper bound payoff of strategies s_{e4}, s_{e5}, and s_{e6} are higher than 54.4 k€, thus these three strategies all have the possibilities of being the attacker's best response. The upper bound payoff of strategy s_{e1} equals exactly 54.4 k€, and it is excluded from the activist's possible best response set, as we explained in section 5.2.3. The defender's payoffs would be -243.6 k€, -243.6 k€, and -185.3 k€ respectively, if the activist responds with s_{e4}, s_{e5}, or s_{e6}. The defender conservatively treats the worst case as the real case, thus the defender has an expect payoff from the IBGS solution of -243.6 k€.

Table 5. 2. Defender's optimal strategy from the IBGS algorithm

Defender Pure Strategy	Probability
2×1×1×1×1×1	0.5817
2×1×1×2×1×1	0.2426
2×2×1×1×1×1	0.1757

Table 5. 3. Defender's optimal strategy from the ICGS algorithm

Index	Defender's Pure Strategy	Probability
$s_{d-ICGS-1}$	2×1×1×1×1×1	0.3238
$s_{d-ICGS-2}$	2×1×1×2×1×1	0.5173
$s_{d-ICGS-3}$	2×2×1×1×1×1	0.1589

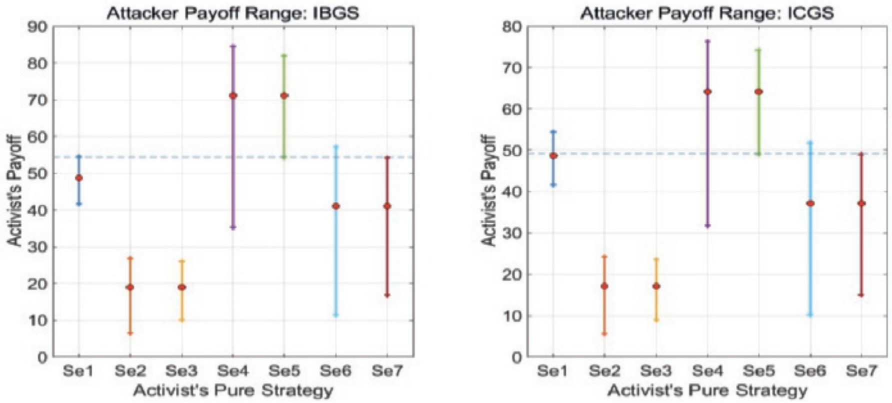


Figure 5. 2. Attacker's payoff range

The right-hand side panel of Figure 5.2 reveals that if the defender plays her ICGS optimal solution, then s_{e5} would deliver the activist the highest lower bound payoff and strategies s_{e1}, s_{e4}, s_{e5}, and s_{e6} are all possible best responses (according to the criteria used in the IBGS algorithm). However, strategies s_{e1}, s_{e4}, and s_{e6} share some parameters (e.g., $\tilde{P}_0^z$) with strategy s_{e5}. The activist's payoff differences $\Delta_{se5-se1}$, $\Delta_{se5-se4}$, and $\Delta_{se5-se6}$ indicate that strategies s_{e1} and s_{e6} are always worse than strategy s_{e5}, as shown in Table 5.4. Therefore, the ICGS algorithm leads to s_{e4} and s_{e5} to be the activist's possible best response strategies. The defender's expected payoff from the ICGS solution equals -238.6 k€, being higher than the defender's expected payoff from the IBGS solution.

Table 5. 4. Payoff differences

Payoff Differences	Result involving the shared parameters	Comments
$\Delta_{se5-se1}$	$0.5971 \cdot \tilde{P}_0^z$	> 0
$\Delta_{se5-se4}$	$-0.0302 \cdot \tilde{P}_0^z \cdot \tilde{p}_y \cdot \tilde{L}$	< 0, the $\tilde{p}_y \cdot \tilde{L}$ denotes the conditional expected loss of target T4
$\Delta_{se5-se6}$	$4.6370 \cdot \tilde{P}_0^z$	> 0

The following explains how $\Delta_{se5-se4}$ can be obtained. $\Delta_{se5-se1}$ and $\Delta_{se5-se6}$ can be calculated analogously, as theoretically explained in section 5.2.4. Numbers used in the following formulas are derived from Table 4.12, in Chapter 4.

$u_a^{min}(s_{e5}, s_{d-ICGS-1}) = \tilde{P}_0^z \cdot 0.26 \cdot 0.78 \cdot\ \tilde{p}_y \cdot \tilde{L} - C_a$

$u_a^{min}(s_{e5}, s_{d-ICGS-2}) = \tilde{P}_0^z \cdot 0.26 \cdot 0.78 \cdot 0.68 \cdot\ \tilde{p}_y \cdot \tilde{L} - C_a$

$u_a^{min}(s_{e5}, s_{d-ICGS-3}) = \tilde{P}_0^z \cdot 0.26 \cdot 0.78 \cdot\ \tilde{p}_y \cdot \tilde{L} - C_a$

$u_a^{max}(s_{e4}, s_{d-ICGS-1}) = \tilde{P}_0^z \cdot 0.32 \cdot 0.8 \cdot\ \tilde{p}_y \cdot \tilde{L} - C_a$

$u_a^{max}(s_{e4}, s_{d-ICGS-2}) = \tilde{P}_0^z \cdot 0.32 \cdot 0.8 \cdot 0.68 \cdot\ \tilde{p}_y \cdot \tilde{L} - C_a$

$u_a^{max}(s_{e4}, s_{d-ICGS-3}) = \tilde{P}_0^z \cdot 0.32 \cdot 0.65 \cdot 0.8 \cdot\ \tilde{p}_y \cdot \tilde{L} - C_a$

$\Delta_{se5-se4}=$
$0.3238 \cdot \left(u_a^{min}(s_{e5}, s_{d-ICGS-1}) - u_a^{max}(s_{e4}, s_{d-ICGS-1})\right) + 0.5173 \cdot \left(u_a^{min}(s_{e5}, s_{d-ICGS-2}) - u_a^{max}(s_{e4}, s_{d-ICGS-2})\right) + 0.1589 \cdot \left(u_a^{min}(s_{e5}, s_{d-ICGS-3}) - u_a^{max}(s_{e4}, s_{d-ICGS-3})\right) =$
$-0.0302 \cdot \tilde{P}_0^z \cdot\ \tilde{p}_y \cdot \tilde{L}$

MoSICP solution

The defender's optimal strategy from the MoSICP solution is given in Table 5.5. U_d^{acti} and Ω^{acti} are the inputs for the algorithm proposed in section 5.3.3.

Table 5.6 shows the Δ matrix. According to the definition of MoSICP in section 5.3.3, we know that:

$E(y) = \{(1,2), (1,3), (4,2), (4,3), (4,6), (5,2), (5,3), (5,7)\}$ ··························(5.48)

$Q(y) = \{x \in X | x_1 \geq x_2, x_3; x_4 \geq x_2, x_3, x_6; x_5 \geq x_1, x_2, x_3, x_7\}$ ······················(5.49)

Table 5. 5. Defender's MoSICP strategy

Index	Defender's Pure Strategy	Probability
$s_{d-MoSCIP-1}$	2×1×1×1×1×1	0.5292
$s_{d-MoSCIP-2}$	2×1×1×2×1×1	0.3754
$s_{d-MoSCIP-3}$	2×1×2×1×1×1	0.0954

Table 5. 6. Δ matrix of the MoSICP

(i,j)	1	2	3	4	5	6	7
1	0	**18.06**	**21.64**	-39.43	-27.33	-11.64	-2.42
2	-45.25	0	-8.33	-71.55	-65.06	-43.71	-39.85
3	-42.75	-7.92	0	-74.66	-61.18	-46.82	-35.98
4	-13.43	**22.19**	**16.16**	0	-25.91	**4.18**	-8.14
5	**0.00**	**25.43**	**31.30**	-21.48	0	-3.64	**10.14**
6	-39.70	-3.77	-8.86	-62.66	-59.50	0	-17.40
7	-36.21	-9.23	-3.24	-68.13	-53.89	-19.09	0

Table 5.7 further shows the activist's strategy in the MoSICP solution. In the work of Jiang et al. [17] (Lemma 1) and Nguyen et al. [28], it is proven that "each action that is played with positive probability in the attacker's mixed strategy is played with equal probability". The above statement means that, any actions that have the probability of being played (i.e., strategies s_{e1}, s_{e4}, and s_{e5} in our case) should have the same probability of being played (i.e., the probabilities shown in the right-hand column should be 0.33, 0.33 and 0.33 for strategies s_{e1}, s_{e4}, and s_{e5} respectively). The mixed strategy shown in Table 5.7 does not fit this lemma. In their work, attacker's parameter uncertainties are not modelled, thus a full order theory is employed. The full order theory results in the equality of probabilities of different played actions. However, in the MoSICP, due to the existence of the distribution-free uncertainties, it is not necessary that all the played actions will be played by the same probabilities. To be more specific, we will notice that set Q (shown in Formula (5.49)) does not show any relationships between x_4 and x_1, or x_4 and x_5.

Table 5. 7. Attacker's strategy in the MoSICP

Activist's strategy	Activist's payoff
s_{e1}	**0.1892**
s_{e2}	0
s_{e3}	0
s_{e4}	**0.6216**
s_{e5}	**0.1892**
s_{e6}	0
s_{e7}	0

The defender's expected payoff from the MoSICP solution is -245.4 k€.

MiniMax solution

Table 5.8 shows the defender's optimal strategy from the MiniMax solution. Table 5.9 further illustrates the defender's and the activist's payoff with regard to different activist's responses to the defender's optimal strategy. The third column shows that the activist's best response should be s_{e4}, bringing the activist a payoff of 81.1 k€. However the second column shows that a MiniMax activist would also play strategy s_{e1}, since s_{e1} minimizes the defender's payoff as well. The second column of Table 5.9 indicates that the defender's optimal payoff from the MiniMax solution would be -251.0 k€.

Table 5. 8. Defender’s MiniMax Solution strategy

Defender’s Pure Strategy	Probability
2×1×1×1×1×1	0.9440
2×2×1×1×1×1	0.0560

Table 5. 9. Player’s payoff when the activist plays different strategies

Activist’s Response	Defender’s Payoff	Activist’s Payoff
s_{e1}	**-251.0002**	48.7420
s_{e2}	-192.4802	21.8000
s_{e3}	-190.9481	20.7056
s_{e4}	**-251.0002**	**81.0800**
s_{e5}	-246.6591	77.1402
s_{e6}	-184.5686	47.0168
s_{e7}	-183.4162	44.7120

(Inter-) Comparison of different solutions

Figure 5.3 shows the defender’s expected payoffs (y axis) from different solutions (x axis). The MiniMax represents the result from an activist who aims at minimizing the defender’s payoff; the MoSICP denotes the result from an activist who would play pure strategies that have higher payoffs with higher probabilities and the defender has distribution-free uncertainties on the activist’s parameters; the IBGS and the ICGS denote the results from a rational activist but where the defender has distribution-free uncertainties on the activist’s parameters; the NE and SSE represent the results (illustrated in section 4.6.3.1) from the game with a rational activist and complete information, for a simultaneous moving game and for a sequential moving game, respectively.

Figure 5.3 reveals that if the defender has complete information of the game while the activist has perfect information of the game (i.e., an SSE solution), the defender will have the maximal payoff, being -229.1 k€. The defender’s uncertainties on the activist would reduce her expected payoff (i) to -238.6 k€ in case of only having distribution-free uncertainties on parameters (i.e., an ICGS solution), (ii) to -245.4 k€ in case of having uncertainties on both the activist’s parameters and rationality (i.e., an MoSICP solution), and (iii) to -251.0 k€ in case of having no information about the attacker at all (i.e., an MiniMax solution). Comparison of the results of the NE solution and the SSE solution reveals the “first-mover advantage” for the defender. The NE brings the defender a payoff of -242.0 k€, being less than the payoff that the defender obtains from the SSE. The payoff differences between the IBGS and the ICGS proves the effectiveness of the algorithm ICGS by taking into account the parameter coupling problem of the CPP game.

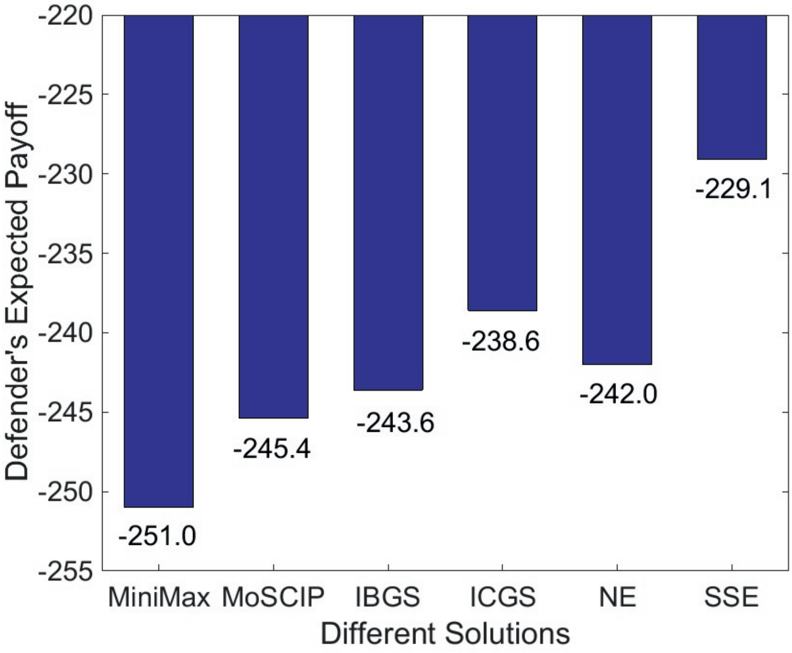


Figure 5. 3. Defender's expected payoff from different game solutions

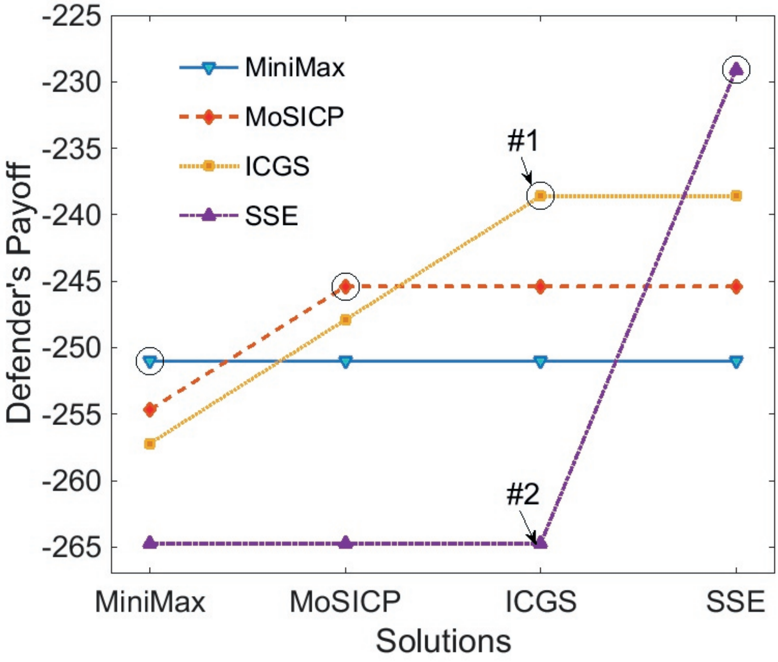


Figure 5. 4. Robustness of different solutions

Figure 5.4 demonstrates the robustness of different solutions to the case study game. Different lines in the figure denote the robustness of different solutions, i.e., as shown in the legend, the MiniMax solution, the MoSICP solution, the ICGS solution, and the Strong Stackelberg Equilibrium. The y axis denotes the defender's payoff. The x axis represents the real situation of the activist's rationality or the defender's information. For instance, the SSE line (in purple colour) denotes that the defender believes that she has complete information of the game and the activist is a rational player. Therefore, the defender plays her SSE strategy (as shown in Table 4.21). However, the real situation may not be the same as the defender thought. The four points in the SSE line represent the defender's real payoff in case of a corresponding real situation. The point "#2", for example, means that the defender's information of the attacker is incorrect and distribution-free uncertainties actually exist (as assumed in the interval CPP game), and the defender thus has a lower payoff, being -264.7 k€. Points with a circle (e.g., the point "#1") denote the situation that the assumptions of the line are satisfied.

Figure 5.4 reveals that solutions with less strict assumptions are more robust than others. The MiniMax solution, which is the most conservative solution for the defender and does not require any assumptions on the activist's behaviour and on the defender's knowledge of the activist's parameters, ensures a payoff of -251.0 k€ to the defender. Other solutions, though promising higher payoffs to the defender if the required assumptions are satisfied, may pull the defender to a quite worse situation if her assumptions would not hold and would not be true. Furthermore, the more strict assumptions the solution need, the worse the result will be if the assumptions would be false.

Two conclusions can be drawn from Figures 5.3 and 5.4:

(i) It is important for the defender to collect useful security data. Sufficient intelligence (data) would support the defender to play a better strategy to obtain a higher payoff.

(ii) The reliability of the collected intelligence (data) is important. Fake/false information is worse than no information.

5.4.2.2 Multiple attacker types: the Terrorist and the Activist

Interval CPP Game solution

Table 5.10 shows the defender's solutions of the Interval CPP game. Figure 5.5 illustrates the attackers' payoffs by responding with different pure strategies. The x and y axis, the vertical lines, the horizontal lines, and the red circles are the same as those we defined in Figure 5.2 in section 5.4.2.1. The two sub-figures on the top (bottom) are the terrorist and the activist's payoffs to the defender's optimal solution from the IBGS (ICGS) algorithm.

Table 5. 10. Defender's optimal strategy for the Interval Game

IBGS algorithm	
Defender Pure Strategy	Probability
2×2×1×1×3×1	0.0542
2×2×1×2×2×1	0.2426
2×3×1×1×2×1	0.7032
ICGS algorithm	
Defender Pure Strategy	Probability
2×2×1×2×1×1	0.5173
2×3×1×1×1×1	0.4827

Figure 5.5 shows that if the defender plays her IBGS strategy, strategies s_{v5} and s_{e5} have the highest lower bound payoffs for the terrorist and for the activist respectively. Moreover, the upper bound values of all other strategies are lower than the lower bound payoffs of these two strategies. Therefore, the terrorist and the activist's possible best responses are s_{v5} and s_{e5} respectively. If the defender plays her ICGS strategy, strategies s_{v5} and s_{e5} also have the highest lower bound payoffs for the terrorist and for the activist respectively. Furthermore, the terrorist's strategy s_{v6} and the activist's strategy s_{e1} have higher upper bound payoffs than the lower bound payoffs of strategies s_{v5} and s_{e5} respectively. Consequently, according to the IBGS algorithm, both the terrorist and the activist would have two possible best response strategies. Conversely, the ICGS algorithm indicates that both attackers have only one possible best response. This is the result of the parameter coupling problem, and Formulas (5.50) and (5.51) explain why s_{v6} and s_{e1} should be excluded from the terrorist's and the activist's possible best response set.

$$\Delta_{sv5-sv6}^{terr} = 52.67 \cdot \tilde{P}_0^z(SAL{:}\,2) \cdot \tilde{P}_{MG}(SAL{:}\,2) + 72.28 \cdot \tilde{P}_0^z(SAL{:}\,2) \cdot \tilde{P}_{MG}(SAL{:}\,3) > 0 \qquad (5.50)$$

$$\Delta_{se5-se1}^{acti} = 0.60 \cdot \tilde{P}_0^z(SAL{:}\,2) > 0 \qquad (5.51)$$

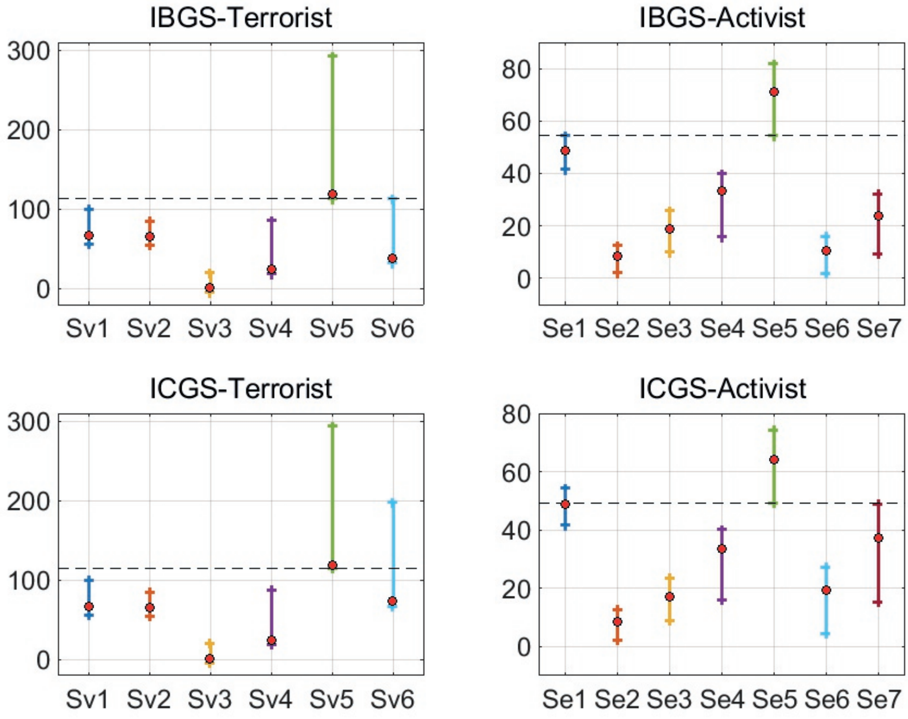


Figure 5. 5. Attackers' payoff range

Robust solution with epsilon-optimal attacker

Table 5.11 shows the defender's optimal strategy from the robust solution with epsilon-optimal attackers. The epsilon is set as $\varepsilon = 1$ for both attackers. Table 5.12 illustrates the attackers' payoffs by responding with different strategies to the defender's strategy shown in Table 5.11. The terrorist will have the highest payoff by playing strategy s_{v5}, being 128.8 k€, while the activist's highest payoff strategy is s_{e5}. The attackers would deviate from their best response strategies, and the range of the deviation is set as $\varepsilon = 1$, thus any strategy that has a payoff 1.0 k€ less than the attackers' payoff from their best response strategies can be their possible best responses. Table 5.12 shows that the terrorist's possible response would only be s_{v5} and the activist's possible response would only be s_{e5}. Note that the activist's payoff by playing s_{e1} is exactly 1.0 k€ less than his payoff by playing s_{e5}, thus s_{e1} is excluded from the activist's possible best responses.

Table 5. 11. Defender's optimal strategy to the epsilon-optimal attackers

Defender Pure Strategy	Probability
2×2×1×2×1×1	0.9042
2×2×2×2×1×1	0.0958

The defender's expected payoff by playing her robust solution as shown in Table 5.11 is therefore -252.1 k€, as calculated by Formula (5.52). Instead, if the defender does not take the attacker's bounded rationality into consideration and plays her Bayesian Stackelberg Equilibrium strategy, as shown in Table 4.25, then the activist may response s_{e1} as well as s_{e5} (though the terrorist still only responds with s_{v5}), resulting in the defender obtaining a quite low payoff, being -272.0 k€. The calculation result of this worst payoff is obtained by using Formula (5.53), and the numbers are taken from Table 4.26.

$$-272.9915 \times \frac{3}{7} - 236.3893 \times \frac{4}{7} = -252.0760. \cdots\cdots\cdots\cdots\cdots\cdots (5.52)$$

$$-273.1719 \times \frac{3}{7} - 271.0995 \times \frac{4}{7} = -271.9877. \cdots\cdots\cdots\cdots\cdots\cdots (5.53)$$

Table 5. 12. Attackers' payoff by responding with different strategies to the defender's strategy from the epsilon-optimal solution

Terrorist Pure Strategy	Terrorist Payoff	Defender Payoff	Activist Pure Strategy	Activist Payoff	Defender's Payoff
s_{v1}	67.5200	-245.0792	s_{e1}	48.7420	-270.9192
s_{v2}	65.5820	-238.6192	s_{e2}	9.2792	-194.8700
s_{v3}	1.9923	-194.8700	s_{e3}	13.0950	-200.2122
s_{v4}	27.0049	-213.3725	s_{e4}	36.0049	-221.2531
s_{v5}	**128.7686**	-272.9915	s_{e5}	**49.7420**	-236.3893
s_{v6}	79.2976	-359.0743	s_{e6}	20.6479	-191.3031
			s_{e7}	28.6841	-195.3212

MiniMax solution

Table 5.13 shows the defender's MiniMax strategy and Table 5.14 illustrates the players' payoffs when the attackers respond different pure strategies to the defender's MiniMax strategy.

Table 5. 13. Defender's optimal strategy from the MiniMax solution

Defender Pure Strategy	Probability
2×2×1×2×2×1	0.5564
2×2×1×2×2×2	0.4436

Table 5. 14. Players' payoffs when the attackers respond different strategies to the defender's optimal strategy

Terrorist Pure Strategy	Terrorist Payoff	Defender Payoff	Activist Pure Strategy	Activist Payoff	Defender's Payoff
s_{v1}	67.5200	-254.0358	s_{e1}	48.7420	**-279.8758**
s_{v2}	65.5820	-247.5758	s_{e2}	9.2792	-203.8267
s_{v3}	1.9923	-203.8267	s_{e3}	13.7598	-210.0996
s_{v4}	27.0049	-222.3291	s_{e4}	36.0049	-230.2098
s_{v5}	128.7686	**-281.9482**	s_{e5}	52.1353	-247.9831
s_{v6}	36.2562	-281.9482	s_{e6}	10.2136	-195.0427
			s_{e7}	15.1016	-197.4867

The defender's worst expected payoff from the MiniMax solution is therefore -280.8 k€, as calculated by Formula (5.54).

$$-281.9482 \times \frac{3}{7} - 279.8758 \times \frac{4}{7} = -280.7640. \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (5.54)$$

(Inter-) Comparison of different solutions

Figure 5.6 shows the defender's expected payoffs from different solutions of the CPP game, considering two types of attackers. The result is similar to the result we show in section 5.4.2.1 where only one type of attacker is considered.

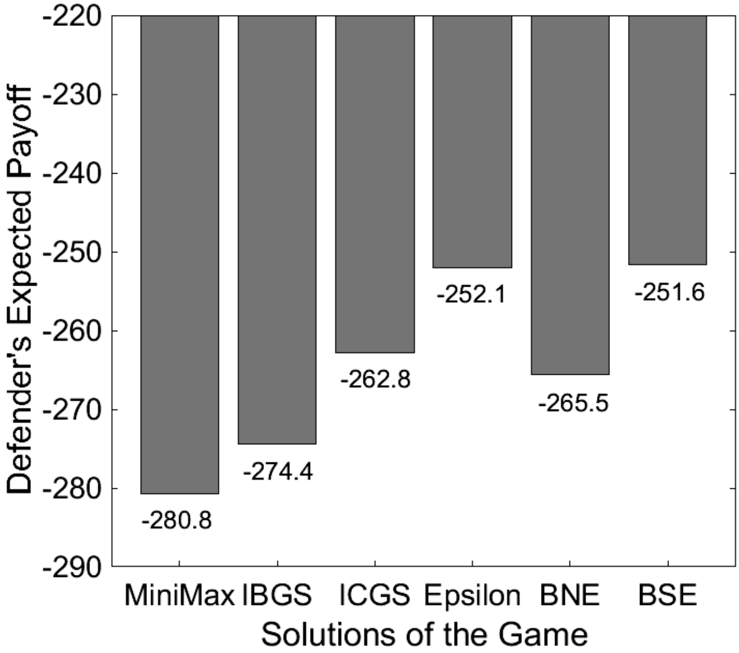


Figure 5. 6. Defender's expected payoffs from different solutions, considering multiple types of attackers

The first conclusion is that, the increase of the defender's uncertainties on the attackers would reduce the defender's expected payoff. The defender has an expected payoff of -251.6 k€ from the BSE solution, in which the defender is assumed to have complete information of the attackers and the attackers are rational players, while the defender's expected payoff is as low as -280.8 k€ from the MiniMax solution, in which the defender does not need any information of the attacker.

The second conclusion is that, in an idealistic situation, which means that both the defender and the attacker have complete information of the game and are rational players, a sequential moving game is better than a simultaneous moving game for the defender. The defender has a payoff of -265.5 k€ from the Bayesian Nash Equilibrium, being 13.9 k€ lower than her payoff from the Bayesian Stackelberg Equilibrium.

The third conclusion is that, in a multiple types of attackers situation and the defender has interval uncertainties on the attackers' parameters, the ICGS algorithm is more efficient for the defender than the IBGS algorithm. As shown in Figure 5.6, in the same setting, the ICGS brings the defender a payoff of -262.8 k€ while the IBGS only brings the defender a payoff of -274.4 k€.

Figure 5.7 further demonstrates our first conclusion by showing the results of the sensitivity analysis. The y axis of the figure denotes the defender's expected payoff. The two horizontal lines represent the defender's payoffs from the Bayesian Stackelberg Equlibrium (the line on the top) and from the MiniMax solution.

The black bold dot line together with the x axis on top show the result of the sensitivity analysis of the robust solution with epsilon-optimal attackers. The x axis denotes the value of epsilon, which can be interpreted as a measurement of the attacker's rationality: the higher the epsilon is, the less rational the attacker is. The black bold dot line reveals that the defender's expected payoff is declining when the attacker is becoming more and more irrational. At the most left-hand side, we have that epsilon equals zero, and the defender's expected payoff is as high as her BSE payoff. At the right-hand side, when epsilon equals 106, the defender's expected payoff becomes as low as her MiniMax payoff. Therefore, if the defender has the confidence that the attackers' decision tolerance is less than 106, then it is useful and necessary to estimate the attacker's real tolerance value (i.e., epsilon). Otherwise, if the defender estimates that the attackers' decision tolerance is higher than 106, then it is not necessary anymore to know the value of the attacker's real tolerance, instead, the defender can play her MiniMax strategy directly.

Furthermore, we study how the interval uncertainties on the attackers would affect the defender's optimal payoffs, and results are shown by the four curves (together with the x axis on the bottom) with legend as "IBGS-s1", "ICGS-s1", "IBGS-s2", and "ICGS-s2". Two experiments are defined, namely, s1, in which the defender has interval uncertainties on all the attackers' parameters, and s2, in which the defender only has interval uncertainties on the attackers' monetary parameters. In both experiments, the defender's parameters are the same as given in Table 4.11 through Table 4.16, while the attacker's parameters are defined by the following rules: Rule 1) an interval radius $\mu \geq 0$ is used, as shown in the bottom x axis of Figure 5.7; Rule 2) all the monetary parameters (i.e., $\tilde{L}_y, C_a$) are bounded in the interval $\left[\sigma^{nominal} \cdot (1-\mu), \sigma^{nominal} \cdot (1+\mu)\right]$, and the $\sigma^{nominal}$ are the nominal values of the attackers' parameters as given in Table 4.11 through Table 4.16. In experiment s1, all the probabilistic parameters (i.e., $\tilde{P}_i^Z, \tilde{P}_i^p, \tilde{p}_y$) are bounded in the interval $\left[\sigma^{nominal} \cdot (1-\mu), \sigma^{nominal} \cdot (1+\mu)\right] \cap [0,1]$, and the $\sigma^{nominal}$ are the nominal values of the attackers' parameters as given in Table 4.11 through Table 4.16. In experiment s2, all the probabilistic parameters (i.e., $\tilde{P}_i^Z, \tilde{P}_i^p, \tilde{p}_y$) are the same as the nominal values as given in Table 4.11 through Table 4.16.

Results shown in Figure 5.7 demonstrate that the increase of interval radius μ would result in a decrease of the defender's optimal payoff. When $\mu = 0.0$, which means no interval uncertainty exists, the defender could have a payoff equal to the payoff from the BSE. When $\mu \geq \mu^*$ (in experiment s1, $\mu^* = 0.12$ for IBGS and $\mu^* = 0.16$ for ICGS, while in s2, $\mu^* = 0.46$ for IBGS and $\mu^* = 0.50$ for ICGS), the defender's payoff could be as low as her MaxiMin payoff. This means that, in the Bayesian Stackelberg CPP game, if the defender could not effectively bound the attacker's parameters into a relatively narrow interval, then her information of the attacker is useless.

Figure 5.7 also shows that with the same interval radius, the ICGS solution could always bring the defender a higher payoff than the IBGS solution, supporting our third conclusion.

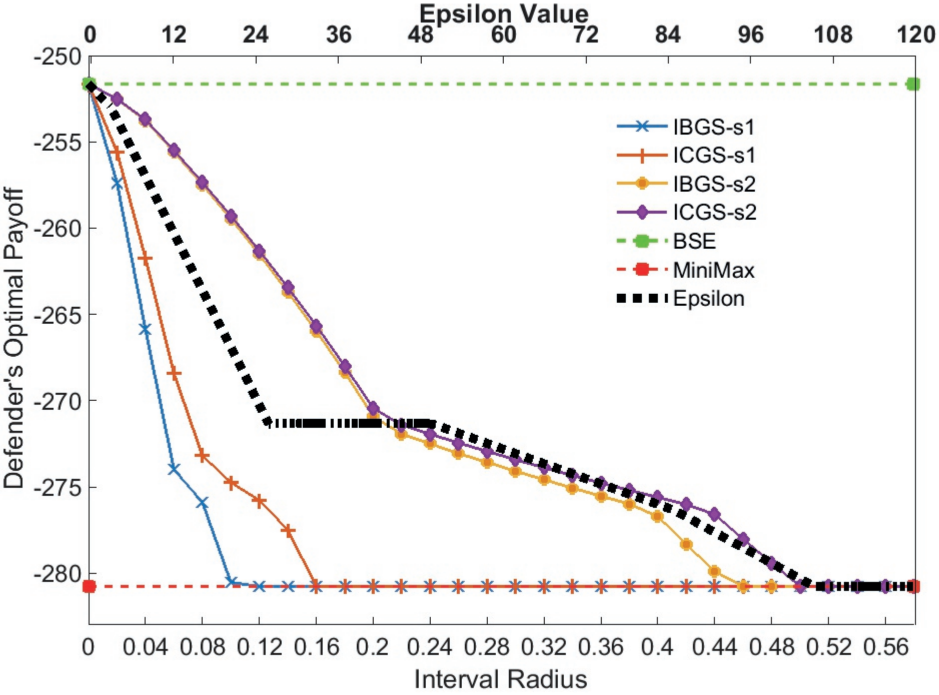


Figure 5. 7. Sensitivity analysis (of the epsilon value in the robust solution and of the interval radius in the interval game solution)

5.5 Conclusions

In this chapter, firstly, the interval CPP game is defined and two algorithms are proposed. In the interval CPP game, the defender knows an interval in which the attacker's parameters will be located, however she does not know what the exact values of the parameters are and how the parameters are distributed in the interval. The defender thus plays conservatively to make the best use of her knowledge. The two proposed algorithms are both based on Mixed Integer Linear Programming techniques. The interval bi-matrix game solver (IBGS) is applicable to any bi-matrix games with such distribution-free uncertainties, while the interval CPP game solver (ICGS), on the other hand, is proposed only for solving CPP games with interval inputs.

Theoretic study shows that the ICGS algorithm can bring the defender higher or equal payoff compared to the IBGS algorithm.

In industrial practice, it is difficult to obtain the exact numbers of the parameters that a CPP game needs. As also illustrated in section 4.5.1, inputs data from conventional security risk assessment methods are always associated with an interval, see for instance, Table 4.3 to 4.5. With the work in this chapter, the CPP game model is able to directly deal with the data from those conventional security risk assessment methods.

Secondly, bounded rational attackers in the chemical plant protection game are modelled in this chapter. The 'Epsilon-optimal attacker', the 'Monotonic optimal attacker', and the 'MiniMax attacker' are defined and integrated into the CPP game. Furthermore, algorithms for solving CPP games with these bounded rational attackers are developed.

Game theoretic models developed in the security domain are criticized for their requirement of quantitative data and their rationality assumption. With the enhancement presented in this chapter, the CPP game is ready to be implemented in industrial practice. Further research could be to compare the results obtained by the game theoretic model with the results following the application of the API SRA methodology in a real chemical plant, for example. Based on the game theoretic models, to develop a user-friendly software, which focuses on inputs and outputs of the model while integrating all the mathematic details as a black box, could be another promising next research step.

References

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

[2] Cox Jr LAT. Game theory and risk analysis. Risk Anal. 2009;29(8):1062-8.

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

[4] Zhang L, Reniers G. Applying a Bayesian Stackelberg game for securing a chemical plant. submitted to Journal of Loss Prevention in the Process Industries. 2017.

[5] Feng Q, Cai H, Chen Z. Using game theory to optimize the allocation of defensive resources on a city scale to protect chemical facilities against multiple types of attackers. Reliability Engineering & System Safety. 2017.

[6] Feng Q, Cai H, Chen Z, Zhao X, Chen Y. Using game theory to optimize allocation of defensive resources to protect multiple chemical facilities in a city against terrorist attacks. J Loss Prev Process Ind. 2016;43:614-28.

[7] Pavlova Y, Reniers G. A sequential-move game for enhancing safety and security cooperation within chemical clusters. J Hazard Mater. 2011;186(1):401-6.

[8] Reniers G, Cuypers S, Pavlova Y. A game-theory based Multi-plant Collaboration Model (MCM) for cross-plant prevention in a chemical cluster. J Hazard Mater. 2012;209-210:164-76.

[9] Reniers G, Soudan K. A game-theoretical approach for reciprocal security-related prevention investment decisions. Reliab Eng Syst Saf. 2010;95(1):1-9.

[10] Talarico L, Reniers G, Sörensen K, Springael J. MISTRAL: A game-theoretical model to allocate security measures in a multi-modal chemical transportation network with adaptive adversaries. Reliab Eng Syst Saf. 2015;138:105-14.

[11] Zhang L, Reniers G. Applying a Bayesian Stackelberg game for securing a chemical plant. Risk Anal. 2016.

[12] Zhang L, Reniers G. A Game-Theoretical Model to Improve Process Plant Protection from Terrorist Attacks. Risk analysis : an official publication of the Society for Risk Analysis. 2016.

[13] Pita J, Jain M, Tambe M, Ordóñez F, Kraus S. Robust solutions to Stackelberg games: Addressing bounded rationality and limited observations in human cognition. Artificial Intelligence. 2010;174(15):1142-71.

[14] Goeree JK, Holt CA, Palfrey TR. Regular quantal response equilibrium. Experimental Economics. 2005;8(4):347-67.

[15] McKelvey RD, Palfrey TR. Quantal response equilibria for normal form games. 1993.

[16] Nagel R. Unraveling in guessing games: An experimental study. The American Economic Review. 1995;85(5):1313-26.

[17] Jiang AX, Nguyen TH, Tambe M, Procaccia AD, editors. Monotonic maximin: A robust stackelberg solution against boundedly rational followers. International Conference on Decision and Game Theory for Security; 2013: Springer.

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

[19] Dantzig G. Linear programming and extensions: Princeton university press; 2016.

[20] Ben-Tal A, Nemirovski A. Robust solutions of uncertain linear programs. Operations research letters. 1999;25(1):1-13.

[21] Kiekintveld C, Marecki J, Tambe M, editors. Approximation methods for infinite Bayesian Stackelberg games: Modeling distributional payoff uncertainty. The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3; 2011: International Foundation for Autonomous Agents and Multiagent Systems.

[22] Kiekintveld C, Islam T, Kreinovich V, editors. Security games with interval uncertainty. Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems; 2013: International Foundation for Autonomous Agents and Multiagent Systems.

[23] Nikoofal ME, Zhuang J. Robust allocation of a defensive budget considering an attacker's private information. Risk Anal. 2012;32(5):930-43.

[24] Zhang L, Reniers G, Chen B, Qiu X. A Chemical Plant Protection Game Incorporating Boundedly Raitonal Attackers and Distribution-free Uncertainties Submitted to Chemial Engineering Science. 2017.

[25] Guikema SD. Game theory models of intelligent actors in reliability analysis: An overview of the state of the art. Game theoretic risk analysis of security threats: Springer; 2009. p. 13-31.

[26] Yang R, Ordonez F, Tambe M, editors. Computing optimal strategy against quantal response in security games. Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2; 2012: International Foundation for Autonomous Agents and Multiagent Systems.

[27] Rothschild C, McLay L, Guikema S. Adversarial risk analysis with incomplete information: A level-k approach. Risk Anal. 2012;32(7):1219-31.

[28] 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.

[29] Nguyen TH, Yang R, Azaria A, Kraus S, Tambe M, editors. Analyzing the Effectiveness of Adversary Modeling in Security Games. AAAI; 2013.

6

MULTI-PLANT PROTECTION: A GAME THEORETICAL MODEL FOR IMPROVING THE SCHEDULING OF CHEMICAL CLUSTER PATROLLING

Chemical clusters can be attractive targets for terrorism, due to the extreme importance of them for society as well as due to the existence of dangerous materials. Patrolling can be 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 therefore proposed in this chapter. 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 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.

This chapter has been published as a journal paper, being:

Zhang L., Reniers G., Chen B., & Qiu X. (2018). CCP game: A game theoretical model for improving the scheduling of chemical cluster patrolling. Accepted for publication in Reliability Engineering and System Safety.

6.1 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 neighbouring 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 practice, some patrollers follow a fixed patrolling route (i.e., the same patrolling route is used for 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 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.

This chapter 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 better secured, by using a game theoretical approach. The remainder of the chapter is organized as follows: Section 6.2 briefly introduces how patrolling is organized in chemical clusters. Section 6.3 proposes the chemical cluster patrolling game. An illustrative case study is investigated in section 6.4. Section 6.5 discusses two uncertainties in the CCP game: the implementation errors and observation errors. Conclusions are drawn in section 6.6.

6.2 Patrolling in chemical clusters

6.2.1 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 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 evidently 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.

6.2.2 Formulating the research question

6.2.2.1. 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.

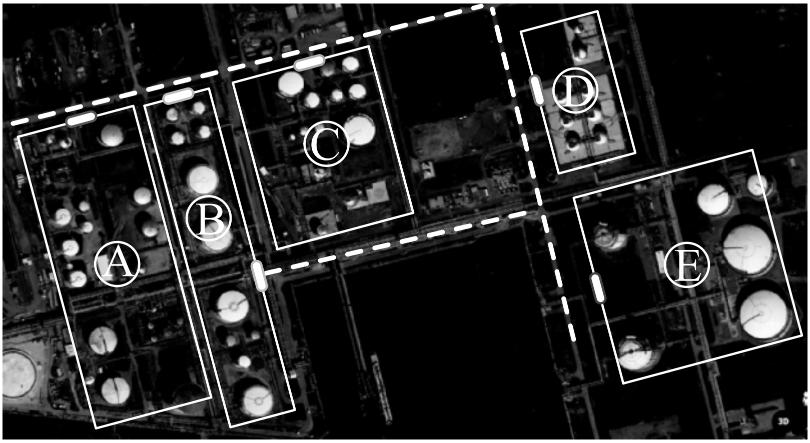


Figure 6. 1. Layout of a chemical park in Antwerp port

For example, Figure 6.1 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 6.2 shows the graph model of the cluster shown in Figure 6.1. As we may notice, plants 'A', 'C', 'D', and 'E' in Figure 6.1 are modelled as a node (with the same name) in Figure 6.2. The cross point of the vehicle road between plant 'D' and 'E' in Figure 6.1 is also denoted as a node in Figure 6.2 (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 6.2 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 node 'B1' and 'B2' because these two nodes belong to the same plant and hence should be connected.

Based on the graphic model, the patrolling scenario in section 6.2.1 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.

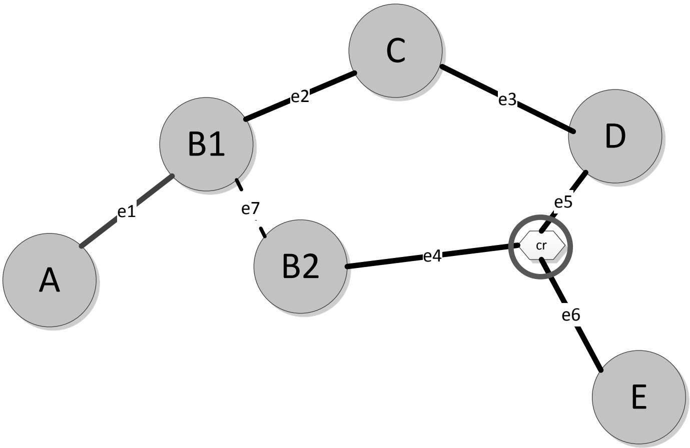


Figure 6. 2. Graphic modelling of the chemical park

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 chapter, 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 6.1 further demonstrates all the notations used in this chapter.

Table 6. 1. Definitions of Notations

Notation	Definition	Type*
$\boldsymbol{G(V,E)}$	The graphic model of the chemical cluster, defined in section 6.2.2.1.	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 6.2.2.2.	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

$\boldsymbol{f_{cpp}}, \boldsymbol{\tilde{f}_{cpp}}$	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 6.3.4.	IN
$\boldsymbol{\tau_r}$	Probability that the patroller would be in the overlap situation r, defined in section 6.3.4.	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 6.2). 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 6.2). 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 6.2, 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 6.2, 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 6.1 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 6.1 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 6.2. t_i^d represents the driving time of edge 'ei' in Figure 6.2. 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 6. 2. Superior connection matrix for Figure 6.2 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

6.2.2.2. 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 6.3 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 6.3 shows the patrolling graph pG for the chemical cluster shown in Figure 6.1, with the data in Table 6.2 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 6. 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 6.3, the x axis denotes the time dimension, while the y axis represents the different nodes in Figure 6.2. Therefore, any coordinates in Figure 6.3 can be a possible node for pG. As we may see, node 1 (at the left-hand side of the figure) in Figure 6.3 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 6.3, 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 $(pe^1, pe^2, \ldots, pe^{len})$ 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} $(i = 1,2, \ldots, len - 1)$ 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 6.3 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 6.3, 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 6.3). For instance, in Figure 6.3, 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 $T = 220$.

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 6.3 should actually be an overlap.

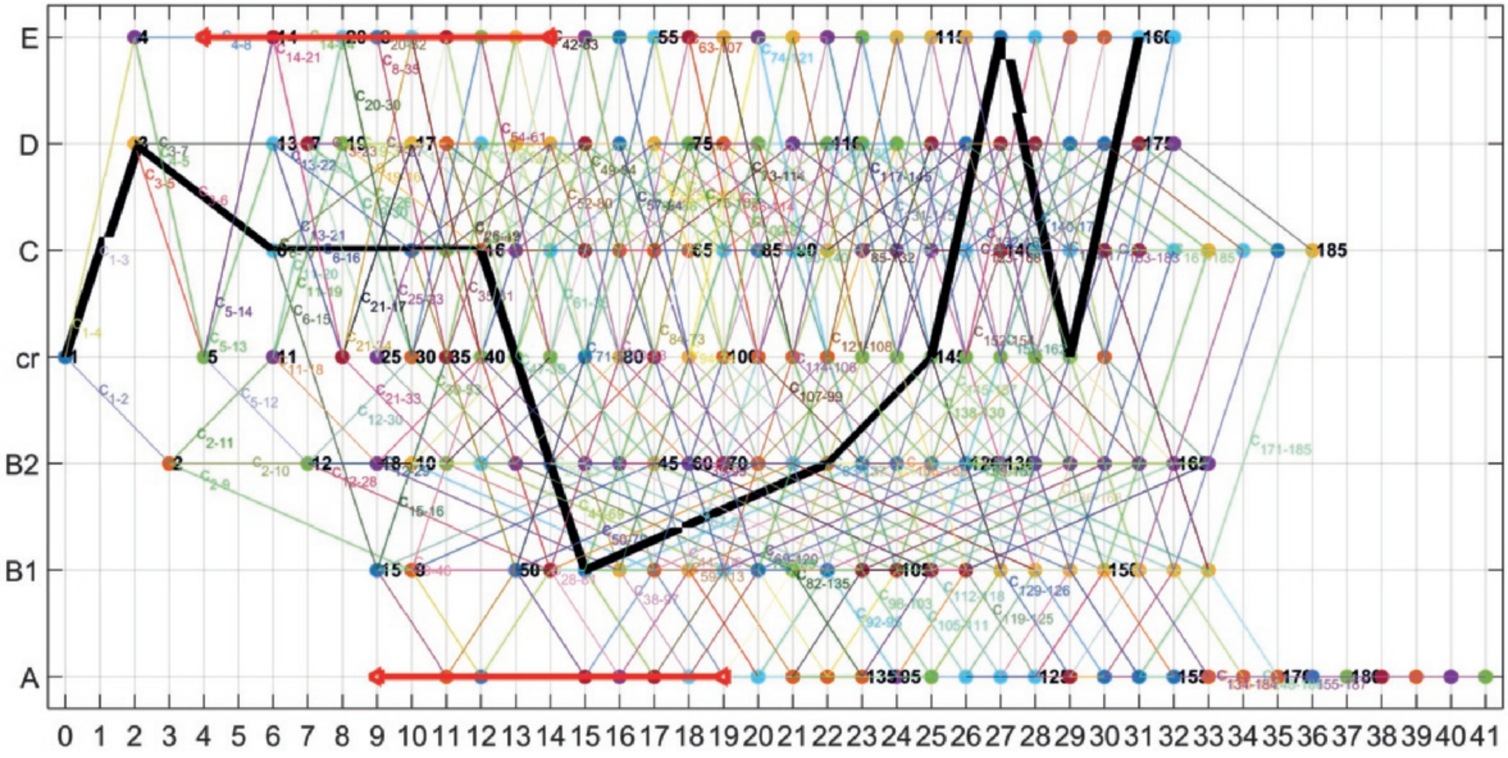


Figure 6. 3. Patrolling Graph of the illustrative example

6.2.2.3. 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 6.3.2, 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 6.3.4. 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.

6.3 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.

6.3.1 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.

6.3.2 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 (6.1).

$s_a = (t, i, k_i)$ ··(6.1)

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 6.3 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 (6.1) implies that the attacker would only attack one plant. The number of the attacker’s pure strategies can be calculated by Formula (6.2). 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.

$$m = n \cdot T \cdot Sce \cdots\cdots (6.2)$$

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 6.2.2, 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 6.2.2), then the patroller’s strategy is the combination of these probabilistic numbers. A mathematic formulation of the defender’s strategy is shown in Formula (6.3).

$$s_d = \prod_{(s,e)\in pE} c_{s-e} \cdots\cdots (6.3)$$

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 6.3 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 (6.4) and (6.5) illustrate the abovementioned two properties.

$$sP_{pv} = \sum_{in\in\{s\in pV|(s,pv)\in pE\}} c_{in-pv} = \sum_{out\in\{e\in pV|(pv,e)\in pE\}} c_{pv-out} \cdots\cdots (6.4)$$

$$\sum_{out\in\{e\in pV|(root,e)\in pE\}} c_{root-out} = 1 \cdots\cdots (6.5)$$

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 (6.6). For instance, if a purely randomized patrolling strategy would be implemented on the patrolling graph shown in Figure 6.3, 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 6.5 in the case study section also illustrates how Formula (6.6) works.

$$cP_{pv}^{out} = \frac{c_{pv-out}}{sP_{pv}}, \text{ for all } out \in \{v \in pV | (pv, v) \in pE\} \cdots\cdots (6.6)$$

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 (6.7) and (6.8) 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.

$$u_d = R^d \cdot f - L^d \cdot (1 - f) \cdots\cdots (6.7)$$

$$u_a = G^a \cdot \left(1 - \tilde{f}\right) - P^a \cdot \tilde{f} \cdots\cdots (6.8)$$

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 (6.3)).

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). 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 (6.9):

$$f = 1 - (1 - f_{cpp}) \cdot (1 - f_p) \cdots\cdots (6.9)$$

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 (6.10), in which st denotes the start time that the patroller stays in plant i. There are two situations of the exact overlap period.

$$Overlap = [max\{t, st\}, min\{t + k_i, st + t_i^p\}] \cdots\cdots (6.10)$$

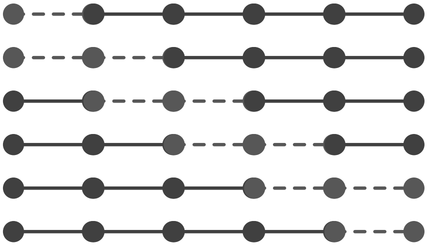


Figure 6. 4. An illustrative figure of the overlap situation

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 6.4 shows an example of the overlap situations with $k_i = 5$, $t_i^p = 2$. In Figure 6.4, 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.

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 $r = 1,2, \ldots, t_i^p + k_i - 1$ in situation 1 and $r = 1,2, \ldots, k_i + k_i - 1$ 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 (6.11). σ_r are user inputs and should be provided by security experts.

$$f_p = \sum_r \sigma_r \cdot \tau_r. \cdots\cdots (6.11)$$

Table 6.4 shows how to calculate τ_r, under the condition of an attacker strategy (t, i, k_i) and a defender strategy $\vec{c}$.

Table 6. 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 6.1 and 6.2, 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 6.5 and Table 6.7 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 6.3, 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:

$$f = \left[\sum_r \left(1 - f_{cpp}\right) \cdot \sigma_r \cdot Coe_r\,, f_{cpp}\right] \cdot [\vec{c}, 1]^T \cdots\cdots (6.12)$$

$$u_d = [(R^d + L^d) \cdot \left(\sum_r \left(1 - f_{cpp}\right) \cdot \sigma_r \cdot Coe_r\right), (R^d + L^d) \cdot f_{cpp} - L^d] \cdot [\vec{c}, 1]^T \cdots\cdots (6.13)$$

$$u_a = [-(G^a + P^a) \cdot \left(\sum_r \left(1 - \tilde{f}_{cpp}\right) \cdot \sigma_r \cdot Coe_r\right), G^a - (G^a + P^a) \cdot \tilde{f}_{cpp}] \cdot [\vec{c}, 1]^T \cdots (6.14)$$

6.3.4 Solutions for the game

6.3.4.1 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:

$$(t^*, i^*, k_i^*) = \text{argmax}_{(t,i,k_i) \in S_a}\{u_a(\vec{c}, (t, i, k_i))\} \cdots\cdots (6.15)$$

$$\overrightarrow{c^*} = \text{argmax}_{\vec{c} \in S_d}\{u_d(\vec{c}, (t^*, i^*, k_i^*))\} \cdots\cdots (6.16)$$

Formula (6.15) 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 (6.16) 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 6.2.2.3), the attacker has a finite number of strategies. Moreover, Formulas (6.13) and (6.14) 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 [37] can be introduced to compute the Stackelberg equilibrium for the CCP game, as shown in Table 6.5.

In the linear programming step, the defender is solving a linear programming problem. The cost function of the linear programming is Formula (6.18) and the constraints are Formulas (6.17), (6.4) and (6.5). 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.

Table 6. 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:
    $u_a(t^{\#}, i^{\#}, k_i^{\#}, \vec{c}) \geq u_a(t, i, k_i, \vec{c}), \quad \forall (t, i, k_i) \in S_a$        (6.17)
    The defender would then aims at:
    $Pof_d\left(t^{\#}, i^{\#}, k_i^{\#}, \overrightarrow{c^{\#}}\right) = \max_{\vec{c} \in S_d} u_d(t^{\#}, i^{\#}, k_i^{\#}, \vec{c})$        (6.18)
    ○ Summary
    The Stackelberg equilibrium
    $(\overrightarrow{c^{*}}, (t^{*}, i^{*}, k_i^{*})) = arg \max_{(t^{\#}, i^{\#}, k_i^{\#}) \in S_a} Pof_d\left(t^{\#}, i^{\#}, k_i^{\#}, \overrightarrow{c^{\#}}\right).$
```

The Stackelberg equilibrium calculated by the MultiLPs algorithm is a Strong Stackelberg Equilibrium [38], and it is therefore based on the "breaking-tie" assumption[1]. 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 (6.17) to be Formula (6.19), 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 [38].

$$u_a(t^{*}, i^{*}, k_i^{*}, \vec{c}) \geq \alpha + u_a(t, i, k_i, \vec{c}), \quad \forall (t, i, k_i) \in S_a \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \text{(6.19)}$$

6.3.4.2 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 6.1) of the game. In security practice, the patroller may obtain some of these parameters by using conventional

[1] 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).

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. [39], we assume that the patroller can obtain an interval of these two parameters and how these two parameters distribute in the interval 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 (6.20) and (6.21) respectively. Note that Formula (6.8) demonstrates that u_a is monotonically increasing on G^a and monotonically decreasing on f. Formula (6.9) demonstrates that f is monotonically increasing on f_{cpp}.

$$u_a^{min} = [-(G^{a\_min} + P^a) \cdot (\sum_r (1 - \tilde{f}_{cpp}^{max}) \cdot \sigma_r \cdot Coe_r), G^{a\_min} - (G^{a\_min} + P^a) \cdot \tilde{f}_{cpp}^{max}] \cdot [\vec{c}, 1]^T \quad (6.20)$$

$$u_a^{max} = [-(G^{a\_max} + P^a) \cdot (\sum_r (1 - \tilde{f}_{cpp}^{min}) \cdot \sigma_r \cdot Coe_r), G^{a\_max} - (G^{a\_max} + P^a) \cdot \tilde{f}_{cpp}^{min}] \cdot [\vec{c}, 1]^T \quad (6.21)$$

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 (6.22) illustrates an algorithm for calculating the patroller's robust solution considering her distribution-free uncertainties on the attacker's parameters. In Formula (6.22), 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.

$$\underset{c,q,R,\gamma}{\text{maximize}}\,\gamma$$
$$s.t.\begin{cases} c1. & NstFlwLef \cdot c = NstFlwRgt \\ c2. & R = u_a^{\min}(J,c) \\ c3. & R \geq u_a^{\min}(j,c), \forall j \in \{1,2,...,\mathrm{m}\} \\ c4. & -q_j \cdot \Gamma \leq R - u_a^{\max}(j,c) \leq (1-\mathrm{q}_j) \cdot \Gamma, \forall j \in \{1,2,...,\mathrm{m}\} - Plt_J \\ c5. & -q_j \cdot \Gamma \leq f_p(j,c) - f_p(J,c) \leq (1-\mathrm{q}_j) \cdot \Gamma, \forall j \in Plt_J \\ c6. & (1-q_j) \cdot \Gamma + u_d(j,c) \geq \gamma, \forall j \in \{1,2,...,\mathrm{m}\} \\ c7. & q_j \in \{0,1\}, c_i \in [0,1], \forall j \in \{1,2,...,\mathrm{m}\}, \forall i \in \{1,2,...,n\} \end{cases} \quad (6.22)$$

Constraint $c1$ reflects the features of the patroller's strategy c, as explained in Formula (6.4) 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 (6.22), 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 (6.22) 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 (6.22) 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.

6.4 Case study

6.4.1 Case study setting

The layout of the cluster, the graph model, and the patrolling graph model of the case study are given in Figure 6.1 through 6.3. 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 6.2. 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 6.6 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 6.4) 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. 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.

6.4.2 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 $t \in \{0, 1, 2, \dots, 29\}$). The patroller has 435 possible actions that she can take, shown as edges in Figure 6.3 and therefore the patroller's strategy can be represented as a vector of 435 entries.

According to Formulas (6.13), (6.14), (6.20), and (6.21), 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.

6.4.3 CCP Game results

6.4.3.1 Stackelberg equilibrium

Figure 6.5 shows the modified Stackelberg Equilibrium (mSE) of the game developed for the case study, calculated by the MultiLPs algorithm shown in Table 6.5 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 denotes 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.[2] 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 (6.6). For instance, the probability that the patroller would arrive at the red node $(6, 'C')$ in Figure 6.5 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 6.5 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 probability of being taken (i.e., $c > 0$) and would have overlap with the attacker's best response strategy. Table 6.7 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 6.3) of the action. The c column shows the probability that the actions would be

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

taken in the mSE, and these numbers are also shown in Figure 6.5. 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 $0.05 \times (13-9) = 0.20$.

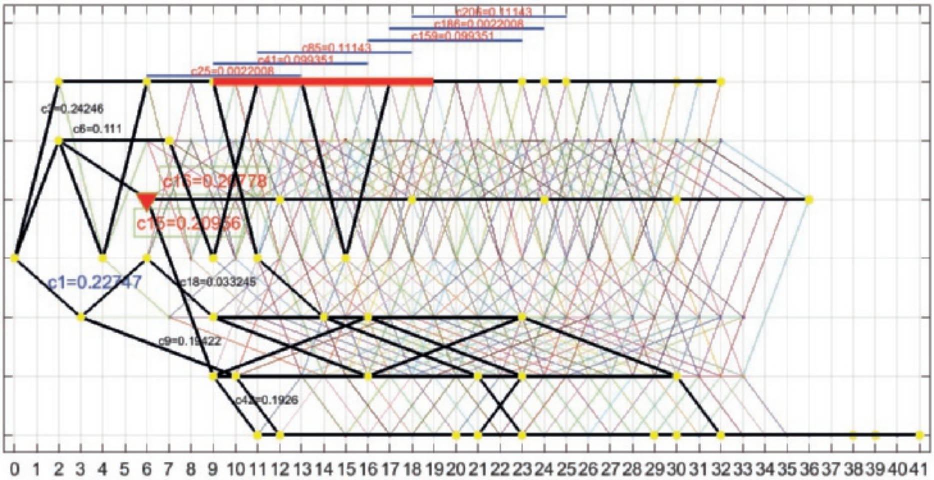


Figure 6. 5. The optimal patrolling strategy and the attacker’s best response

Based on the results in Table 6.7, recalling Formula (6.11) and the τ_r calculation algorithm, we have that:

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

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

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

Table 6. 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 6.3, 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 $1/3$.

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 6. 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

Table 6.8 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 6.8, 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 (6.4) and (6.5), represents a fixed patrolling route. The bold route shown in Figure 6.6 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 6.6 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 6.6, 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.

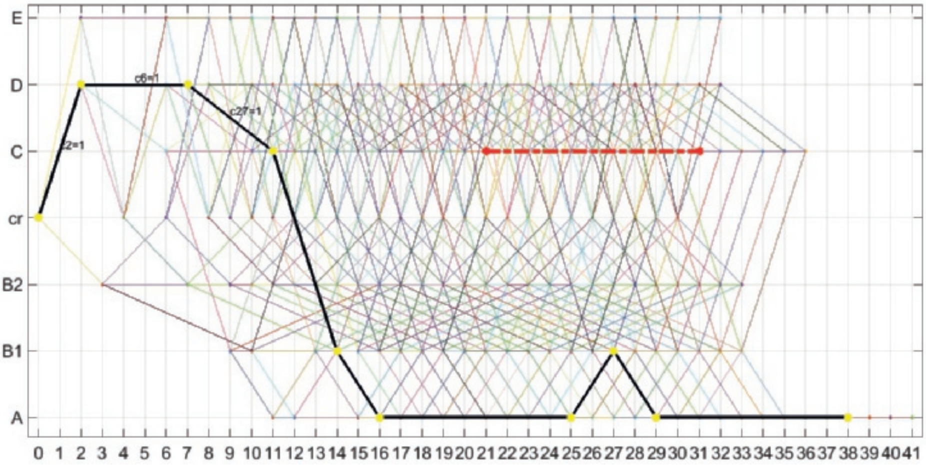


Figure 6. 6. The patroller's optimal fixed patrolling route and the attacker's best response

6.4.3.2 Robust equilibrium

Figure 6.7 shows the robust solution of the Interval Chemical Cluster Patrolling game, based on the input data from Table 6.6. Notations of Figure 6.7 are the same as defined in Figure 6.5. 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 6.7. Furthermore we have:

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

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

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

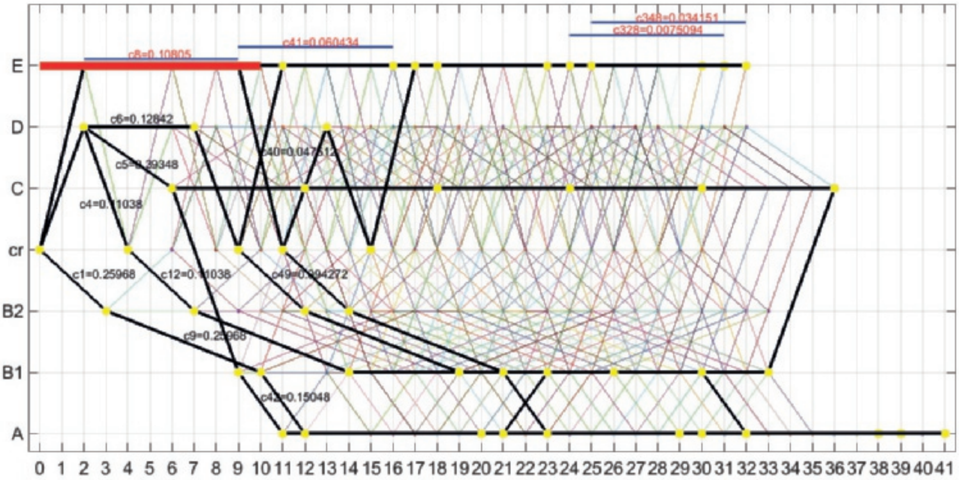


Figure 6. 7. Robust solution of the interval CCP game

Figure 6.8 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 6.8, 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 (6.22), instead of by the payoff range constraint (i.e., Constraint c4 in Formula (6.22)).

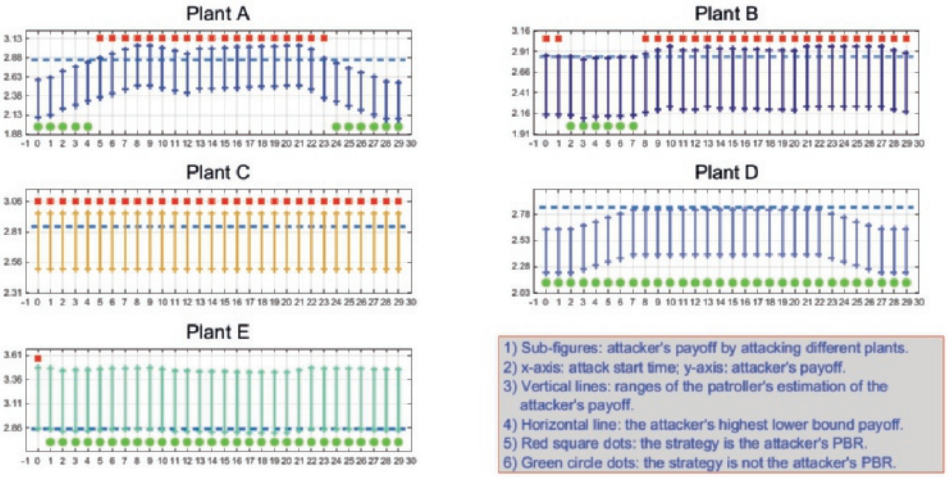


Figure 6. 8. Attacker payoff information of the robust solution of the Interval CCP game (PBR: possible best response)

6.5 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. (40) can be employed. However, the algorithm in Nguyen et al. (40) 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.

6.6 Conclusions

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 chapter. 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 (41).

References

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

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

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

[40] 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.

[41] 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.

7

PROTECTION BETWEEN PLANTS: OPTIMAL PATROL SCHEDULING OF HAZARDOUS PIPELINES USING GAME THEORY

An approach based on game theory is proposed to schedule security patrolling for a pipeline system. The developed method proposes numbers of patrolling paths according to the risk of security incidents on the pipeline system to allow the patrol covering high-risk segments more than low-risk segments. Patrolling of the pipeline system was modelled mathematically, based on time and distance discretization. Patrolling of a single unit that can be a motorcycle, a vehicle, a drone or an helicopter was considered, depending on its velocity. The overall approach also examines the presence of security countermeasures on a pipeline system, and their effects on the patrolling schedule. The application of the method is explained by an illustrative case study.

This chapter has been published in the following paper:

Rezazadeh, A., Zhang, L., Reniers, G., Khakzad, N., & Cozzani, V. (2017). Optimal patrol scheduling of hazardous pipelines using game theory. Process Safety and Environmental Protection, 109, 242-256.

7.1 Introduction

We have developed game theoretical models for protecting chemical sites and chemical clusters. In this chapter, we apply the Bayesian Stackelberg game for scheduling the patrolling on a pipeline system. This game theoretical model is called **P**ipeline **P**atrolling **G**ame (**PPG**). PPG is a Bayesian Stackelberg game in which the defender is the leader whereas the attacker is the follower. In this methodology, there is only one type of the defender, but the attacker may be of different types.

In recent years, several successful attacks on oil & gas pipelines accured, [1] demonstrating both the attractiveness of pipelines to terrorists and the terrorists' capabilities of implementing an attack to pipelines. Despite fixed countermeasures, [2] patrolling is also scheduled for protecting oil & gas pipelines. However, the patrolling of pipelines are different from the patrolling of multiple plants. Therefore we can not directly use the CCP game to optimize the patrolling of pipelines. In the CCP game, the patroller can pass a plant (a node in the graph) without patrolling it while in the pipeline patrolling, the patroller is not able to jump from one segment (see segmentation in section 7.3.1) to another without patrolling the segments between them (see details in section 7.3.4).

The PPG has been built based on a credible security risk assessment. Each part of the pipeline system, due to its location, design and operation characteristics, can be more vulnerable when compared to other parts. These different parts can have a different attraction, or perceived value, from the attacker and the defender points of view. Therefore a risk analysis framework is required to systematically examine the components and characteristics of the risk of different pipeline segments and presenting the results in a rank ordering form, to build the utility functions that estimate the gain or loss of the players.

Reniers and Dullaert [3] have made a so-called TePiTri method to determine relative terrorist-related security risk levels of a pipeline transportation system. In the TePiTri method a likelihood grade and a consequence grade are determined by discretizing the pipeline route through an analysis procedure, and these grades define the security risk level of a segment. One of the determining factors of security risk is threat assessment which is well planned by Reniers et al.[4] in the TARP model for the chemical industry. TARP lists different security threats and proposes a decision flowchart for the assessments and a guideline for revising this threat assessment periodically or when needed. CCPS [5] explains qualitative and semi-quantitative approaches for evaluating safety and security risks using either risk indexing or risk ranking matrixes. CCPS [6] and the American Petroleum Institute [7] propose a Security Risk Assessment method (SRA) which defines the security risk as a function of Vulnerability, Attractiveness, and Consequences. Also, they define a procedure by which the security risk can be found from a ranking matrix through asset-based and scenario-based approaches.

Among the abovementioned security risk assessments methods, the API SRA method [7], providing a risk ranking baseline, will be used for the security risk assessment in the PPG model.

In this chapter, section 7.2 discusses the players of the game. Modeling of the patrolling is treated in section 7.3. Section 7.4 explains security risk assessment of PPG. For solving the proposed security game the PPG algorithm is introduced in section 7.5. In section 7.6 we will explain an illustrative case study in addition to its results and discussion. Finally, in section 7.7 conclusions are provided.

7.2 Modelling the players of the game

Today security forces are faced with different types of adversaries with various characteristics. Each of them has its specific intention and capabilities. They may plan different malicious acts to achieve their separate goals. Accordingly, when the security of pipelines or any other facility is discussed, at first, the type of facing an attacker should be clarified. Also, it is important to categorize them and evaluate their intentions, capabilities to attack and the level in which they are active in a region.

For developing the PPG, only one patrolling unit that can be a vehicle, helicopter or drone and only one intruder who can be any type of attacker are considered. The pipeline route will be separated into some segments to model patrolling paths. In this game, the main assumption is that, if the patrol is present at a pipeline segment location, this will guarantee to stop the attacker at that location, and the attack cannot occur successfully. Moreover, it is assumed that if the security barriers of the pipeline system detect any malicious act, the defender again can stop the attack or prevent it from happening. The attacker has complete information about the probability of which the patrol may be present at a segment, and they are fully rational, but they do not know the exact patrolling plan of security forces.

The PPG game is a 2-player game, namely, the patrol and the attacker. Since “defender” is a general term who can be a security department of a pipeline company planning patrolling paths for a patrol, indeed the front player of the attacker is the patrol; from now on, we call leader or defender the patrol. Therefore the players of the PPG game are the patrol and the attacker. To reduce ambiguity, the patrol is considered as a female and the attacker as a male so that in the remainder of the chapter, the pronoun “he” will be used for an attacker and “she” for a patrol.

7.2.1 Categorization of attacker types

We are going to categorize the different types of attackers systematically. Therefore we applied the API SRA method [7] categorizing attackers to international and domestic terrorists ($k = 1$), criminals ($k = 2$), disgruntled personnel ($k = 3$), or extreme activists ($k = 4$). These different adversaries pose different threats to a pipeline system. In this chapter we are going to carry out a security risk analysis for the different types of adversaries separately.

7.2.2 Scenario identification

For evaluating the consequence of a security incident, a scenario analysis should be applied. To do so, experts of the security department of a pipeline company should list all the expected scenarios from the adversaries, and then categorize them according to the type of the attacker. In Table 7.1 some of the common scenarios are illustrated.

Table 7. 1. Typical scenarios for various adversaries

	Terrorist	Criminal	Disgruntled insider	Activist
Scenario	• Causing an explosion	• Operation disruption	• Operation disruption	• Operation disruption
	• Release of chemical	• Release of chemical	• Release of chemical	• Damage to properties
	• Theft	• Theft	• Theft	• -
	• -	• -	• Damage to properties	• -

If more than one scenario is possible, the security department can choose either the worst case scenario or the most credible scenario for each type of the attacker, based on their policies, and consider it for further processing (consequence assessments). Thus each type of attacker is paired with a particular scenario. As an illustration, the consequence of a terrorist attack can be a pipeline explosion, while the consequence of a criminal or a disgruntled insider may be a theft or an operation disruption, and the consequence of activist interference can be damages to properties resulting in operation disruption.

7.2.3 Threat assessment

According to CCPS [(6)], threats can come from these three sources:

- Internal
- External
- Collusion (Internal and External)

Threat Acts may be perpetrated by insiders, outsiders or a combination of the two. Insiders are those personnel that has internal knowledge routine and unescorted access to areas where outsiders are not allowed without an escort. Collusion between the two may be the result of monetary incentive, ideological sympathy, or coercion.

The threat can also be defined as the intention and capability of a threat to undertake actions that would be detrimental to the pipeline system. Threat assessment is an important part of a PPG security assessment, especially in light of today's international terrorism. There is a need to determine the threats facing the pipeline system properly in building the present security game. The API SRA method [(7)] provides an approach for assessing threat levels of each types of threat, as shown in Table 4.3 in Chapter 4.

In fact in a Bayesian Stackelberg game, the probability distribution of different types of a player should be known. Consequently, we are going to define a conditional probability for each type of the attacker to show how likely this type of attacker will contribute in a security incident in comparison to the others. Thus for solving the PPG, the probability distribution over four types of the attacker is identified as ρ^k, which represents the conditional probability if the attacker has type k. The ρ^k can be calculated also by Formula (4.2) in Chapter 4, in which the TL should be set as $TL = \{domestic\ terrorists, criminals, disgruntled\ personnel, exterme\ activists\}$ and the threat level of each type of threat should be assessed according to the criteria shown in Table 4.3 in Chapter 4.

7.2.4 Identify player types

Concerning the API method [(7)], we classified the expected consequences of security incidents to:

I. Fatalities and injuries
II. Environmental impacts
III. Property damage
IV. Business interruption
V. Damage to reputation or negative publicity

These types of consequences will provide us a framework for categorizing attacker types. Since each type of adversary has its intention and follows distinguished objectives, each kind of consequences has a specific value for them. In other word, based on the characterization of attacker types, the same consequences may have a different value for them. Therefore, in the PPG security risk assessment we identify these various perceived values by a discrete set of weighing factors (indicated by WF)) for each kind of consequences, it means for different attacker types, the experts of the security department should define a distinct set of WF for separate kinds of consequences of a security incident (expert based). It is clear that, these kinds of consequences will have a different contribution to the PPG security risk assessment.

In this study, disparate types of adversaries are presented by distinct sets of WFs associated with each type of consequences. Table 7.2 shows example sets of WFs for characterizing the attacker in PPG. Similarly, the consequences have different values for the patrol. So, in Table 7.2 also the patrol is characterized by a set of WFs indicating her perceived values of the consequences.

Table 7. 2. Set of WF identifying player types

Kind of consequence	WF				
	Terrorist	Criminal	Disgruntled insider	Activist	patrol
Fatalities and injuries	3	0	0	0	3
Environmental impacts	1	0	1	3	1
Property damage	2	3	1	2	2
Business interruption	2	0	2	2	1
Damage to reputation	3	0	3	3	2

7.3 Strategy modelling

PPG player's strategies are modelled in this section. First of all, the pipeline system being patrolled as well as the patrolling time are segmented in sub-section 7.3.1. Based on this segmentation, the patrolling routes are defined in sub-section 7.3.2. Finally in sub-section 7.3.3 and 7.3.4 the game strategies for the attacker and the patrol are defined.

7.3.1 Segmentation

In PPG we discretize the time and route into intervals that are called ***Time Segment*** and ***Pipeline Segment***, respectively. To do this, first of all, we divide the time into equal segments. Then the pipeline route is discretized according to equal time intervals into various *Pipeline Segments*.

In the first step of segmentation, the discretization of time is performed according to three factors: (i) the speed of patrol, (ii) the length of the pipeline, and (iii) the length of the time of patrolling. Then a number of time intervals and their duration are estimated. The duration of time intervals should be equal.

In the second step, the pipeline route is discretized to *Pipeline Segments*. The length of each *Pipeline Segment* is not necessarily equal to each other.

Division of the pipeline route is based on the main assumption of PPG. For this purpose, these segments should be thoroughly visible for the patrol as long as she is present in that segment. In other words, as she enters the *Pipeline Segment* and until she leaves, she should be able to detect any movement from the attacker in that segment. Therefore the two determining factors for pipeline route segmentation are natural or man-made visual obstacles.

The *Time Segments* are presented with nodes indicated by j. There will be two nodes, one for the beginning and the other for the end of the *Time Segment*. For example, in case of 20

Time Segments, there will be 21 nodes from $j = 0$ to $j = 21$. In this description, the nth *Time Segment* is from node $j = n - 1$ to node $j = n$. (See Figure 7.1, x axis)

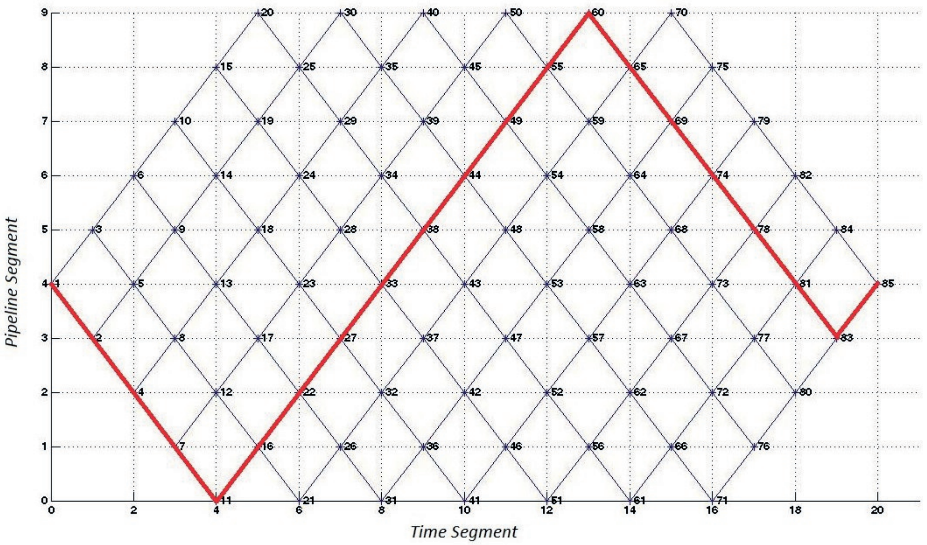


Figure 7. 1. Graphical model of patrolling path

The PPG is designed to schedule patrolling in day or night separately. Thus the game should be run one time to schedule the patrolling during the day and one time to schedule it in the night. Also, the relatively lower visibility at night is a determining factor that changes the length of segments.

In PPG, the *Pipeline Segment*s are determined by nodes. For each *Pipeline Segment,* we define two nodes at both ends. So two adjacent *Pipeline Segments* have one shared node that is the end of the first segment and the beginning of the second one. These nodes are indicated by i and if we have 9 *Pipeline Segment*s, there will be 10 nodes from $i = 0$ to $\text{i} = 9$. Consequently the nth *Pipeline Segment* is between node $i = n - 1$ and node $i = n$ (See Figure 7.1, the y axis).

Subsequently, a Length Examination test should be performed to identify whether or not the length of *Time Segment* is enough to pass and inspect the *Pipeline Segments* completely. If not, the algorithm goes back to step one to modify the time segmentation and repeat the iteration to reach an applicable segmentation in which all the *Pipeline Segments* can be inspected thoroughly with dedicated *Time Segment*.

This segmentation procedure is presented schematically in Figure 7.2. For example, assume that the security department of a pipeline company intends to schedule the patrolling on a 5 km sweet gas pipeline between city A and B in a day shift of 1.8hr by one patrolling unit, which has an average speed of about 20km/hr. The time can be divided into 20 *Time Segments*, each of about 6 min, and 9 *Pipeline Segments* are defined on this pipeline route. Considering the accessibility of *Pipeline Segment*s and the obstacles that limit the visibility, these *Pipeline Segments* have a lenght between 0.5 to 0.8 km.

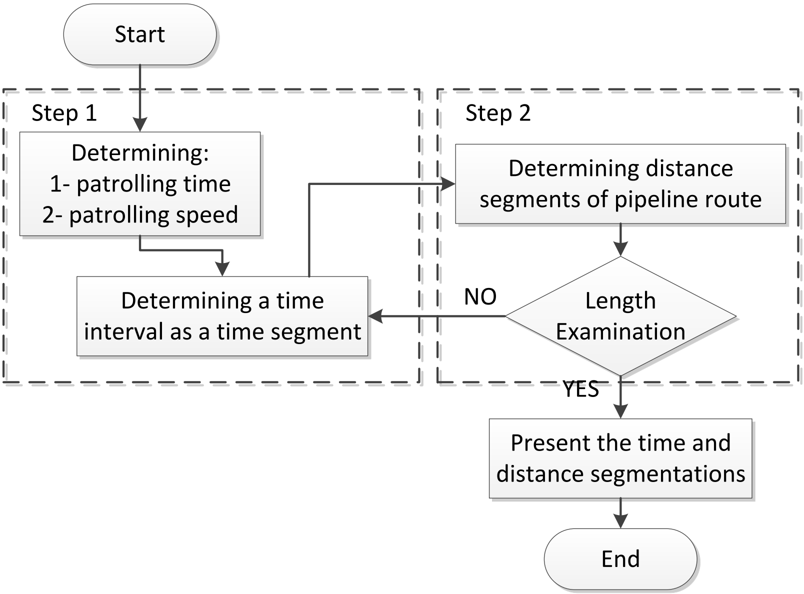

Start
Step 1
Determining:
1- patrolling time
2- patrolling speed
Determining a time interval as a time segment
Step 2
Determining distance segments of pipeline route
NO
Length Examination
YES
Present the time and distance segmentations
End

Figure 7. 2. Flowchart of segmentation procedure

7.3.2 Route identification

Following previous subsection, there is a fixed number of *Route* and *Time Segment*s. The patrol starts from *Time Segment* one and as time goes on she passes all *Time Segments* till the end of her duty. In each *Time Segment,* the patrol stays in one *Pipeline Segment,* and she inspects that segment thoroughly. As time passes from one *Time Segment* to the next, the patrol should decide where to go for the next time interval. Since each *Pipeline Segment* is shown with two nodes at the both ends, at the end of each *Time Segment* the patrol will be at one of these two nodes. Then they have two options: to come back to the previous node which means stay in their current *Pipeline Segment* or go to the next node to inspect the next *Pipeline Segment*. In this patrolling schedule, as time is passing, *Time Segment*s proceed; and the patrol can go forward, backward or stay in the same segment.

The temporal and spatial starting points are arbitrary, but the entire patrolling should take place during the day or night. Because the visibility in night and day can be different, the PPG should schedule the patrolling for day and night separately. Also, the weather condition and thus visibility variation can result in different segmentation: the PPG will have a separate schedule in this case.

The patrolling schedule can be represented graphically. For instance, following the example of section 7.3.1, a Patrolling path can be defined by the bold red colour line in Figure 7.1. In that illustration path, patrol started inspecting the pipeline system from the end of *Pipeline Segment* 4 and then she went backward to the beginning of the pipeline route. After that, she went straight to the end of the route, *Pipeline Segment* 9. Then she came back to *Pipeline Segment* 4 and inspected it in two *Time Segments* till the patrolling time ended. As you see, she finished inspection at the end of *Pipeline Segment* 4 where she started the patrolling.

The number of feasible paths would increase exponentially as the number of pipeline segments and time segments increase. For instance, the case shown in Figure 7.1, has more than 1000 paths. PPG will examine which paths are better to follow by the patrol to secure the pipeline system more effectively. Hence, individual patrolling paths are deemed as available strategies for the patrol, and this will be explained more in subsection 7.3.4 as the patrol's strategy.

7.3.3 Attacker strategy

The attacker faces a whole pipeline system with its different *Pipeline Segment*s as available options for his attack. Accordingly, a pure strategy of the attacker is to choose a segment to attack. Therefore, the attacker's pure strategy set can be defined as the segments set, as shown in Formula (7.1), in which $nSeg$ denotes the number of segments.

$$S_a = \{1,2, \ldots, nSeg\} \cdots\cdots (7.1)$$

From the previous example shown in Figure 7.1, there will be nine strategies for attacking. The attacker will analyze to attack one of either *Pipeline Segment* one, as his first strategy, or *Pipeline Segment* 2 as his second strategy, till *Pipeline Segment* 9 to obtain the highest results.

7.3.4 Patrol strategy

The patrol strategies are derived from the probability of being present in a *Pipeline Segment*. This probability is called the Probability of Coverage(PoC), which can be defined as shown in Formula (7.2), in which t_i represents the time slices that the patrol spent on segment i and nT denotes the total time segments.

$$PoC_i = \frac{t_i}{nT} \cdots\cdots (7.2)$$

The patrol has some different paths to choose from. Each patrolling path presents one set of PoC on different *Pipeline Segments*. Therefore along with Figure 7.1, the patrolling paths can be represented as sets of PoCs which the patrol provides for a pipeline route. This set of PoCs for the bold patrolling path of Figure 6.1 can be shown in Table 7.3.

Table 7. 3. Example of PoC distribution on pipeline *Pipeline Segments*

Pipeline Segment	1	2	3	4	5	6	7	8	9
PoC	$^2/_{20}$	$^2/_{20}$	$^2/_{20}$	$^4/_{20}$	$^2/_{20}$	$^2/_{20}$	$^2/_{20}$	$^2/_{20}$	$^2/_{20}$

If all the patrolling paths are represented as in Table 7.3, it can be inferred that some of them have the same distribution of PoCs on different *Pipeline Segments*, meaning that they are equivalent. Consequently, we can combine these equivalent patrolling paths. This combination plays a crucial role in solving the PPG in the case of facing long pipeline routes and several *Pipeline Segments*; because otherwise, the game may become very large and complicated and hence difficult or impossible to solve.

Therefore, many patrolling paths may become equivalent, and they can be categorized in a set of paths with the same distribution of PoC. Therefore, the strategy set of the patrol can be defined as allocations of her total patrolling time among all the pipeline segments, as shown in Formula (7.3), in which, x_i denotes the time spent on pipeline segment i and Π denotes the Cartesian product.

$$S_d = \{\prod_{i=1}^{nSeg} x_i \,|\, \sum_{i=1}^{nSeg} x_i = nT\} \cdots\cdots (7.3)$$

Definition 1. A time allocation $s_d \in S_d$ is called Feasible Path Category (FPC) if at least one patrolling route can be generated.

For example, in the case shown in Figure 7.1, $s_d = (2,2,2,4,2,2,2,2,2)$ is a FPC, since at least the bold path shown in Figure 7.1 can be generated according to Γ, while $\Gamma = (4,4,2,0,4,4,2,0,0)$ is not an FPC, since the patrol cannot spend some time in segment 1,2,3 and 4,6,7 respectively but spend no time in segment 4. Thus no route can be generated according to this Γ.

Observation 1. Any time allocation $x = (x_1, x_2, \dots, x_{nSeg})$ which satisfies the following constraints c1 to c6 is an FPC (sufficient condition), and any FPC should satisfy these six constraints (necessary condition).

$\sum_{i=1}^{nSeg} x_i \leq nT$ (c1)

$x_i \in \mathbb{Z}_0^+, x_i\ is\ a\ even\ number$ (c2)

$x_i \leq nT - 2 \cdot (i - s - 1), \quad i = s + 1, s + 2, \dots, nSeg$ (c3)

$x_i \leq nT - 2 \cdot (s - i), \quad i = 1,2, \dots, s$ (c4)

$x_i \geq \varepsilon \cdot x_{i+1}, \quad i = s + 1, s + 2, \dots, nSeg - 1$ (c5)

$x_{i+1} \geq \varepsilon \cdot x_i, \quad i = 1,2, \dots, s - 1$ (c6)

In which $nSeg$ denotes the number of route segments, nT denotes the number of time segments, $s \in \{0,1, \dots, nSeg\}$ is the start point, $\varepsilon > 0$ is a small positive real number.

Proof: For the sufficient condition, we construct a patrolling path for the given plan category which satisfies these constraints c1 to c6. Given the x which satisfies these constraints, the patrol starts from the start node, goes to one direction until the last segment k which satisfies $x_k > 0$, and she oscillates in this segment until she stays enough time in this segment, then she goes to the segment x_{k-1}, and she oscillates in this segment until she stays enough time in this segment, and so forth. When she comes back to the start point, she goes to another direction and repeats the procedure. By obeying these steps, the patrol will find a patrolling path which satisfies the coverage constraint.

For the necessary condition, constraint c1 is obvious; constraint c2 reflects that the patrol will start from one node, and finally will come back to the start node; in this case, she will definitely stay in each segment for even times; constraint c3 and c4 reflect the fact that, starting from node i, the time the patrol is spending on further pipeline segments will not exceed the total time minus the time she walks from the starting node to the further segments. In Figure 7.1, c3 and c4 are shown as the trapezoid shape of the graph. constraint c5 and constraint c6 indicates that if the patrol spends some time on further segments, she should spend some time on closer segments since she has to walk to further segments through the closer segments.

Besides the patrol, fixed countermeasures are also employed for securing pipelines. Countermeasures can be classified in Table 7.4 with refers to Talarico et al. [(8)]. These security measures are installed on a pipeline system to reduce the likelihood and/or the consequences of the security incident. With this countermeasures, the attacker would have a probability of being detected, independent from the patrol, denoting as PoD.

As mentioned before, in PPG, it is assumed that if the patrol is present in a pipeline segment, she will catch the attacker and stop the attack. Therefore catching the attacker and stop the security incident depends on two factors, the Probability of Coverage (PoC) and Probability of Detection (PoD). The prospect of stopping adversarial attempts or security incidents is described with Probability of Stop (PoS) and can be clarified as shown in Formula (7.4).

Table 7. 4. Classification of countermeasures for the pipeline system

Goal: Reducing Likelihood Countermeasures		
Group	Description	Abbreviation
Traditional Countermeasures	Lighting	Li
	Fences	Fe
	Access Control ID	AC
	Integrated electronic access control	IEA
	Ground Patrol	GP
	Arial Patrol	AP
Advanced Countermeasures	Open-Air Intrusion Detection Sensors	PIDS
	Not Open-Air Sensor	NOAS
	Remote Sensing Systems	RSS
Recent Technologies	Drones Unmanned Aerial Vehicle	DUAV
	Distributed acoustic sensing	DAS
	Thermal Infrared Sensor	TIS
	Other ground sensors	GS
Goal: Reducing Consequences Countermeasures		
Other Countermeasures	Trained Personnel	TP
	Isolation Valve and ESD	ESD
	Non-flammable supports	NFS
	Procedures and emergency response plans	ERP
	Non-flammable valves and gaskets	NFVG
	PMS or monitoring system	PMS

The security countermeasures are designed and installed on pipeline system, so they have a constant PoD, nonetheless the PoC is variable and we are examining its different values as patrol strategies.

$$PoS = PoD + PoC - PoD * PoC \cdots\cdots\cdots (7.4)$$

The PoD and the PoC are derived from two independent events, the former reflects the effectiveness of a physical security countermeasures and the latter indicates the activities of the patrol.

7.4 Payoff modeling

The PPG is built based on the security risk assessment of the pipeline system to identify the patrolling schedules. The PPG risk assessment provides the basis for rank ordering of the penalties and the rewards of players in a defined *Pipeline Segment* according to their specific types. The penalty and the reward indicate benefits that the players can gain or lose in the game. These outcomes are calculated according to the consequences of the identified scenarios and their perceived values for both sides of the game. For this purpose, each *Pipeline Segment* is subjected to independent security risk assessment. Thus, PPG risk assessment will be applied to each *Pipeline Segment* separately. It should be stated that for every attacker type in each *Pipeline Segment,* based on the associated scenario, his penalty and reward of attack is different. For example, having four types of attackers leads to four sets of penalty and reward ranks in any *Pipeline Segment.*

From sub-section 7.4.1 to 7.4.4 all the penalties and rewards of attacking and defending the pipeline system for the patrol and attacker are evaluated. Later on, in sub-section 7.4.5, these disparate sets contribute to the calculation of payoff functions.

7.4.1 Attacker's reward

As stated before the probable consequences of security incidents are characterized by five different kinds. According to API recommanded practice 780 each consequence kind is classified from 1 to 5, as shown in Table 4.5 in Chapter 4 of this dissertation.

Because different types of attackers are seeking to various outcomes, the value of the results or their rewards in the case of a successful attack is different. Then the calculation of the Reward for each type of the attacker, whenever he attacks successfully to a specific *Pipeline Segment* (e.g., if the terrorist explodes the pipeline) can be done in a way which is illustrated in Table 7.5. WFs are derived from Table 7.2 and in this example WFs are belong to terrorist.

Table 7. 5. Calculation of the terrorist's reward

Type of consequence	WF	A	B	C	D	E	Score
		1	2	3	4	5	
Fatalities and injuries	3		$3 \cdot 2$				6
Environmental impacts	1			$1 \cdot 3$			3
Property damage	2	$2 \cdot 1$					2
Business interruption	2					$2 \cdot 5$	10
Damage to Reputation	3				$3 \cdot 4$		12
						Sum	33

Like Table 7.5, for each type of attacker, we can find the rewards separately. Specifically, there are four types of attackers, so four disparate attacker's rewards will be found. For example R_a^1 and R_a^2 indicate the Rewards of Terrorists and criminals, respectively, for attacking the pipeline.

7.4.2 Patrol's penalty

From the patrol's point of view, the consequences of security incidents on the pipeline system have particular effects on the company and also on the society where the incident happens. The fatalities, environmental impact or other kinds of damages have different values for the patrol compared to the attacker.

Following Table 4.5 in Chapter 4, the security department has assigned a set of WFs for various kinds of consequences. These WFs reflect the perceived values of each kind of consequences and their contributions to estimating the patrol's penalty when a security incident happens.

Next, similar to the calculation of the attacker's reward through Table 7.5, the Penalty for the patrol after a successful attack can be estimated by the same procedure. It should be noted that the penalty of the patrol is the same for all types of attackers.

7.4.3 Patrol's reward

Before, we discussed the outcomes in case the security incident takes place. Now we are going to find the gains of both players – attacker and patrol – if the patrol stops the incident from happening.

Aside from preventing direct consequences, stopping the security incidents can have some benefits for the patrol, such as obtaining critical information about the vulnerabilities and weak points in the security measures, gaining a positive impact on the effectiveness of the security team along with favorable feedbacks from public media. Accordingly, the calculation of the reward of the patrol facing attacker's type k (indicated as R_d^k) can be formulated like Table 7.6. For instance R_d^1 means the Reward of the patrol if she stops the terrorist from any security incident while R_d^2 is her reward in preventing such an incident from criminals. In this table, for each type of attacker, a score can be assigned to specific kinds of consequences, and the summation of the scores can be deemed as the patrol's Reward. Table 7.6 presents an example of Patrol's Reward calculation (later it will be used in our case study).

Table 7. 6. Calculating the patrol's Reward

Kinds of consequence	R_d^1	R_d^2	R_d^3	R_d^4
	Terrorist	Criminal	Disgruntled insider	Activist
Information (5-10)	9	5	7	6
Media (1-5)	5	4	1	3
Reputation (1-5)	5	5	3	4
Sum	19	14	11	13

7.4.4 Attacker's penalty

After an unsuccessful attack to a *Pipeline Segment,* the attacker may be caught by the security forces, and he will lose all of his investments. The most important loss of the attacker is the information that he will give to the patrol. This information can have more significant value if he belongs to a larger group. Also, the unsuccessful security incidents make the security measures stricter and actually improve the security level.

The Penalty of the attacker of type k, indicated by P_a^k, when he is stopped by the patrol, can be estimated similar to Patrols Rewards. There are three kinds of consequences for the attacker after being stopped by the patrol such as Information, Investment attack, Sentenced to the jail. For each kind, a score from 1 to 5 can be given. Like Table 7.6, the summation of these scores presents the attacker's Penalty.

Here P_a^1 means the Penalty of the terrorists when they are stopped by the patrol whereas P_a^4 is the activist's Penalty for the unsuccessful attack.

7.4.5 Payoff function

For calculating the payoff to each player when the patrol faces the attacker of type k and in case of the patrol's pure strategy s_d and attacker's pure strategy s_a: if the attack fails, then the patrol would gain the reward R_d while the attacker would receive a penalty P_a ; otherwise the patrol would receive a penalty P_d and the attacker would gain a reward R_a.

u_a^k is defined as a payoff for the attacker of type k.

$$u_a^k(s_d, s_a) = (1 - PoS) * R_a^k - PoS * P_a^k \cdots\cdots(7.5)$$

Similarly u_d^k is the patrol payoff facing the attacker of type k.

$$u_d^k(s_d, s_a) = PoS * R_d^k - (1 - PoS) * P_d \cdots\cdots(7.6)$$

The PoS is the probability of stopping the attacker by the patrol. The whole process of PPG model can be presented schematically in Figure 7.3.

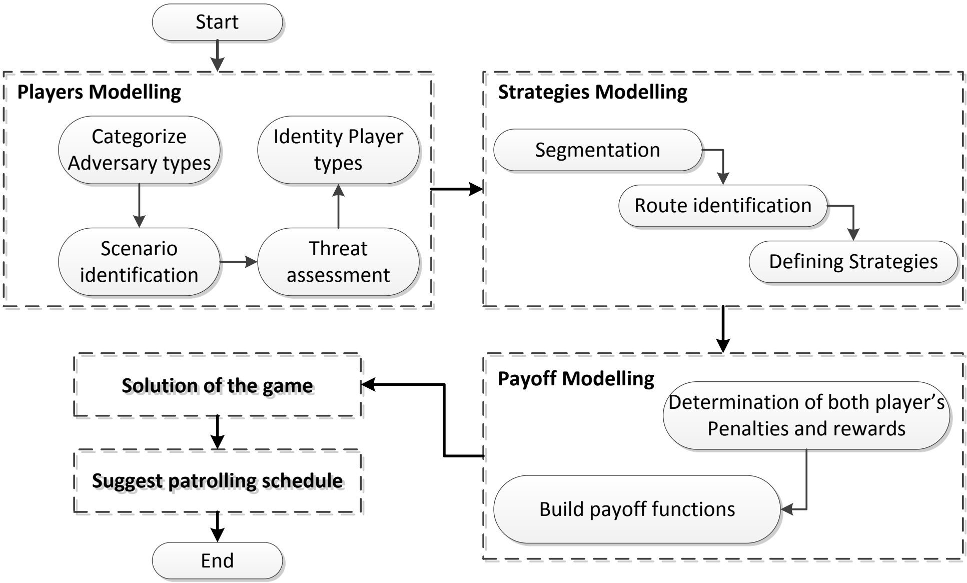


Figure 7. 3. Schematic presentation of PPG process

7.5 PPG algorithm

The PPG is a Bayesian Stackelberg game: the defender moves first, considering the potential different types of attackers, and knowing that the attackers will play their best response to her committed strategy, thus she plays accordingly. To this end, the so-called Strong Stackelberg Equilibrium can be used to predict the output of the game.

Definition 2: A Strong Stackelberg Equilibrium (SSE) (x^*, i) for the PPG can be defined as:

$x^* = arg\max_{x \in S_d} \sum_{k=1}^{4} \rho^k \cdot u_d^k(x, i^k(x))$ ··(7.7)

$i^k(x) = arg\max_{i^k \in S_a} u_a^k(x, i^k)\, , \forall k = 1{,}2{,}3{,}4$··(7.8)

In which x must also be an FPC as defined in definition 1 in section 7.3.4, a vector with length of number of route segments; ρ is the Bayesian probability of different types of attacker, as defined in section 7.2.3; u_d^k (u_a^k) is the patrol's (attacker's) payoff when facing the k^{th} type of attacker, as defined in Formulas (7.5) and (7.6), in section 7.4.5; $i(x)$ denotes the attacker's best response to the patrol's committed strategy, which is a pipeline segment.

For solving the PPG game and finding the SSE, an mixed-integer linear programming (MILP) algorithm is proposed, shown as Algorithm 1. For the sake of clarity, all the inputs of the algorithm are summarized in Table 7.7, and variables are presented in Table 7.8.

The objective function "OF" indicates that the patrol faces several types of adversaries, and she wants to maximize her expected payoff concerning these different adversaries.

Constraint (1) indicates that the sum of time spent in each route segment cannot exceed the total patrolling time. Constraints (2), (3), and (5) mean that observing the patrol's strategy, each type of adversary will play its best response, which is a pure strategy. Note that in (3),

$q_i^l = 1$ indicates the attacker's best response, and thus: $a^k = (1 - PoS_i) \cdot RA_i^k - PoS_i \cdot PA_i^k$; otherwise, $q_i^k = 0$, and thus: $a^k \geq (1 - PoS_i) \cdot RA_i^k - PoS_i \cdot PA_i^k$.

For constraint (4), the inequality will be tight only in case of $q_i^k = 1$, indicating that the γ^k is the defender's payoff when the attacker chooses route segment i to attack. Constraint (6) refers to Formula (7.4) in section 7.3.4. Constraints (8) to (12) are used to make sure that the allocation of the time segment is a FCP as defined in definition 1 in section 7.3.4.

Note that constraint (8) is not a standard linear constraint. However, when the algorithm is implemented, all the x_i can be replaced by a $x_i = 2y_i$, which will not change the linear property of other constraints.

Table 7. 7. Input of the algorithm

Notation	Definition	Comments
TL	Set of different attacker types	terrorists, criminals etc.
$nSeg$	Number of route segments	
nT	Patrolling time segments	
s	Patrolling start point	an integer from 0 to $nSeg$
RA^k	Reward vector for the k^{th} type of attacker	a $1 \times nSeg$ vector, $\forall k \in TL$
RD^k	Patrol's Reward vector for detecting the k^{th} type of attacker	a $1 \times nSeg$ vector, $\forall k \in TL$
PA^k	Penalty vector for the k^{th} type of attacker	a $1 \times nSeg$ vector, $\forall k \in TL$
PD^k	Patrol's Penalty vector of a success attack from the k^{th} type of attacker	a $1 \times nSeg$ vector, $\forall k \in TL$
PoD	Probability of detection	a $1 \times nSeg$ vector
ρ^k	probability if the attacker has type k	
TS	Threat level of each attackers	a $1 \times \|TL\|$ vector

Table 7. 8. Variables of the algorithm

Notation	Definition	comments
x	time segments allocation plan	➢ x is a $1 \times nSeg$ vector, and $\sum x_i \leq nT$
q^k	Strategy vector of the k^{th} type attacker	➢ q^k is a $1 \times nSeg$ vector, $q_i^k \in \{0,1\}$, and $\sum q_i^k = 1$, for all $l \in TL$. $q_i^k = 1$ indicates that pipeline segment i is the k^{th} attacker's best response target.
a	Attacker's optimal payoff	
γ^k	Patrol's optimal payoff facing attacker of type k	➢ γ^k is a $1 \times \|TL\|$ vector

Algorithm 1 Finding the optimal PoC.

$$max \sum_{k\in TL} \rho^k \cdot \gamma^k \tag{OF}$$

$$\sum_{i=1}^{nSeg} x_i \leq nT \tag{1}$$

$$\sum_{i=1}^{nSeg} q_i^k = 1 \tag{2}$$

$$0 \leq a^k - \left[(1 - PoS_i) \cdot RA_i^k - PoS_i \cdot PA_i^k\right] \leq \left(1 - q_i^k\right) \cdot M, \forall i = 1,2, \dots, nSeg \tag{3}$$

$$M \cdot \left(1 - q_i^k\right) + \left[PoS_i \cdot RD_i^k - (1 - PoS_i) \cdot PD_i^k\right] \geq \gamma^k, \forall i = 1,2, \dots, nSeg \tag{4}$$

$$q_i^k \in \{0,1\}, \ \ \forall i = 1,2, \dots, nSeg \tag{5}$$

$$PoS_i = \frac{x_i}{nT} + PoD_i - PoD_i \cdot \frac{x_i}{nT} \tag{6}$$

$$a^k \in \mathcal{R}, \gamma^k \in \mathcal{R} \tag{7}$$

$$x_i \in \mathbb{Z}_0^+, x_i \ is \ a \ even \ number \tag{8}$$

$$x_i \leq nT - 2 \cdot (i - s - 1), \ i = s + 1, s + 2, \dots, nSeg \tag{9}$$

$$x_i \leq nT - 2 \cdot (s - i), \ i = 1,2, \dots, s \tag{10}$$

$$x_i \geq \varepsilon \cdot x_{i+1}, \quad i = s + 1, s + 2, \dots, nSeg - 1 \tag{11}$$

$$x_{i+1} \geq \varepsilon \cdot x_i, \quad i = 1,2, \dots, s - 1 \tag{12}$$

Algorithm 2 Generating patrolling routes from the path category

```
GPP(TG, $x^*$)
    Set RouteList := {};
    Stack S := {};   ( start with an empty stack )
    for each vertex u, list u.visitedset := {};
    push S, s;
    while (S is not empty) do
        v := top S;
        node set cv := nodes which can be reached from v by one step
        for each node i in cv
            if i∉S AND i∉v.visitedset
                push S, i;
                add v.visitedset i;
            end if
        end for
        nv := top S;
        if nv and v the same
            clear v.visitedset
            pop S;
        else if S not satisfy $x^*$
            pop S;
        else if nv and t the same
            add Routelist S;
            pop S;
        end if
    end while
END GPP
```

Algorithm 1 only computes the optimal allocation of time segments, or the so-named path category. The pipeline security management department needs the exact patrolling routes. To this end, the algorithm 2 is proposed to generate patrolling routes from the given optimal path category, and it is a depth-first-search (DFS) based algorithm.

Patrolling routes are generated by Algorithm 2 from the path category. The input of the algorithm is the patrolling graph TG (as shown in Figure 7.1), and the defender's BSE strategy x^{*}. The output of this algorithm is the entire route list which belongs to this path category.

7.6 Case study

In order to understand the PPG, a short piece of the pipeline route has been chosen for finding an optimum patrolling schedule. Sub-section 7.6.1. defines this case study and results are discussed in sub-section 7.6.2.

7.6.1 Case study definition

We follow the example of Section 7.3 (strategy modelling), in which all the patrolling paths can be modeled through Figure 7.1. This pipeline route is presented more specifically in Table 7.9, in which the patrol starts from, and ends to, the starting point of segment 5:

Table 7. 9. Pipeline route

Number	Section	Description	Route Length(km)	Length
1		Under river	0.5	
2	1. industrial area	Pass a road	0.5	1.5
3		Warehouse	0.5	
4		Metering and branching	0.6	
5		pass a river	0.5	
6	2. Urban area	Pass a highway	0.5	2.7
7		Roadhouse	0.5	
8		Pass a highway	0.6	
9	3. Country side	Hiking path	0.8	0.8

The duration of patrolling is 1.8 hour, and the length of pipeline is 5 km. This patrolling game is modeled by 9 *Pipeline Segment*s and 20 *Time Segments*. Consequently, there are nine attack strategies for the attacker.

On these 9 *Pipeline Segments,* countermeasures may also be employed to increase the security level of the pipeline system. We will solve the PPG in two cases: the first one in the absence of any physical countermeasure on the pipeline and the second in the presence of a set of countermeasures. These two cases are presented in Table 7.10 with the efficiency of the measures.

Table 7. 10. Probabilities of detections of the two cases *

	RT 1	RT 2	RT 3	RT 4	RT 5	RT 6	RT 7	RT 8	RT 9
Case 1	0	0	0	0	0	0	0	0	0
Case 2	0	0	0	0	0.4	0.3	0.5	0.4	0

* Note: The *Route Segment* is abbreviated with RT

Following player modelling, there are four types of attackers. These types of attackers in addition to the patrol can be characterized by sets of WFs similar to what is presented in Table

7.2. In this illustration, just one scenario for the security incident system is considered for all types of the attacker, which is an explosion of the pipeline.

In the payoff modeling part, the consequences of this unique scenario are evaluated according to the risk ranking system of Section 7.4 and the results are estimated for these 9 *Pipeline Segments* in Table 7.11.

Table 7. 11. Ranking different kinds of consequences

	Industrial area			**Urban area**					**Countryside**
Kinds of consequences	***Route Segment* Number**								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Fatalities and injuries	3	4	3	3	3	4	5	4	3
Environmental impacts	3	2	2	2	3	2	2	2	4
Property damage	2	4	4	5	3	4	5	4	3
Business interruption	2	3	3	4	2	3	3	3	2
Damage to Reputation	4	4	4	5	5	5	5	5	4

The threat assessment of these adversaries in this illustrative case results in level 4 for the terrorist ($T^1 = 4$), level 3 for the criminals ($T^2 = 3$), level 1 for the disgruntled insider ($T^3 = 1$), and level 2 for the activists ($T^4 = 2$). Accordingly, the ρ^ks can be calculated, resulting in 0.4, 0.3, 0.1, and 0.2, for the terrorist, the criminals, the disgruntled insider, and the activists, respectively.

7.6.2 Results and discussion

The illustrative pipeline system was explained before, and in this subsection, we are solving it through the PPG algorithm introduced in section 7.5.

In the first case, the BSE strategy for the defender is $x^* = (0,2,2,4,2,4,4,2,0)$, which means if the defender schedules her patrolling path based on this coverage, in comparison to the other choices, she will secure the pipeline system in the best possible way.

Since there isn't any countermeasure on this pipeline system, all the PoSs are equal to the PoCs. This means that if the attacker wants to attack *Pipeline Segment* 8, the probability of presence of the patrol is $2/20$, consequently the probability of getting caught is only of $2/20$.

Table 7. 12. Payoffs in case one of the illustrating case study

Seg	PoD	PoC	PoS	u_a^1	u_d^1	u_a^2	u_d^2	u_a^3	u_d^3	u_a^4	u_d^4
1	0	0	0.0	32.00	-26.00	6.00	-26.00	21.00	-26.00	29.00	-26.00
2	0	2/20	0.1	34.70	-27.80	9.60	-28.30	20.70	-28.60	27.90	-28.40
3	0	2/20	0.1	32.00	-25.10	9.60	-25.60	20.70	-25.90	27.90	-25.70
4	0	4/20	0.2	32.60	-24.20	**9.60**	-25.20	22.20	-25.80	29.40	-25.40
5	0	2/20	0.1	32.00	-25.10	6.90	-25.60	21.60	-25.90	29.70	-25.70
6	0	4/20	0.2	31.80	-24.20	7.20	-25.20	19.80	-25.80	26.20	-25.40
7	0	4/20	0.2	35.80	-28.20	9.60	-29.20	20.60	-29.80	27.80	-29.40
8	0	2/20	0.1	**37.40**	-29.60	9.60	-30.10	**23.40**	-30.40	30.60	-30.20
9	0	0	0.0	35.00	-29.00	9.00	-29.00	23.00	-29.00	**34.00**	-29.00

The outcome for each player when the attacker attacks to each Pipeline Segment in case one are presented in Table 7.12. The corresponding payoffs of the best response strategy for each type of attacker are bolded in the table. Since the attacker is assumed to be completely rational, these strategies, which lead him to the highest payoff, are the most probable ones to be chosen. The second type of attacker (i.e., the criminal) is indifferent between attacking segments 2, 3, 4, 7, or 8, all resulting a payoff of 9.6. However, the BSE that was employed in this chapter is the Strong Stackelberg Equilibrium. Therefore, the criminal would break the tie in the favour of the defender, leading to the result that segment 4 is his best response strategy target.

In addition to payoffs of different types of the attacker, the defender expected payoffs in the face of each type of the attacker are presented in Table 7.12. Nevertheless, by considering the conditional probability on each type of the attacker, the overall defender's payoff by playing her BSE strategy is equal to -28.24. According to Table 7.12, in case one, the PPG algorithm 2 provides us a set of 36 patrolling paths having the optimum probability of the coverage (PoC). Some of these paths are presented in Figure 7.4.

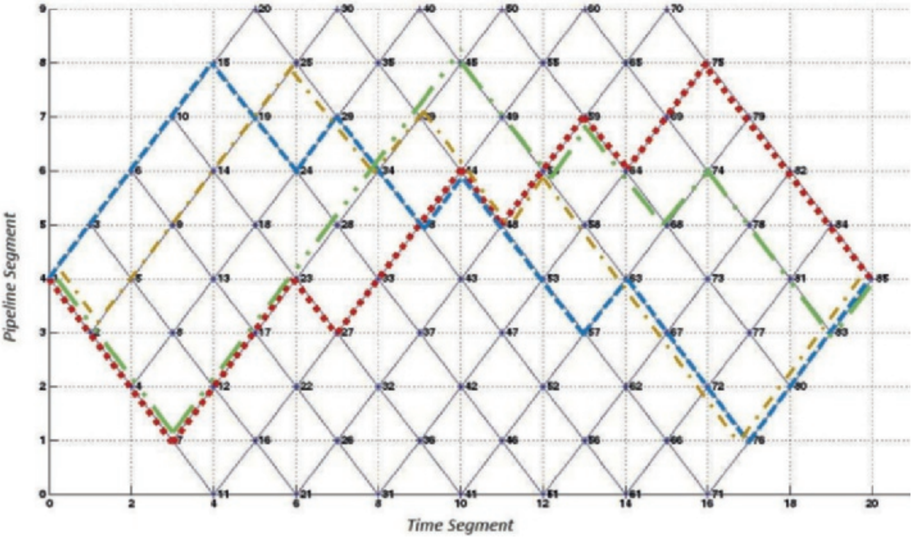


Figure 7. 4. Patrolling paths in case one

In the second case, in the presence of countermeasures, the patrol's BSE strategy is $x^* = (0,4,2,4,2,2,2,2,2)$. Table 7.13 demonstrates some further information, if the defender would play her BSE strategy. As shown in the table, the attackers' best responses would be attacking segment 4, 4, 4, and 9, for the domestic terrorists, the criminals, the disgruntled insider, and the extreme activists, respectively.

As shown in Table 7.13, by implementing a set of countermeasures on this illustrative pipeline system, the probability of catching the attacker in some *Pipeline Segments* has slightly increased. For instance, the probability of existing a patrol in *Pipeline Segment* 6 is $2/20$, but by installing a countermeasure with PoD of 30% the probability of stopping the malicious act increases to 37%. For *Pipeline Segment* 6, although the attack consequences are higher than *Pipeline Segment* 3, the duration of the patrol inspections are the same, both for 2 *Time Segments*, because a countermeasure implementation in *Pipeline Segment* 6 provides a PoS of 37% and patrol doesn't need to spend more time at that location.

Table 7. 13. Payoffs in case two of illustrated case study

Seg	PoD	PoC	PoS	u_a^1	u_d^1	u_a^2	u_d^2	u_a^3	u_d^3	u_a^4	u_d^4
1	0	0	0	32.00	-26.00	6.00	-26.00	21.00	-26.00	29.00	-26.00
2	0	4/20	0.2	29.40	-22.60	7.20	-23.60	17.40	-24.20	23.80	-23.80
3	0	2/20	0.1	32.00	-25.10	9.60	-25.60	20.70	-25.90	27.90	-25.70
4	0	4/20	0.2	**32.60**	-24.20	**9.60**	-25.20	**22.20**	-25.80	29.40	-25.40
5	0.4	2/20	0.46	14.00	-7.46	-0.66	-9.76	9.36	-11.14	14.22	-10.22
6	0.3	2/20	0.37	22.28	-15.02	3.12	-16.87	13.68	-17.98	18.72	-17.24
7	0.5	2/20	0.55	14.45	-7.55	0.15	-10.30	7.65	-11.95	11.70	-10.85
8	0.4	2/20	0.46	17.24	-10.16	0.96	-12.46	10.44	-13.84	14.76	-12.92
9	0	2/20	0.1	30.20	-24.20	6.90	-24.70	19.80	-25.00	**29.70**	-24.80

Comparing the results in Table 7.12 to the results in Table 7.13, the effect of installing a set of countermeasures on the pipeline system can be found. *Pipeline Segment* 7 is more critical than other *Pipeline Segments*, therefore in case one, the patrol should spend more time inspecting this *Pipeline Segment* than the others. However, if a countermeasure with a *PoD* of 50% is installed on that segment, the criticality is decreased so as to the patrol can inspect it less or like the other segments (in 2 *Time Segments*). For *Pipeline Segment* 2 the situation is different. Before installing any countermeasure, the patrol should plan to inspect this *Pipeline Segment*, which is passing a road, in two *Time Segments* to have the highest payoffs. Having a countermeasure implemented on other *Pipeline Segment*s decreases the probability of a successful attack in those segments. Thus in comparison to other segments this *Pipeline Segment* becomes more critical and the patrol is better to inspect it in four *Times Segment*s to have higher payoffs. In *Pipeline Segment* 5, installation of a countermeasure with 40% *PoD* can increase the probability of stopping the attack from 10% to 46%, nonetheless the number of patrol inspections is the same.

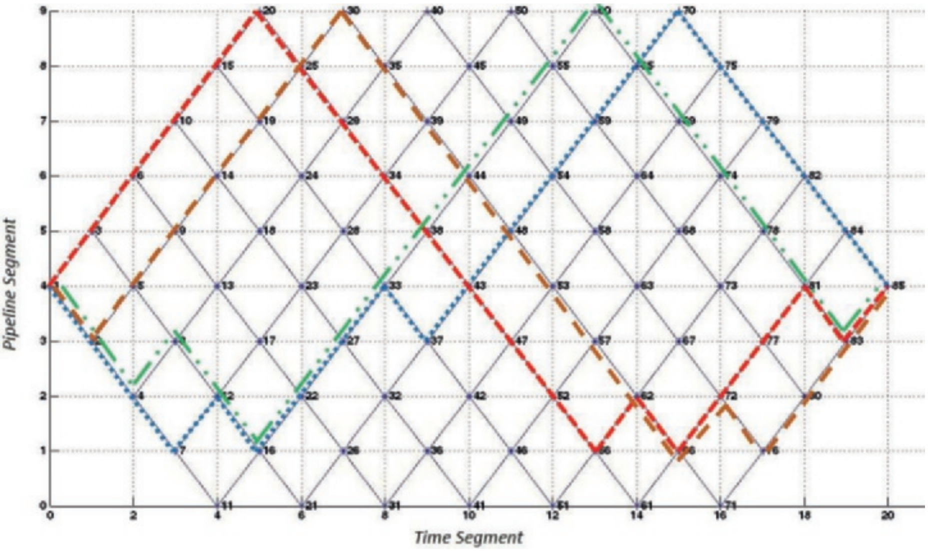


Figure 7. 5. Patrolling paths in case two

Six different patrolling routes can be generated according to the defender's BSE strategy. Figure 7.5 shows 4 of these routes, in different colours.

The illustrative case study thus shows that PPG properly produces the optimum paths for the pipeline considered. Moreover, the illustration shows that PPG considers the effectiveness of countermeasures and how they can change the optimal patrolling paths. This feature gives the security department of a pipeline company an advantage of considering the physical protection system in their patrol route and managing their schedule by installing additional countermeasures on their system.

7.7 Conclusions

In this methodology, however, we assumed that both players are completely rational. Since rationality is not always applied in real security challenges, modelling of the bounded rationality of the players should be taken into account in future research, as the Quantal Response Equilibrium, described by Mckelvery and Palfrey (9, 10), that has a superior ability to model human behaviour in simultaneous and extensive move games. An et al. (11) for instance used this model in the algorithm of their patrol scheduling game, and the application of such bounded rationality models is postponed for further development of PPG.

We have developed a methodology – the so-called **P**ipeline **P**atrolling **G**ame (PPG) – based on a security game to identify the optimal patrolling schedule on the pipeline system. The method is based on a discretization of patrolling time and distance. This discretization provides a framework to model the patrolling paths mathematically. In this framework, we can implement a risk rank ordering system to assess the security risk on the whole pipeline system. The developed methodology produces different patrolling paths according to the results of security risk assessments by which the pipeline system can be protected by an optimal strategy.

By the PPG a pipeline company can ensure that their pipeline system is protected in the best way. Since PPG can produce a set of patrolling paths that except security department nobody knows which of them will be chosen. Security Department can choose one of these paths randomly and be confident that they will inspect the high risky areas more than less risky parts. According the assumptions, although the attacker can know the importance of different sections or guess the frequency of inspecting a section, they don't know the exact plan of the security department and they won't be successful more than specific level.

References

[1] Reniers GLL, Jongh KD, Gorrens B, Lauwers D, Leest MV, Witlox F. Transportation Risk ANalysis tool for hazardous Substances (TRANS) - A user-friendly, semi-quantitative multi-mode hazmat transport route safety risk estimation methodology for Flanders. Transp Res Part D Transp Environ. 2010;15(8):489-96.

[2] Rezazadeh A, Talarico L, Reniers G, Cozzani V, Zhang L. Applying game theory for securing oil and gas pipelines against terrorism. Reliability Engineering & System Safety. 2018.

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

[4] Reniers, Herdewel, Wybo. A Threat Assessment Review Planning (TARP) decision flowchart for complex industrial areas. Journal of Loss Prevention in the Process Industries. 2013;26(6):1662-9.

[5] CCPS CfcPS. Guidelines for chemical transportation safety, security, and risk management. New York, USA: America Institute of Chemical Engineers, John Wiley & Sons; 2008.

[6] CCPS CfCPS. Guidelines for Analyzing and Managing the Security Vulnerabilities of Fixed Chemical Sites. New York, USA: America Institute of Chemical Engineers, John Wiley & Sons; 2010.

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

[8] Talarico, Sorensen, Reniers, Springael. pipeline securiry. In: Hakim, Albert, Shiftan, editors. Securing Transportation Systems: John Wiley & sons, Inc.; 2015. p. 285-315.

[9] McKelvey, Palfrey. Quantal Response Equilibria for Normal Form Games. Games and Economic Behavior. 1995;10(1):6-38.

[10] Mckelvey, Palfrey. Quantal Response Equilibria for Extensive Form Games. Experimental Economics. 1998;1(1):9-41.

[11] Bo An, Shieh, Yang, Tambe, Baldwin, DiRenzo, et al. PROTECT - A Deployed Game-Theoretic System for Strategic Security Allocation for the United States Coast Guard. AI Magazine. 2012.

8

VALIDATING THE MODELS: REFLECTIONS FROM INDUSTRIAL PRACTITIONERS

This chapter concerns the validation of the models proposed in previous chapters from an industrial practice perspective. Among several model validation methods, the method of expert judgment is used. Six senior security managers from Sitech (NL), Solvay (BE), Port of Antwerp (BE), BASF SE Antwerp site (BE), Shell (NL), and CIMIND (BE) were invited to evaluate the proposed models. The possibility of applying the models in industrial practice was questioned and the gaps between the models and current industrial practice are clarified and shown.

8.1 Motivation

This dissertation developed several quantitative models (i.e., the DAMS model in Chapter 3, the CPP game in Chapters 4 and 5, the CCP game in Chapter 6, and the PPG in Chapter 7) for bettering security protection in chemical industrial areas. However, validating these models seems to be more challenging than developing them. Box [(1)] argued that "all models are wrong, but some of them are useful". Therefore, the verification and validation of proposed/developed models are needed, to guarantee the usefulness of these models.

Verification is intended to ensure that the models do what they are developed for. [(2)] All the models in this dissertation are developed for dealing with some specific challenges. The DAMS model is developed for modelling the propagation of domino effects, considering higher-level escalation, synergistic effects, and time dependences. The CPP game is for the purpose of protecting a single chemical plant, taking into account the intelligent interactions between the defender and the attacker. The CCP game and the PPG model aim at optimizing patrolling in a multiple chemical plant industrial park and of a pipeline respectively, regarding adaptive attackers. For each model (and for its extensions), one or more case studies are employed, to verify that the calculations of the model are correct and the model addresses the challenges that it is elaborated for.

Validation is the procedure of ensuring that the model represents the reality. Models are the simplified forms of reality and they are developed for studying certain characteristics of the actual system. For instance, the models developed in this dissertation concern the security protection perspective of chemical industries, while the chemical process procedures are ignored. Furthermore, even for investigating the same perspective of the actual system, the model may be varied on the resolution, according to the purpose of the study. The CPP game, for example, focuses on setting different security alert levels at each entrance and in each (sub-)zone of a single plant, while a higher resolution security model may be exploring the vulnerability of an entrance under a certain security alert level and certain attack scenario. Therefore, validating a model means ensuring that the model maintains the core characteristics of the actual system and only ignores the unessential properties, and further ensuring that the model is built on a proper resolution and the results of the model have enough fidelity to the actual system.

This chapter aims to discuss the validation of the proposed models from an industrial practice viewpoint, while the verification of the models was achieved and demonstrated by the case studies in every chapter. Section 8.2 illustrates multiple model validation methods and the method of expert judgment is chosen to validate the models in this dissertation. Section 8.3 shows the results, i.e., the gaps between the models and industrial practice, what kind of improvements are needed to make the models applicable in practice, etc., using the results derived from the interviews of six senior security managers from chemical companies or industrial activities related to them. Conclusions are given in section 8.4.

8.2 Model validation methods

The proposed models can be validated via four possible approaches:

Benchmark Comparison: Security is now being managed in industrial practice, based on some standards/regulations/methods, for example, the API SRA. [(3)] Therefore, for the same chemical plant/cluster, if we implement the proposed models in this dissertation and compare the results with that from the currently used methodology, we could see the advantages and disadvantages of the models.

Several reasons prevent us to employ the method of benchmark comparison as our validation method. Firstly, in industrial practice, documents concerning the preparation, implementation, and results of the security risk assessment are confidential. We do not have access to these data, which should be the benchmark in the method. Secondly, indicators for evaluating the effectiveness of security protection are not well defined yet. "One hundred percent secure" is impossible. Every method has its own advantages and disadvantages. Due to the lack of well-defined performance indicators, even if we could be able to get all the data of the implementation of both the benchmark method and of the proposed models, we would not be able to compare them. Thirdly, the benchmark models and the proposed models have different overall objectives. The benchmark models focus on obtaining data and they process these data in a simple but robust approach, aiming to reach a "balanced protection", or, in game theoretic terminology, minimizing the maximal vulnerability. On the contrary, the proposed game theoretic models process the obtained data in a more complex approach, to make the best use of the defender's knowledge about the attackers, resulting in optimal but less robust recommendations.

Red Team Experiments: In cybersecurity, among others, ethic attackers test the efficiency of security measures. The idea can also be used in the physical security domain. A group of people can be temporally employed as "ethic attackers" to try to intrude the plant under certain security strategies recommended by a certain security method. By comparing the success rate of intrusion, the efficiency of different methods can be compared.

However, to conduct a red team experiment can be quite expensive. For some probabilistic results, we have to repeat the experiment many times, in order to get the statistic results. A red team experiment is also difficult to organize. If security staff members are informed that a security drill is going on, they would then be more careful than they usually are, while if they are not informed, the repeated implementation of the drill will reduce the awareness of the security staff members.

Simulation Gaming: One way to reduce the cost and managerial difficulties of implementing a red team experiment is to use simulation gaming. A digital twin of the security defence system of a chemical plant/cluster can be built. The defender may implement different strategies on the defence system, while people can be invited to play the game as an attacker. In this way, it is relatively easy to ask different people to repeatedly play the game and therefore, statistical data can be collected. Different security strategies can also be easily implemented. The digital twin of the chemical plant/cluster can be an online game and people can, therefore, play the game remotely, leading to a lower cost. The simulation gaming approach may suffer from the difficulties of building a digital twin of the actual system. Conversely, a sand table simulation is much easier to be executed.

However, despite the difficulties of building such systems, the verification, validation, and accreditation (VV&A) of a simulation model, either a computer simulation model or a sand table simulation model, is a challenging topic in itself.

Expert Judgment: Models can also be evaluated by experts of the actual system. Security management is a part of the daily operation in chemical plants/clusters. Security practitioners know what are the bottlenecks of the current security management method and what are the valuable models that may help them to improve the protection of their assets. By introducing the proposed models to industrial security practitioners, and asking their opinion of the models, we can know how much the models reflect reality and what are the gaps between the proposed models and the security practice and how to fill the gaps.

If the method of expert judgment is used, the validation of the models is based on the experts' intuitions. An obvious drawback of using expert judgment is that the validation is not based on solid evidence. A perfect validation of a model includes the validation of the assumptions, the inputs, the static and dynamic model, and the outputs. However, as stated in Chapter 2, there is a lack of security related data in the chemical industries, which not only is a challenge for developing models, but also is an obstacle for validating security related models. Therefore, a solid validation of the proposed models is difficult, if not impossible, to achieve.

Furthermore, security risks are caused by deliberate behaviours. Human intentions are involved in the risk analysis procedure, making the validation of these security risk analysis models even more difficult. On the one hand, if experiments would be conducted, the cost can be quite high. On the other hand, human behaviour (especially intentional behaviours) analysis is a difficult topic since individuals are heterogeneous and those who are terrorists (which is one of the threats to the current chemical industries) are obviously different to ordinary people.

Based on the above analyses, expert judgment is chosen as the method to validate the models proposed in this dissertation.

8.3 Reflection from industrial practitioners

Six senior security managers were interviewed. This section reports how the interviews are organized. Results from the six interviews are also given, by aggregating all the comments from the six interviews.

8.3.1 Methodology

Preparation: The models in this dissertation are developed for the purpose of improving the protection of chemical areas. Therefore, we sent invitations to multiple first-line security mangers from chemical plants located in the Netherlands and Belgium. Finally, six managers agreed to participate in the evaluation of the models. Table 8.1 shows the information of the interviewees.

Table 8. 1. Information of the interviewees

No.	Name	Company/site	Position	Years of experiences of security management
1	Jos Weijers	Sitech Services	Senior Security Manager	Over 15 years
2	Werner Cooreman	Solvay	Group Security Director	Over 16 years
3	Kathy Dua	Port of Antwerp	Consultant Port Security & Safety	Over 16 years
4	Alexander Holzer	BASF Antwerp	Site Security Manager	Over 15 years
5	Dick Brummelhuis	Shell Pernis	Security & Crisis-manager	Over 33 years
6	Paul van Lerberghe	CIMIND	General Manager	Over 30 years

In most chemical plants, the security department is separated from the health, safety, and environment (HSE) department, being an independent unit. Therefore, it is worth noting that all the interviewees except Mrs. Dua shown in Table 8.1 are from the security department. Mrs. Dua works for the security and safety department of the Port of Antwerp, auditing the compliance of security & safety regulations (e.g., the International Ship and Port Facility

Security Code) in each chemical site located in the Antwerp port. Large chemical companies, such as Solvay, BASF, and Shell, have their own security department. Sitech service is a third-party company who provides security service for a large chemical industrial park named Chemelot in the Netherlands. CIMIND is a security consultant company who provides security consultancy not only to European companies but also to U.S. companies.

As shown in Table 8.1, all interviewees have more than 15 years of experience working on physical security. For each interview, the interviewee received the related documents introducing the models developed in this dissertation. The mathematical details of the proposed models were omitted, and the purpose/input/output of each model was briefly explained. The interviewees were asked to read the documents before the interviews took place.

Elicitation: According to the interviewees' convenience, the first interview with Weijers took place on 24 May 2018 at Chemelot while the last one with van Lerberghe took place on 31 July 2018 at TUDelft. Each interviewee was asked for a one-hour appointment for the interview. However, actually five of the six interviews lasted for more than 1.5 hours. Furthermore, guided site tours were provided by Weijers, Holzer, and Brummelhuis, at Chemelot, BASF Antwerp site, and Shell Pernis respectively, for obtaining a real impression of their security protection.

During each interview, instead of asking the interviewee a long list of closed questions, we asked for the interviewee only to consider two questions: is any one of the models (i.e., DAMS, CPP game, CCP game, and PPG) applicable to his/her site? What is the gap between the models and industrial practice? To answer these two questions, interviewees needed to be informed about more details of the model. Therefore, we further introduced the models to the interviewee during each interview by giving a presentation. The interviewees commented and provided feedback during the presentation.

Aggregation: Since only six interviews were conducted and we did not list closed-questions, we opted for a qualitative process for drawing results and conclusions. During the interviews, different interviewees had different comments on the models. However, due to the similarity of their background (security managers), some of their comments overlapped, though expressed differently. Therefore, we summarize the interviewees' comments and feedbacks and list the key information from the interviews, as shown in section 8.3.2.

Limitation: Using the expert judgment approach for validating the usefulness of our models also has drawbacks. Game theory is quite a new thing for chemical security managers. The interviewees had to spent a lot of time to read documents before the interview and concentrate their attention during the interview, to understand our game theoretic models. The interviewees were very cautious to give feedback before they believed that they understood the models correctly.

8.3.2 Results

All of the interviews were recorded. All the interviewees believed that by explicitly modelling the intelligent interactions between the defender and the attackers, the proposed models have the potential to improve the security of their sites. All six interviewees agree that assumptions and hypothesis which are the foundation of the present dissertation, such as i) attackers are adaptive; ii) attackers would collect information before their attack; and iii) quantitative models are needed for security risk assessment, reflect the reality of security management. All interviewees think that the models can be implemented on their sites if the comments that they provided would be considered. The interviewees' suggestions for improving the models are listed below:

1) Domino effects assessment in a security context should gain more attention

Chemical industries are critical infrastructures, not only because of their importance to society (via the production chain) or because of the existence of flammable/explosive/toxic materials, but also because of the potential propagation of domino effects, which may worsen the impact of an initial event. The propagation of domino effects in the chemical industry has been well studied by the safety research community. [4] Domino effect in a safety context is defined as a series of accidents caused by an initial accident via escalation vectors such as heat radiation, overpressure, or fragments. Therefore domino effect assessment models in a safety context are built upon a hypothesis that the propagation of the event chain starts from one initial event. This is a reasonable assumption since independent simultaneous failures at multiple installations is extremely unlikely if not impossible.

Conversely, in a security context, malicious attackers may implement an attack at several installations at the same time, resulting in a more serious cascading event. Therefore, although safety risks and security risks share some similarities on the consequence aspect, new models are needed for the evaluation of domino effects in security events.

The interviews with the security experts clearly indicate that the experts are not interested in domino effects. When they make security decisions, they ask for accident consequence information from the safety department of the plant. This approach in itself may thus be improved since security-related accident consequences might be quite different from safety-related accident consequences.

2) CPP game can be extended to deal with inside attackers

The CPP game is developed based on the general physical intrusion detection system in chemical plants and we claimed in Chapter 4 that the CPP game only works for the prevention of external attackers. Interviewees think that the CPP game is also applicable for the prevention of internal attackers. van Lerberghe: "it (the general intrusion detection approach) is also applicable for internal (attackers), but then you (the attacker) start from an internal location (a certain zone)".

In the CPP game, the attacker is assumed as an external attacker without authority to enter the plant, and he therefore is restricted to start his attack from Zone 0 (i.e., outside of the plant). However, in industrial practice, employees or contractors (possible internal attackers) are not allowed to enter higher level zones than the zone where their work should be carried out. For instance, if a contracted driver is authorized to drive his truck into the plant to a loading point which locates in Zone level 1, then he will only have the badge to pass the first layer perimeter. In these cases, the intrusion detection system would also work for these internal attackers if they aim to attack a target in a higher level zone. While for attacking a target in the zone (or lower level ones) where he is authorized to enter, the probability of arriving at the target would then be 1 and the conditional probability of a successful attack (i.e., p_y in the CPP game) can be used to describe the possibility of being detected during the implementation of the attack.

Therefore, in the intrusion and attack procedure shown in Figure 4.2, the restriction that the attacker must start from zone 0 should be removed. Instead, the (internal) attacker may start from any zone. If he aims to attack a target located in higher level zones, he still has to pass the perimeters and he might be detected in the higher level zones, just like an external attacker. If he aims to attack a target located in the zone (or lower level zones) where he is allowed to enter, his probability of successfully arriving at the target can be set as 1.

3) Exit procedure should also be considered in the CPP game

In the CPP game, an attacker's strategy is modelled as a combination of the target, the intrusion path, and the attack scenario. However, van Lerberghe emphasized that "a terrorist's objective is to get to the location, and then he stops. Other adversaries (non-terrorist) are going inside the location, and they are going back outside. …. Which means that an adversary can also be intercept when he is going back.". This is included in the "PICER" principle who describes that an intrusion attack contains the following steps: i) Preparing for the attack, such as collecting information about the target plant; ii) Intruding the plant to reach the target; iii) Collecting the stuff or committing the attack; iv) Exiting the plant (for non-terrorist attackers); and v) Rewarding from the attack, e.g., sale the stolen goods. Furthermore, a thief is one of the major threats to chemical plants and the exit procedure is an important and difficult step for a thief.

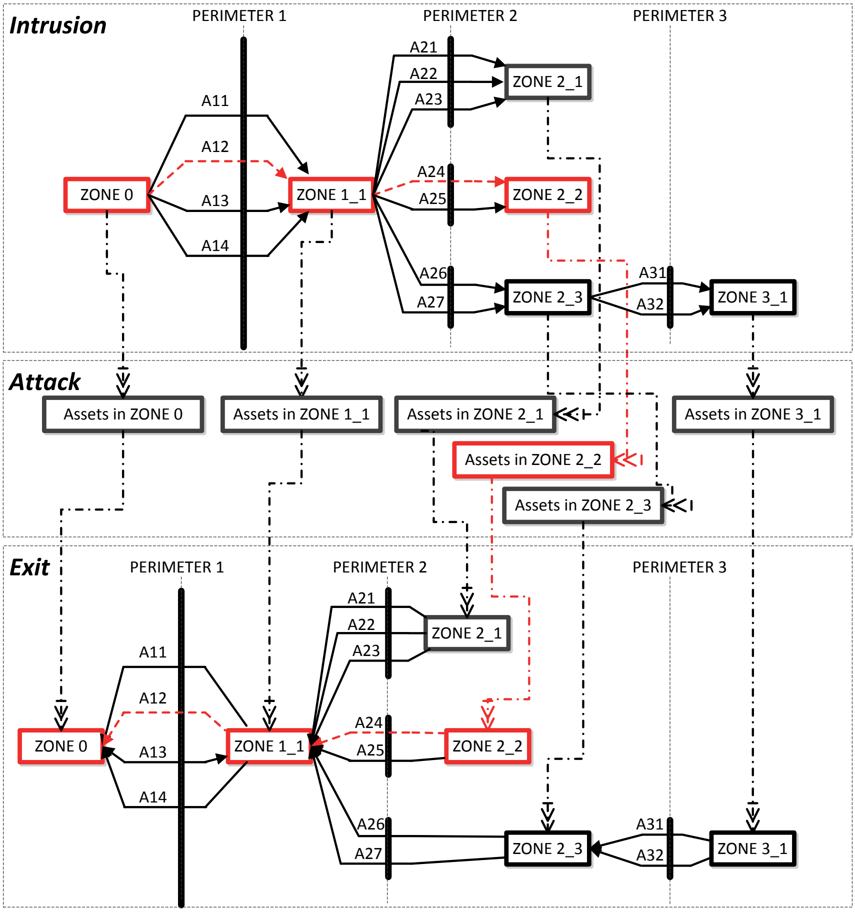


Figure 8. 1. The intrusion, attack, and exit procedure

The CPP game can easily be extended to take the exit procedure into consideration. Figure 8.1, revised from Figure 4.2, illustrates the intrusion, attack, and exit procedure of an attack. The exit procedure is added in the bottom of the figure. The red dotted route denotes that the attacker exits the plant from the route that he uses to intrude. This is the most likely scenario since the intruder would enter the plant from the most vulnerable path (in the attacker's perception, not necessarily the real most vulnerable path) and if he intrudes successfully, he confirms that the path is secure for him and therefore he would usually exit using the same route. However, the model does not limit the attacker's exit path to be the same as his intrusion path, to provide maximal flexibility.

If the exit procedure would be considered in the CPP game, then we must distinguish suicide attackers from other attackers. A suicide attacker such as a terrorist aiming to cause an explosion only is interested in the intrusion and attack steps. Other attackers, e.g., thieves, information spies, and terrorists who want to steal materials for further terrorist actions etc., are more concerned with the exit procedure. It is also worth noting that, in the intrusion step and in the exit step, the probabilities of passing the same entrance or the same zone can be different. For example, if a contractor wants to steal some material, then the probability of a successful intrusion will be much higher than the probability of exiting the plant with the materials.

4) Vulnerability to an attack scenario is either 0 or 1

In the CPP game, vulnerabilities are expressed as a probability range from 0 to 1. The probability should be provided by experts, or in theoretical research, can be calculated by the contest success function (CSF). (5) However, several interviewees argue that in security practice, vulnerabilities are either 0 or 1.

In current practice, security management is based on principal scenarios. A group of security staff sits together. Possible attack scenarios to the plant are discussed and the current defence plan is evaluated to see if it is sufficient enough to defend against the possible attack scenarios. In current practice, an attack scenario can either be definitely stopped (thus the vulnerability is 0) or not (thus the vulnerability is 1). For instance, if the plant has the regulation that all the trucks that leave the plant should be checked, then a scenario of stealing materials and transporting these materials by truck would never be possible. Conversely, if there is one entrance that trucks can leave without being checked, then the same scenario would always be successful.

Vulnerability in industrial practice comes from the result of an incomplete set of attack scenarios or from a shortage of security budget. Security managers list several principal scenarios. If the current defence plan is not efficient enough to deal with these scenarios, a recommendation of implementing new countermeasures will be given. Before the implementation of the new countermeasures, the vulnerability to certain scenarios is 1 while after the implementation, the vulnerability becomes 0. Furthermore, if the attacker attacks the plant by a scenario from the principal scenario set, then the vulnerability of the plant is 0, while if the attacker attacks with an unexpected scenario, the vulnerability will be 1. A further example can be used to clarify the idea: deploying a badge reader and a tyre killer at the main entrance of the plant can definitely prevent a force intrusion of a truck from the entrances, however, it does not work if the attacker firstly steals an authorized badge from an employee and then drives normally into the plant.

Security defence in practice is like an arms race: the defence aims to reduce the attacker's possible (cheap and easy) attack means, increasing the cost and difficulty level of an attack. However, if the attacker has the capability to implement an attack that is beyond the defence level, then the vulnerability of the plant would be 1.

To this end, the CPP game has the drawback of expressing the vulnerabilities as a probability. Game theory can be used to study the arms race between the defender and the attacker, balancing the defence cost and the security level.

5) Human resources are expensive and bribe would happen in regions where human resources are cheap

The CCP game and PPG model are developed for scheduling patrolling. All six interviewees agree that the models are interesting. Holzer argues about the price of using human patrollers and he mentioned "Using security agents for patrolling is expensive and we always have to deal with the "human factor" (laziness, inattention, wrong decisions, risk of bribing, illness etc.)".

In industrial practice, the cost of hiring security staff consumes the majority of the total security budget. Moreover, using security staff for patrolling always needs dealing with the "human factor" (laziness, inattention, wrong decisions, bribing, illness …). Dealing with these human factors is a huge management task itself. On the contrary, the technical means are more stable and easier to ensure their effectiveness. The cost of technical means (e.g., camera monitored fence) is becoming cheaper and cheaper while their functions are becoming more and more reliable.

In principle, technical equipment is employed wherever there is a need for a permanent, reliable surveillance, e.g., for perimeter detection, intrusion alarms, etc. The limited number of security staff members (due to the limited budget) are employed to do jobs that technical systems cannot do, such as sitting in the control room to receive and deal with alarms triggered by technical systems, while sending them to do a patrol job has a very low priority.

In countries where human resources are comparably cheaper than technical means, patrolling can be a mean to be used. For instance, chemical sites in Southeast Asia regularly substitute CCTV-cameras with human security agents, because with the total cost of one camera for one year five or six security agents can be hired. However, the risk of bribing in these cases is quite high. A collusion of the external attackers and the employed patrollers can easily avoid all other security countermeasures and successfully implement an attack.

In a nutshell, the interviewee argues the necessity of using patrols. Indeed there is patrolling in chemical plants. These patrollers are however aiming at safety/security inspections and aiming to provide quick responses, instead of for detection purpose. However, the patrolling games are still interesting for industrial practice, in two aspects: i) the model can be used for optimizing robotic patrolling; ii) in some special period of time (during an election or some sensitive time), extra patrolling can be temporarily scheduled.

6) Patrolling for multiple objectives: detection, response, and inspection

"In weekends and night, they (the patrollers) patrol the fence of the site. In the day time, they have also first-aid (tasks), when there is a fire or a first-aid case, they are also equipped to help them. In the weekend, they do extra jobs to visit offices." (Brummelhuis)

The developed patrolling games are single objective oriented, that is, for detecting possible attackers. However, as we also mentioned before, patrolling in current industrial practice is for the purpose of response and equipment/building inspection. In order to convince the industry department to implement the developed patrolling games, we must integrate multiple objectives into the model, namely, the objectives of detection, response, and inspection.

Patrol for security detection has been clearly explained in Chapter 6 and 7. Patrol for response is used in the case that an alarm is triggered while the intruder is not well traced. For instance, an intruder climbs over a fence and the camera detects the intrusion and an alarm is given. The response team, however, can only arrive at the intrusion point several minutes later, and when the team arrives, they cannot find the intruder. In this case, the response team has to search the area around the intrusion point and find the intruder. Furthermore, knowing that the intruder will not be well tracked, the response team may optimize its route to go to the intrusion point, to minimize the time needed to catch the intruder and taking into consideration the intruder's adaptive (to the response team's route) behaviour.

Meanwhile, a patrolling team may also be required to check certain equipment or a building within a certain time period. This is for both safety and security purposes. On the one hand, companies want to ensure that the equipment is working well and therefore they ask the patroller to check it. A very normal task can be, for example, to check whether all the lights are turned off and whether all the windows are closed. On the other hand, inspection of these buildings can deter possible intentional malicious behaviours and provide a quick response if something happened.

7) Human patrollers would not like to follow the patrolling routes generated by a model

Some interviewees point out that (experienced) patrollers would not be happy if a system is going to tell them how to perform a patrol. For example, Weijers pointed out that "Experienced patrollers would not like to obey a computer system to tell them how to perform a patrol". Patrollers have their own patrolling habit. Although we can theoretically prove that their fixed or purely randomized patrolling routes are not optimal, however as long as no serious security attack happens, it is difficult to ask the patrollers to change their working mode to obey a mathematical model. Furthermore, even if an attack happens (e.g., a chemical site frequently loses materials), it is difficult to clarify that it is the patrollers' responsibility.

We may record the patrollers' daily patrolling routes (time and location). By analysing these data, we may see whether the patrollers fall into a fixed patrolling route. A possible situation is that the patrollers think they are patrolling randomly, but they actually are following a fixed patrolling route, being self-cheating. Furthermore, if they are not using fixed patrolling routes, we may see whether they visit more hazardous targets more frequently. In this way, we aim to prove that the routes generated by the models are randomized and cover more on vital assets, when compared to the current patrolling.

8) Patrolling can easily be interrupted by a fake attack and therefore fail to prevent a real attack

"How would I be able to notice diversion attempts – for instance someone triggers the awareness of your patrols so that another team can enter on a different place?" (Holzer)

A not-so-stupid attacker would firstly trigger an alarm to attract the attention of the patrollers and then implement a real attack somewhere else by himself or by his accomplice. Patrolling has difficulties in dealing with this kind of "cheating-scenarios". The patrolling models must be further improved to fill this gap.

9) When an attacker attacks, his perception of the success probability is 100%

In the game theoretical models developed in this dissertation, we do distinguish between the differences of the attacker's and the defender's perception of the success probability. In Chapter 4, we show that a risk-seeking attacker may over-estimate the success rate while a risk-averse attacker may under-estimate the probability. However, in this dissertation, the attacker is assumed to implement an attack if the successful rate multiplied by the potential gain (consequence for the defender) is large enough.

van Lerberghe argues that "When an attacker starts his attacks, he is convinced that his success rate is 100%. Many people think it is lower." An attacker would not attack unless he thinks the success rate is 100 percent. It is not necessary that the rate is really 100 percent (remember that the real success rate can be different from either the attacker's or the defender's perception, and it remains unknown). For instance, the defender may have implemented countermeasures to protect a target and the attacker does not know the existence of these countermeasures. In this situation, the real success rate of an attack by a certain scenario can be 10%, while the defender may think the rate is 0%, and the attacker will treat the rate as 100% and subsequently, implements an attack.

This comment about "a 100% success rate of an attack" is a consistent comment when compared to the comment that "vulnerability can only be 0 or 1" (see section 8.3.4). An attack will happen only when the attacker finds a vulnerability of the defence system and the success rate of the attack is 100% from the attacker's point of view while the vulnerability will be 1 from the defender's perspective. The defender's attention should therefore be given to find out the possible vulnerabilities of the current system and fix them, to reduce the vulnerability from 1 to 0.

10) The necessity of protecting European chemical sites from terrorist attacks should be better demonstrated

Several types of threats to the chemical industry are clarified in this dissertation, namely, thieves (external or internal), environmental activists, information spies, terrorists (domestic or international), and sabotages. Another threat that often happens in practice is workplace violence, i.e., two or more employees or contractors fighting with each other in the plant, and the security department has to deal with this situation. Among these threats, our main attention is given to terrorists. However, interviewees point out that thieves (loss of assets) rather than terrorists pose the main threat to their sites. Chemical sites in the Netherlands and Belgium are not attractive to terrorists (or are not deemed to be by the interviewees).

For instance, Brummelhuis mentioned that "at the moment in Holland, the threat (of terrorist) to a chemical site or refinery is low, to soft targets is high". Meanwhile, he also pointed out that "… and terrorist, until now, we are lucky though, but at the moment, it can happen". In Chapter 2, we emphasized that the occurrence probability of a security event cannot be predicted by historical data. The World Trade Center in New York had never been threatened by a terrorist using an airplane and later on, the building was entirely destroyed by an attack, leading to incalculable losses. Nevertheless, historical data analysis is still listed as a method of analysing security threats in some security risk assessment methods in the chemical industry, such as in the API SRA document. [3] Up to now, there has been no serious terrorist attack successfully conducted at European chemical sites. Therefore, it is reasonable that security managers in chemical plants pay less attention to the threat of terrorists and more attention is given to the prevention of loss of assets, which are frequently happening. However, the two attacks which happened in France reveal the possibility of a successful attack in a European chemical plant.

To prevent a terrorist attack is also very challenging. All interviewees think that the defence at their site can simply be broken if the attacker has some advanced weapons or if the attacker is a bit professional. If the chemical plants are ambitious to really protect their plants from a terrorist attack, the security budget needs to be quite high. Conversely, the security budget in some European chemical sites is decreasing, as explained by Weijers. Therefore, practitioners prefer to spend the limited money to the prevention of the frequently happening security events (e.g., loss of assets), instead of to the prevention of the high impact but less likely events, or terrorist attacks. Furthermore, the implemented countermeasures increase the difficulties of conducting an attack, although they are not efficient enough for preventing a well-planned (e.g., armed attackers) terrorist attack. In this way, the practitioners believe that the terrorist attackers, if they really exist in Europe, would be more interested in attacking soft targets, such as airports, government departments, schools etc.

8.4 Conclusions

Developing a model is easy while validating it or demonstrating its usefulness is difficult. This dissertation aims at improving security in chemical areas by using game theory. Several models are proposed in previous chapters and this chapter discusses the validation of these models.

General model validation methods, namely, the benchmark comparison approach, the red team experiment approach, the simulation gaming approach, and the expert judgment approach, were introduced and compared. We find it difficult to solidly validate our models. Therefore, the expert judgment approach was chosen to validate our model.

Six senior security experts from Sitech, Solvay, Antwerp port, BASF SE, Shell, and CIMIND were interviewed. The DAMS model, the CPP game, the CCP game, and the PPG model were introduced to these experts and their opinions about the implementation of these models in reality are collected and shown in this chapter. All interviewees agree that the proposed models successfully capture the key characteristic of security risks, that is, the attackers are intelligent and they will adapt their attack strategy according to the defender's plan. However, there are gaps between the models and industrial practice. These gaps are: domino effects caused by intentional behaviours are not a concern (yet) for security managers; the CPP game should be extended to take insiders and the exit procedure of an attack into consideration; vulnerability should be represented as a binary number rather than a probability, while it is either 0 or 1 depending on whether the defender successfully considers the attack scenario; the necessity of scheduling security patrolling and the objectives of patrolling; a fake attack can easily attract the patroller's attention and afterwards a real attack can happen; and the main threat to chemical sites are thieves instead of terrorists.

References

[1] Box GE. Science and statistics. Journal of the American Statistical Association. 1976;71(356):791-9.

[2] Carson I, John S, editors. Verification validation: model verification and validation. Proceedings of the 34th conference on Winter simulation: exploring new frontiers; 2002: Winter Simulation Conference.

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

[4] Reniers G, Cozzani V. Domino Effects in the Process Industries: Modelling, Prevention and Managing: Elsevier B.V.; 2013. 1-372 p.

[5] Skaperdas S. Contest success functions. Economic theory. 1996;7(2):283-90.

9

EPILOGUE

This chapter summaries the dissertation. Conclusions, recommendations, and future research directions, are clarified and given.

9.1 Conclusions and recommendations

Chemicals-using industries have an important role in modern society for providing the basic ingredients (fuels, chemicals, intermediates and consumer products) for our modern day lives and luxury. However, they also pose huge threats to society due to the mere use and storage of large amounts of hazardous materials with sometimes extreme processing conditions. The prevention of unintentionally caused events, which is the field of occupational safety and process safety, has been significantly improved in the process industries. Conversely, the physical protection of chemical plants and areas from malicious attacks, being the field of physical security, has not received enough attention yet by both academic researchers and industrial practitioners.

Several qualitative and semi-quantitative security risk assessment methods have been published. For instance, the Security Risk Factor Table (SRFT) and the American Petroleum Institute recommended standard on "Security Risk Assessment Methodology for Petroleum and Petrochemical Industries" (the API SRA). These conventional security risk assessment methods, though having been extensively used in industrial practice in the United States, have the drawback that they are not able to consider intelligent interactions between defender and the potential attackers.

To counter the current disadvantage of security risk assessments, we introduce game theory as a decision-support mathematical approach for managing security in chemical industrial areas. The Chemical Plant Protection (CPP) game, which purpose it is to optimally set security alert levels at every entrance and every zone in chemical plants, the Chemical Cluster Patrolling (CCP) game, which can be employed to randomly but strategically schedule security guard patrolling among different plants, and the Pipeline Patrolling Game (PPG), which can optimize pipeline patrolling, are elaborated.

Nine conclusions are formulated. Conclusions 1, 2, 3, 4, and 6 are the answer of research question SRQ1. Conclusions 5, 7, and 8 answer SRQ2. Research question SRQ3 is answered by conclusions 8 and 9. Nine recommendations are given based on the conclusions that we draw.

Conclusion 1

When analysing the domino effect in chemical industries, the failure evaluation of a single unit or target is already a complex task, involving the assessment of the target response to the escalation vector: overpressure, fragment damage, or heat radiation. The complexity is increasing when dealing with the analysis of the domino propagation among multiple units. The agent-based modelling and simulation approach, being a bottom-up approach, is suitable for modelling the domino effects.

Conclusion 2

Conventional security assessment methods, such as the SRFT and the API SRA, are mostly developed by senior security experts with plenty of experience and expertise on physical security policies and management. Therefore, these methods have the advantage of being practically implementable in industrial practice and of being understandable and used by practitioners. However, these methods are mainly qualitative or semi-quantitative, and are not able to provide adequate information for decision makers to quantitatively allocate security resources; we refer for this observation to arguments given in Cox [1, 2]. Furthermore, failing to model the intelligent interactions between defender and attackers, these conventional methods may lead to incorrect results with respect to the allocation of security resources; see also further arguments in Powell [3].

Game theory, conversely, developed by mathematicians and economists, is quite abstract to the practice of chemical security management. However, game theory has the advantage on modelling strategic decision making in a multiple players setting and on providing quantitative results as output information. Several game theory-based security systems have been developed and implemented, such as the ARMOR system for the Los Angeles airport, the PROTECT system for the US coast guard, and the IRIS system for the Federal Air Marshal's service etc [(4)]. In this dissertation, we developed such an analogous game-theory-based security system for the chemical industry.

- **Recommendation 1**

To improve security within the chemical industry, conventional security risk assessment methods and game theory need to be integrated. In the integrated framework, game theoretical models need to be provided with inputs from conventional methods, and game theoretical results need to be 'translated' to industrial practice.

Conclusion 3

There are many types of security countermeasures. Even in the situation of a limited budget, the defender can still combine several countermeasures, in order to secure her assets. In conventional security risk assessment methods, the effectiveness of a bundle of countermeasures is not assessed. For instance, in the API SRA methodology, the SRA team only re-estimates vulnerabilities and consequences presuming that one proposed countermeasure is implemented (see Form 6 in the API SRA document [(5)]). However, synergistic effects of multiple countermeasures should not be under-estimated. An example of a synergistic effect is the combination of a camera system and having fences. Cameras without fences or fences without cameras are much less efficient than both together.

- **Recommendation 2**

Risks reductions by bundles of countermeasures should also be estimated. In fact, there might be a great number of such bundles. For instance, in case of a recommendation of in total 10 countermeasures and bundles existing of two countermeasures, there can be $\boldsymbol{2^{10} = 1024}$ of these bundles of countermeasures. However, the number can be significantly reduced by budget constraints as well as by using field knowledge.

Conclusion 4

As already mentioned, an important challenge of assessing and managing security risks in chemical plants is that the defender deals with intelligent adaptive adversaries. To fight with these intelligent attackers, the defender should not only pay attention to her own interests, but also study the attacker's interests, since intelligent attackers may exhibit a high probability to attack a target which from a safety viewpoint is quite safe.

Conventional security risk assessment methods, however, mainly focus on the defender's interests and implicitly assume that the attacker has opposite interests to the defender. Nonetheless, potential attackers within the chemical industry are various and different attackers have different goals. Therefore, it is not necessary that attackers always have an opposite interest to the defender.

- **Recommendation 3**

In a security risk assessment procedure, attention should be paid to the data assessment from the adversaries' viewpoint. Putting "the defender's feet also in the attackers' shoes" can be helpful for security management.

Conclusion 5

In a 'deep uncertainty' case, which means that the defender has huge uncertainties on the attackers' interests, rationalities, and capabilities etc., the defender is better off if she minimizes her worst/maximal loss. Therefore, although we conclude in *conclusion 3* that the defender should pay attention to learn the attackers' interest, in the current stage, if the learning is too difficult (e.g., due to the lack of reliable data), ignoring the attacker's interests can be a feasible solution for the defender.

- **Recommendation 4**

Even from a game theoretic point of view, conventional security risk assessment methods have their rationales on implicitly assuming that the attackers have opposite interests to the defender's interest. Due to the difficulties of obtaining knowledge and data about the attackers, huge uncertainties of the attackers may exist. In this case, the defender is secure to play her MiniMax solution, which is also the optimal solution in a zero-sum game.

Conclusion 6

Risk scoring methods are still extensively used in security risk assessment procedures, after being proved theoretically incorrect. Moreover, on the one hand practitioners say it is difficult to obtain quantitative data, while on the other hand in some qualitative methods, the security risk management team decides a security risk score based on quantitative descriptions. An example of such practice can be found as Table 4.5 in Chapter 4.

- **Recommendation 5**

The security risk assessment team should work on quantitative data directly, instead of transferring these data into scores. Quantitative data extracted from industrial practice are often associated with uncertainties, for instance, instead of knowing an exact number of the consequence of a certain event, it is more likely that we know a lower and an upper bound of the value of the consequence. Current game theoretical models are able to deal with data with this type of uncertainties, (see for instance Chapter 5 of this book) and should therefore be used in security risk assessment.

Conclusion 7

In case that the attackers' interests are not strictly opposite to the defender's interests, which means that the security game is not a strategic zero-sum game, then the defender's payoff from a sequential game is higher and more stable than her payoff from a simultaneous game. Otherwise, if the defender believes that the attackers always have strictly opposite interests to her, then it does not matter whether the game is played sequentially or simultaneously.

- **Recommendation 6**

In a situation that the security information of a chemical plant is publicly known (thus the common knowledge assumption of a game can easily hold), then for defending those premediated attackers (premediated attackers are more

likely to be strategic attackers, e.g., an ISIS terrorist), industrial managers are suggested to make their security plan public, to deter and stop those attackers.

Conclusion 8

Being mathematically complicated and being too abstract for industrial practice prevent game theory to be more attractive to, and popular among, industrial practitioners. As we may notice from Chapter 4 to Chapter 7 in this book, game theory uses plenty of mathematical formulas and numbers, and regretfully, at least for optimal decision-making support, chemical security related terminologies (e.g., assessing vulnerabilities, threats, etc.) does not. Industrial practitioners doubt the usefulness of these formulas and the practical meanings of these numbers.

Furthermore, the correctness of results from game theoretic models strictly relies on the assumptions that the modeller uses. Some assumptions used in game theoretic models are quite unrealistic, e.g., the 'common knowledge' assumption. Therefore, industrial practitioners doubt the correctness of game theoretic results.

- **Recommendation 7.1**

User-friendly interfaces should be developed for game theoretic models. With the interface, a security risk assessment team can use game theoretic models as a black-box tool, and this way, it is possible for security managers to only have to provide the black-box tool with input data and afterwards to analyse the outputs of the tool.

Figure 9.1 (an extension of Figure 2.6 in Chapter 2) shows an extended framework of integrating conventional security risk assessment methods and security game theory. In the first step (L1), the security risk assessment team should evaluate what kind of threats the plant is faced with. Moreover, based on the current information and the team's judgements, the team should estimate whether these potential attackers are rational players or not, and they should estimate how much information the team has about the attackers. In the second step (L2), the team chooses a proper security game model from the so-called "security game model library" and learns what kind of input data is needed for the chosen security game model. In the third step (L3, L4), the team extracts the needed input data, by using a conventional security risk assessment method, the API SRA, for instance. In the fourth step (L5), the team simply runs the chosen security game model without necessarily knowing the details of the model. In the fifth step (L6, L7), the team translates the outputs of the chosen security game model into implementable recommendations.

In Figure 9.1, steps L1, L2, L4, and L7 are closely related to the practice of industrial security, and therefore they can be carried out by a security risk assessment team independently. Steps L3 and L6 should be done cooperatively by an SRA team and a security game developer. In step L3, the game developer informs the SRA team what kind of data is needed and what are the meanings of the data. In the meantime, the SRA team judges whether the data is achievable. If the answer is 'yes', then the game developer and the SRA team discuss the data structure of the inputs, while if the answer is 'no', then the game developer must revise the security game to be able to deal with achievable data. In step L6, the SRA team and the game developer discuss what kind of outputs are meaningful and how to build the map between the game outputs and the implementable

recommendations. Step L5 concerns purely game theoretic calculations, and the SRA team should not pay attention to this step.

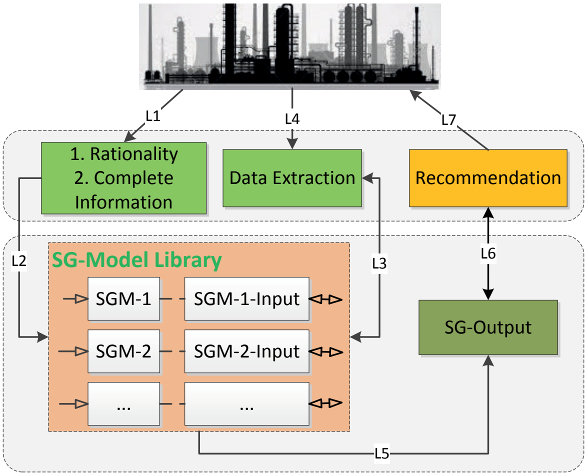


Figure 9. 1. An extended framework of integrating conventional security risk assessment methods and security game

In summary, the bottom grey part of Figure 9.1 should be a black-box for the SRA team.

- Recommendation 7.2

Game theoretic models for dealing with various uncertainties should be developed. In other words, the SG-Model Library in Figure 9.1 should be complete, to make sure that whatever the result of 'L2' is, a security game model always exists.

Fortunately, developments on computational game theory have provided models and algorithms for studying games played by bounded rational players and games where 'common knowledge' does not hold. Figure 9.2 (adopted from Zhang and Reniers [6]) shows the uncertainty space of the Chemical Plant Protection game (CPP game) [7]. The origin point is the CPP game with rational players and common knowledge assumptions. The x-axis represents the attacker's rationality, such as the epsilon-optimal attackers, quantal response attackers, etc. The y-axis denotes the defender's uncertainty on the attacker's payoffs, such as the discrete uncertainty, Bayesian uncertainty, interval uncertainty, etc. Each point in the uncertainty space corresponds to a realistic situation and a cluster of models and algorithms. If the uncertain space of the Chemical Cluster Patrolling (CCP) game would be plotted, a third dimension named "uncertainty on the attacker's observation" should also be added.

The output of 'L2' in Figure 9.1 decides a coordinate in Figure 9.2. Therefore, models and algorithms should be developed for all the meaningful coordinates in Figure 9.2. To achieve this goal, models and algorithms for dealing with combinations of multiple types of uncertainties need to be enhanced. There are abundant studies on dealing with a single type of uncertainty, i.e., points on axis in Figure 9.2. However, in reality, a defender often faces multiple types of

uncertainties, e.g., point #1 in Figure 9.2 represents multiple types of attackers and each type of attackers are epsilon-optimal players [(8)].

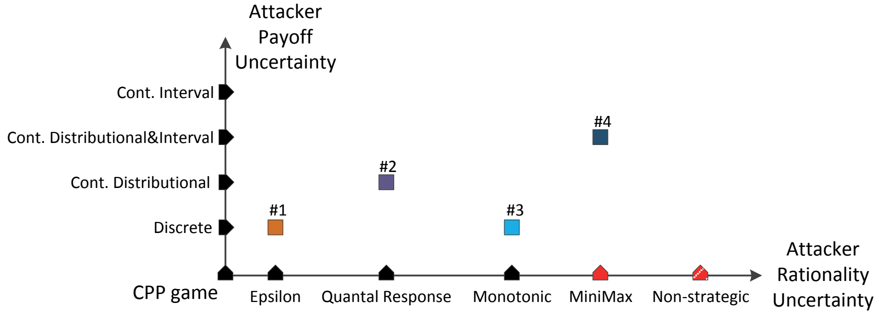


Figure 9. 2. Uncertainty space for the CPP game

Conclusion 9

A purely randomized patrolling route or a fixed patrolling route does not make best use of the security guard patrolling team. A purely randomized patrolling route fails to cover more hazardous plants or pipeline segments more frequently. The downside of a fixed patrolling route is that the patroller's position may be predictable to an attacker. Game theory can therefore be used to generate random (thus being unpredictable) but strategic (thus patrolling higher hazardous plants/segments more often) patrolling routes.

- **Recommendation 8**

Security patrolling in current industrial practice should be re-thought and re-conceptualized by using game-theoretical models.

9.2 Future research

Security risk, although having been recognized since thousands of years ago (e.g., the use of fences), still needs more research efforts. New technologies such as cameras, advanced detection machines, and even drones etc. are being employed for better securing chemical plants. However, while the defenders are progressing on defence, adversaries are also advancing their attack scenarios, making the defence-attack procedure a dynamic interaction and the only way of remaining secure is to be always ahead of the adversaries.

To the best of our knowledge, this dissertation is the first research that employs agent-based modelling and simulation (ABM&S) for assessing domino effects in the chemical industries and applies game theory (GT) for bettering the protection of chemical facilities. Nonetheless, the DAMS model, the CPP game, the CCP game, as well as the PPG are still on-going research. Future research can be conducted from several directions.

Integrating preventive barriers into consideration and developing a user-friendly interface, are interesting extensions of the DAMS model. Among others, Landucci et al. [(9)] studied the performance of different types of barriers in a domino effect triggered by fire. Safety barriers are actually also working on individual equipment, for instance, by cooling down a pressurized tank (Water Deluge System). Therefore, the DAMS model can also be applied to estimate the performance of safety barriers. If a user interface would be developed, users can drag the models from the left-hand column to the main window. The user may also edit the models in the main window. After deploying the model, the user may run the model by simply clicking a button. Furthermore, result analysis functions could also be integrated to the interface.

Although the CPP game has been extended to deal with the defender's uncertainties about the attackers, yet the model cannot handle multiple monotonic maximal attackers. In Chapter 5, the MoSICP solution takes into consideration both the defender's distribution-free uncertainties about the attacker's parameters and the defender's uncertainties about the attacker's rationality. However, the MoSICP solution fails on considering multiple types of attackers. A more realistic solution, which has the advantages of MoSICP and is able to take into account multiple types of attackers, should be further defined for the CPP game.

Observation errors and implementation errors should be modelled both in the CCP game and in the PPG. The patrolling games generate randomized patrolling routes and the randomized routes mean that the patroller must go to a certain node by a certain probability. However, if the attacker learns the patroller's route by long-term observations, he would not be able to get the exact probabilities. Moreover, the defender cannot perfectly obey the generated probabilistic routes. For instance, the patroller may have to go to toilet or the patroller may be delayed by a traffic light. There are researches concerning the observation errors and implementation errors in patrolling games, see for instance, Nguyen et al. (10). However, those models suffer from high computation complexities.

In Chapter 8, we mentioned that patrolling can easily be interrupted by a fake attack and therefore the attacker may implement a real attack somewhere else either by himself or by his accomplice. New patrolling models and algorithms should be developed for dealing with this situation.

Last but not the least, game theory is an idealistic method on modelling strategic decision making in a multiple players environment, while the security practice is full of complexity. Therefore, as we also stated in recommendation 7.2, more game theoretical models considering multiple types of uncertainties should be developed. Being able to handle all kinds of uncertainties, game theoretical models should then be advertised to industrial security managers, for bettering the protection of chemical facilities.

References

[1] Cox Jr LAT. Some limitations of "Risk= Threat× Vulnerability× Consequence" for risk analysis of terrorist attacks. Risk Anal. 2008;28(6):1749-61.

[2] Cox L. What's wrong with risk matrices? Risk Anal. 2008;28(2):497-512.

[3] Powell R. Defending against terrorist attacks with limited resources. American Political Science Review. 2007;101(03):527-41.

[4] Tambe M. Security and game theory: algorithms, deployed systems, lessons learned: Cambridge University Press; 2011.

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

[6] Zhang, Reniers. Applying game theory for adversarial risk analysis in process plants. In: Reniers G, Khakzad N, van Gelder P, editors. Security risk assessment and management in the chemical and process industry: De Gruyter; 2018.

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

[8] Pita J, Jain M, Tambe M, Ordóñez F, Kraus S. Robust solutions to Stackelberg games: Addressing bounded rationality and limited observations in human cognition. Artificial Intelligence. 2010;174(15):1142-71.

[9] Landucci G, Argenti F, Spadoni G, Cozzani V. Domino effect frequency assessment: The role of safety barriers. J Loss Prev Process Ind. 2016.

[10] 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.

APPENDIX

A Calculation of ttf

The dynamic evolution of tanks exposed to fire is a complex problem, which would need a dedicated assessment through physical models, especially contemplating the possible synergistic effects due to increment of heat load during fire exposure. Some simplifying assumption were needed to instruct model and simulations.

Assuming a tank receives Q_i at time $t_i(i = 1,2,3, \dots, T)$, the following text discusses how to compute the time t_r that the tank would employs the vulnerability model to judge whether being damaged or not.

If $T = 1$, which means the tank only receives one time the heat radiation, then the formula A.1 and A.2 can be employed to compute the ttf,[33] for atmospheric and pressurized equipment respectively. In these cases, we have $t_r = t_1 + ttf_1$.

$$\ln(ttf) = -1.13 \cdot \ln(Q) - 2.67 \cdot 10^{-5}V + 9.9 \qquad \text{(A.1)}$$

$$\ln(ttf) = -0.95 \cdot \ln(Q) + 8.845V^{0.032} \qquad \text{(A.2)}$$

If $T > 1$, which means the tank receives heat radiation from more than one sources. Note that A.1 and A.2 can be re-written as:

$$ttf = Q^{-1.13} \cdot e^{-2.67 \cdot 10^{-5} \cdot V + 9.9} \qquad \text{(A.3)}$$

$$ttf = Q^{-0.95} \cdot e^{8.845 \cdot V^{0.032}} \qquad \text{(A.4)}$$

Furthermore, we define $k = 1.13$ and $g(v) = e^{-2.67 \cdot 10^{-5} \cdot V + 9.9}$, then A.3 can be re-written as:

$$ttf^{\frac{1}{k}} \cdot Q = g(v)^{\frac{1}{k}} \qquad \text{(A.5)}$$

A.4 can be re-written in the same format, by using different $k, g(v)$.

A.5 is a linear equation for the heat radiation Q, thus it can be used to compute the t_r. According to the physical meaning of ttf, A.6 and A.7 computes the t_r when the tank receives multiple times heat radiation.

$$\sum_{i=1}^{N} (t_r - t_i)^{\frac{1}{k}} \cdot Q_i = g(v)^{\frac{1}{k}} \qquad \text{(A.6)}$$

In condition of that:

$$t_r^{N-1} > t_N \qquad \text{(A.7)}$$

In A.6 and A.7, the t_r^{N-1} denotes the root of A.6 when $N - 1$ times heat radiation received.

The explanation of A.6 and A.7:

Recalling the A.5, given a target tank, the right part of A.5 is a constant number, while the left part can be re-written as $(t_r - t)^{\frac{1}{k}} \cdot Q$, showing the relationship of the time needed to heat up the tank and the heat radiation received. Furthermore, A.5 is a linear equation of Q, so that the addition law can be used, reaching the A.6. Formula A.7 further explains only when the tank is not heat up yet, then it is necessary to use A.6.

Figure A.1 gives the procedure of how to use A.6 and A.7.

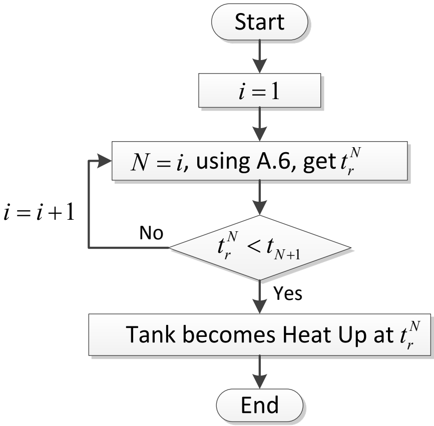

Start
$i=1$
$N=i$, using A.6, get t_r^N
$i=i+1$
No
$t_r^N < t_{N+1}$
Yes
Tank becomes Heat Up at t_r^N
End

Figure A.1 Procedure of Computing t_r.

Observation A.1 In condition of A.7, formula A.6 has and only has one root between time interval (t_N, t_r^{N-1}).

Proof: Define $f(t_r) = \sum_{i=1}^{N}(t_r - t_i)^{\frac{1}{k}} \cdot Q_i - g(v)^{\frac{1}{k}}$. Firstly, $f(t_r)$is an elementary function, and $f(t_r)' > 0$ when $t_r \geq t_N$; Secondly, $f(t_r) = \sum_{i=1}^{N}(t_N - t_i)^{\frac{1}{k}} \cdot Q_i - g(v)^{\frac{1}{k}} = \sum_{i=1}^{N-1}(t_N - t_i)^{\frac{1}{k}} \cdot Q_i - g(v)^{\frac{1}{k}}$, considering that $f(t_r)' > 0$ and $t_N < t_r^{N-1}$, we have $f(t_r) < \sum_{i=1}^{N-1}(t_r^{N-1} - t_i)^{\frac{1}{k}} \cdot Q_i - g(v)^{\frac{1}{k}} = 0$; Thirdly, $f(t_r^{N-1}) = \sum_{i=1}^{N}(t_r^{N-1} - t_i)^{\frac{1}{k}} \cdot Q_i - g(v)^{\frac{1}{k}} = (t_r^{N-1} - t_N)^{\frac{1}{k}} \cdot Q_N > 0$. combining the above 3 condition, we know that there exist and only exist one $\bar{t}_r \in (t_N, t_r^{N-1})$ s.t. $f(\bar{t}_r) = 0$.□

Based on the Observation A.1, Figure A.2 gives an approximately algorithm to solve A.6.

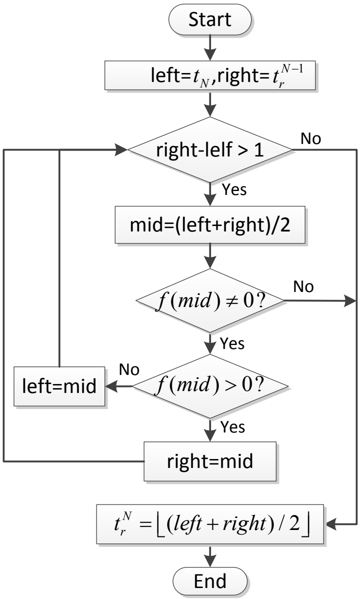


Figure A.2 A Binary Search Algorihtm for Solving A.6.

However, when using the Vulnerability Model to compute the probabilities of being damaged, formula A.1 and A.2 should be used: using $Q = \sum_N Q_i$ as input, getting the ttf, and use the ttf as the input for Equation (3) (in section 3.2.2).

B Post-release event tree

Figure B.1 reports the quantified event trees considered for the analysis.[46] Table B.I shows the exact parameters used for the ETA1 and ETA2 modules.

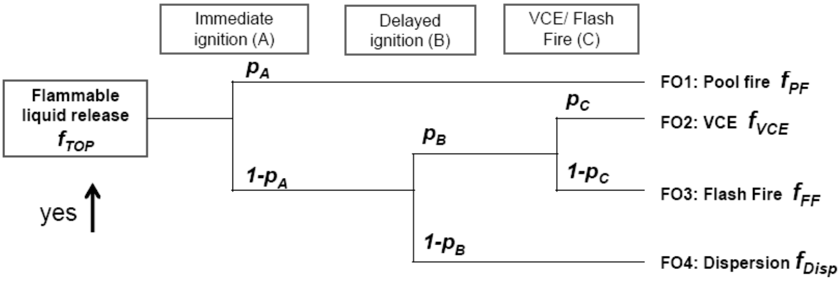


Figure B.1. Event Tree Analysis of Quantification in case of Flammable Liquid Releasing.

Table B.I. Parameters in Figure B.1

	p_A	p_B	p_C
$ETA1$	0.1	0	0
$ETA2$	0.6	0	0

Since ETA1 will only be used by the primary unit, which means at that time, there is no tank on fire yet, thus the immediate ignition probability p_A is set to be a smaller number, i.e. 0.1. The ETA2 will only be used by the domino units, which means there is one or more tanks on fire already, thus the immediate ignition probability p_A for ETA2 is set to be a greater number, i.e. 0.6. Both for ETA1 and ETA2, p_B is set to be 0, which means in this study, only the heat radiation escalation is considered, while the overpressure, fragment etc. are not considered. Since p_B is set to be 0, it does not matter what value the p_c is, specifically, we set $p_c = 0$.

C Number of replications and result accurancy

In this appendix, we will discuss how many replications are necessary for the computational experiment.

Table C.I Notations in this appendix

Notation	Definition
N	experiment times
p	the probability of the concerned event
v_N	the number of times that the concerned event happens, within the N times repeat experiments
$\tilde{p}$	$\tilde{p} = v_N / N$, the observed probability of the concerned event

Since every replications are independent, thus the N times experiments are Bernoulli process. According to the properties of Bernoulli process,(52) we have

$$E(v_N) = Np \tag{C.1}$$

And

$$V[v_N] = Np(1-p). \tag{C.2}$$

(C.2) is equal to

$$V\left[\frac{v_N}{N}\right] = \frac{p(1-p)}{N}. \tag{C.3}$$

Furthermore, according to the central limit theorem (CLT) and law of large numbers(52) we have

$$z = \frac{(v_N/N)-p}{\sqrt{V[\frac{v_N}{N}]}} \sim N(0,1). \tag{C.4}$$

Thus we have

$$P\left\{\left|\frac{(v_N/N)-p}{\sqrt{V[\frac{v_N}{N}]}}\right| \leq Z_{1-\alpha/2}\right\} = 1-\alpha. \tag{C.5}$$

Normally we set $\alpha = 0.05$ (i.e. 95% confidence interval), we have $Z_{0.975} = 1.96$. Considering to (C.3), (C.5) can be rewritten as:

$$P\left\{\left|\frac{(\tilde{p}-p)\sqrt{N}}{\sqrt{p(1-p)}}\right| \leq 1.96\right\} = 0.95 \tag{C.6}$$

Furthermore (C.6) can be rewritten as:

$$P\left\{|\tilde{p}-p| \leq \frac{1.96\sqrt{p(1-p)}}{\sqrt{N}}\right\} = 0.95 \tag{C.7}$$

$|\tilde{p}-p|$ represents the error of the observed probability and the real probability.

In this study, we are doing experiments under the condition that the primary unit is already on fire (as explained in section 5.1), thus the events whose occurrence probabilities are lower than 10^{-3} can be ignored. So it is acceptable if $|\tilde{p} - p| \leq 0.001$. Thus we need $\frac{1.96\sqrt{p(1-p)}}{\sqrt{N}} \leq 0.001$, getting:

$$N \geq 3.8416 \cdot p(1-p) \cdot 10^6 = N_{min} \qquad \text{(C.8)}$$

It is obvious that $N = 10^6$ satisfies the formula (C.8).

This means if we do $N = 10^6$ experiments, then all the conditional probabilistic results are 95% reliable on the thousandths, and the accuracy will not be influenced by the scale of the question (i.e. the number of tanks in this study).

D DAMS Case study #2 setting

The layout of the tank farm considered for case study #2 is reported in Figure 7(b). The vessels are atmospheric tanks containing flammable liquids. The features of the tanks, the stored substances and the considered inventories are summarized in Table D.1. The meteorological conditions assumed for case study are T = 15°C, relative humidity = 25%, limited wind speed, leading to negligible flame tilting effects. The consequence assessment of the pool fires following the ignition of the total inventory of tank T17 after a major leakage are analysed applying the software ALOHA. The heat radiation received by each of the other vessels is reported in Table D.2 at the correspondent ID.

Table D.1 Main features of the vessels considered in case study 2.

ID	Diameter (m)	Height (m)	Capacity (m^3)	Filling level (%)	Substance	Density (kg/m^3)	Inventory (ton)
T1	24	12.6	5710	0.5	Crude oil	950	2712
T2	30	10.8	7630	0.7	Crude oil	950	5074
T3	30	10.8	7630	0.7	Crude oil	950	5074
T4	24	12.6	5710	0.49	Crude oil	950	2658
T5	15	12.6	2225	0.44	Gasoline	750	734
T6	27	9	5182	0.26	Crude oil	950	1280
T7	21	12.6	4350	0.8	Gasoline	750	2610
T8	15	12.6	2225	0.27	Gasoline	750	451
T9	24	12.6	5710	0.58	Gasoline	750	2484
T10	18	7.2	1843	0.39	Gasoline	750	539
T11	18	7.2	1843	0.59	Gasoline	750	815
T12	12	7.2	806	0.23	Gasoline	750	139
T13	7.5	9	397	0.76	Benzene	876	265
T14	7.5	9	397	0.14	Benzene	876	49
T15	15	12.6	2225	0.62	Toluene	867	1196
T16	27	9	5129	0.46	Methanol	792	1869
T17	21	12.6	4350	0.74	Ethanol	789	2540
T18	21	12.6	4350	0.36	Ethanol	789	1236
T19	15	12.6	2225	0.32	Methanol	792	564
T20	21	12.6	4350	0.42	Solvent	650	1188
T21	13.5	9	1282	0.28	Solvent	650	233
T22	12	7.2	806	0.44	Solvent	650	231
T23	15	18	3179	0.46	Solvent	650	951
T24	18	7.2	1843	0.72	Ethanol	789	1047
T25	21	12.6	4350	0.34	Ethanol	789	1167
T26	15	14.4	2543	0.45	Benzene	876	1003
T27	24	12.6	5710	0.77	Benzene	876	3852
T28	18	7.2	1843	0.8	Gasoline	750	1106
T29	18	7.2	1843	0.43	Toluene	867	687

T30	18	7.2	1843	0.63	Methanol	792	919
T31	12	7.2	806	0.23	Solvent	650	121
T32	15	9	1590	0.42	Solvent	650	434
T33	15	9	1590	0.54	Solvent	650	558
T34	21	12.6	4350	0.68	Gasoline	750	2219

Table D.2 Results of the consequence assessment of the pool fires associated to each tank. Radiation received by each target is reported in kW/m^2. Values lower than 10 kW/m^2 are excluded (Na).

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25	T26	T27	T28	T29	T30	T31	T32	T33	T34
T1	-	23.7	Na	10.9	Na	Na	Na	Na	Na	Na	15.4	15.2	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na
T2	30.1	-	55.6	43.3	24.8	12.2	Na	12.9	Na	11.4	13.8	17.9	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na
T3	12.3	55.6	-	32.2	44.2	11.4	Na	17.5	Na	Na	Na	10.3	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na
T4	11.1	33.5	25.5	-	35.4	29.5	13.8	24.9	Na	17.0	14.0	23.4	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na
T5	Na	13.2	21.2	34.2	-	22.4	23.6	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	12.8	Na	Na	Na	Na	Na	Na
T6	Na	Na	Na	26.1	14.1	-	35.6	27.0	Na	25.2	11.5	17.3	Na	10.3	10.3	14.1	16.3	Na	Na	Na	Na	Na	Na	31.3	16.8	27.0	17.8	18.9	Na	Na	Na	Na	Na	Na
T7	Na	10.2	12.3	23.8	25.2	57.1	-	63.2	Na	15.7	Na	12.5	Na	Na	Na	12.2	16.0	Na	Na	Na	Na	12.7	11.0	36.5	30.8	52.8	57.1	91.7	20.4	26.5	12.0	17.1	12.0	Na
T8	Na	13.2	21.2	34.2	70.0	22.4	23.6	-	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	12.8	Na	Na	Na	Na	Na	Na
T9	Na	Na	Na	Na	Na	Na	Na	Na	-	21.4	22.9	14.7	37.6	36.5	51.1	36.5	18.3	31.7	18.7	16.4	Na	11.3	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na
T10	Na	Na	Na	13.3	Na	24.7	Na	Na	Na	-	43.5	51.0	28.3	36.2	29.7	25.9	13.8	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na
T11	10.6	Na	Na	Na	Na	Na	Na	Na	Na	40.2	-	58.8	42.5	31.2	18.3	11.4	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na
T12	Na	Na	Na	10.6	Na	Na	Na	Na	Na	33.4	42.5	-	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na
T13	Na	Na	Na	Na	Na	11.4	Na	Na	25.5	55.2	58.7	26.5	-	99.0	91.8	36.8	14.1	14.6	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na
T14	Na	Na	Na	Na	Na	11.4	Na	Na	25.5	55.2	58.7	26.5	99.0	-	91.8	36.8	14.1	14.6	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na
T15	Na	Na	Na	Na	Na	10.1	Na	Na	10.8	22.9	10.1	Na	15.9	23.9	-	73.8	73.8	26.3	13.8	Na	Na	10.5	Na	10.1	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na
T16	Na	Na	Na	Na	Na	10.1	Na	Na	10.8	22.9	10.1	Na	15.9	23.9	73.8	-	73.8	26.3	13.8	Na	Na	10.5	Na	10.1	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na
T17	Na	Na	Na	Na	Na	10.1	Na	Na	10.8	22.9	10.1	Na	15.9	23.9	73.8	73.8	-	26.3	13.8	Na	Na	10.5	Na	10.1	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na
T18	Na	Na	Na	Na	Na	Na	Na	Na	15.5	Na	Na	Na	Na	12.0	24.6	44.0	29.8	-	46.2	29.1	17.2	19.9	11.1	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na
T19	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	18.9	-	26.0	15.0	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na
T20	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	20.6	23.0	54.5	78.2	-	42.3	28.9	22.2	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na
T21	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	10.9	19.7	15.0	24.8	25.8	-	57.8	77.1	12.8	17.2	Na	Na	Na	16.6	Na	12.4	Na	Na	Na
T22	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	21.9	56.2	23.8	39.6	24.9	65.0	-	60.2	23.8	26.0	12.0	Na	Na	16.0	Na	Na	Na	Na	Na
T23	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	10.9	19.7	15.0	24.8	25.8	77.1	57.8	-	12.8	17.2	Na	Na	Na	16.6	Na	12.4	Na	Na	Na
T24	Na	Na	Na	Na	Na	16.1	24.8	30.3	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	-	36.5	36.5	36.5	36.5	26.7	31.2	10.8	17.3	Na	Na
T25	Na	Na	Na	Na	Na	16.1	24.8	30.3	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	36.5	-	36.5	36.5	36.5	26.7	31.2	10.8	17.3	Na	Na
T26	Na	Na	Na	12.1	10.3	33.2	49.3	59.7	Na	13.3	Na	Na	Na	Na	Na	15.2	23.3	11.9	12.8	10.5	16.2	22.2	22.2	74.3	74.3	-	74.3	74.3	53.0	61.5	25.6	35.3	23.3	16.8
T27	Na	Na	Na	12.1	10.3	33.2	49.3	59.7	Na	13.3	Na	Na	Na	Na	Na	15.2	23.3	11.9	12.8	10.5	16.2	22.2	22.2	74.3	74.3	74.3	-	74.3	53.0	61.5	25.6	35.3	23.3	16.8
T28	Na	Na	Na	10.8	Na	33.2	49.7	60.6	Na	12.0	Na	Na	Na	Na	Na	14.1	22.7	10.6	11.5	Na	15.1	21.6	21.6	75.5	75.5	75.5	75.5	-	53.5	62.4	25.1	35.3	22.7	15.8
T29	Na	Na	Na	Na	Na	12.5	15.5	Na	Na	Na	Na	Na	Na	Na	Na	12.4	15.8	10.5	12.9	12.5	21.1	21.1	31.0	22.6	39.6	25.0	35.7	20.7	-	79.8	79.8	79.8	79.8	79.8
T30	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	Na	11.4	Na	15.5	Na	13.7	Na	29.5	-	29.5	29.5	29.5	29.5
T31	Na	Na	Na	Na	Na	12.8	16.0	Na	Na	Na	Na	Na	Na	Na	Na	12.6	16.4	10.6	13.3	12.8	22.3	22.3	33.1	23.9	42.3	26.5	38.2	21.7	85.2	85.2	-	85.2	85.2	85.2
T32	Na	Na	Na	Na	Na	12.8	16.0	Na	Na	Na	Na	Na	Na	Na	Na	12.6	16.4	10.6	13.3	12.8	22.3	22.3	33.1	23.9	42.3	26.5	38.2	21.7	85.2	85.2	85.2	-	85.2	85.2
T33	Na	Na	Na	Na	Na	12.8	16.0	Na	Na	Na	Na	Na	Na	Na	Na	12.6	16.4	10.6	13.3	12.8	22.3	22.3	33.1	23.9	42.3	26.5	38.2	21.7	85.2	85.2	85.2	85.2	-	85.2
T34	Na	Na	Na	Na	Na	10.5	13.4	Na	Na	Na	Na	Na	Na	Na	Na	10.3	13.7	Na	10.9	10.5	19.2	19.2	29.3	20.8	37.8	23.2	34.0	18.7	75.6	75.6	75.6	75.6	75.6	-

CURRICULUM VITAE

Laobing Zhang

Basic Information

Name: Laobing Zhang (Chinese: 张烙兵)

M/F: Male

Date of Birth: 1989/08/20

Nationality: P.R. China

E-mail: laobing.zhang@tudelft.nl

Or laobingzhang.nudt@gmail.com

Research Interests

- ➢ Game Theory
- ➢ Security
- ➢ Critical Infrastructure Protection
- ➢ System Simulation

Education

● PhD Candidate (2014.10 – 2018.12)

- Major: Game theory for managing security in chemical industrial areas
- Delft University of Technology, Delft, The Netherlands
- Supervisor: Prof. dr. ir. Genserik Reniers

● Master of Science (2012.09 - 2014.09)

- Major: meta modelling for large scale artificial society simulation
- National University of Defense Technology, Changsha, P.R.China
- Supervisor: Prof. dr. Xiaogang Qiu

● Bachelor of Science (2008.09 - 2012.06)

- Major: Simulation Engineering
- National University of Defense Technology, Changsha, P.R.China

List of Publications

Journal Publications (# indicates that I am the communication author)

1. **Zhang, L.,** & Reniers, G. (2016). A Game-Theoretical Model to Improve Process Plant Protection from Terrorist Attacks. *Risk analysis*, *36*(12), 2285-2297.
2. **Zhang, L.,** Landucci, G., Reniers, G., Khakzad, N., & Zhou, J. (2017). DAMS: A Model to Assess Domino Effects by Using Agent-Based Modeling and Simulation. *Risk analysis*.
3. **Zhang, L.,** & Reniers, G. (2018). Applying a Bayesian Stackelberg game for securing a chemical plant. *Journal of Loss Prevention in the Process Industries*, *51*, 72-83.
4. **Zhang, L.,** Reniers, G., & Qiu, X. (2017). Playing chemical plant protection game with distribution-free uncertainties. *Reliability Engineering & System Safety*.
5. **Zhang, L.,** Reniers, G., Chen, B., & Qiu, X. (2018). Integrating the API SRA methodology and game theory for improving chemical plant protection. *Journal of Loss Prevention in the Process Industries*, *51*, 8-16.
6. **Zhang, L.,** Reniers, G., Chen, B., & Qiu, X. (2018). CCP game: A game theoretical model for improving the schedulling of chemical cluster patrolling. *Reliability Engineering & System Safety*.
7. **Zhang, L.,** Reniers, G., Chen, B., & Qiu, X. (2018). A Chemical Plant Protection Game Incorporating Bounded Rational Attackers and Distribution-free Uncertainties, *Submitted to Risk Analysis.*
8. **Zhang, L.,** & Reniers, G. (2018). Applying game theory for improving security in the process industries. *Journal of Integrated Security Science*, 2(1), 19-24.
9. **Zhang, L.**, Landucci, G., Reniers, G. L. L. M. E., Ovidi, F., Khakzad, N., & Zhou, J. (2018). Applying Agent Based Modelling and Simulation for Domino Effect Assessment in the Chemical Industries. *Chemical Engineering Transactions*.
10. Rezazadeh, A., **Zhang, L.,** Reniers, G., Khakzad, N., & Cozzani, V. (2017). Optimal patrol scheduling of hazardous pipelines using game theory. *Process Safety and Environmental Protection*, *109*, 242-256.
11. $^{\#}$ Rezazadeh, A., Talarico, L., Reniers, G., Cozzani, V., & **Zhang, L.** (2018). Applying game theory for securing oil and gas pipelines against terrorism. *Reliability Engineering & System Safety*.

12. Zhou, J., Reniers, G., & **Zhang, L.** (2017). A weighted fuzzy Petri-net based approach for security risk assessment in the chemical industry. *Chemical Engineering Science*, *174*, 136-145.
13. Zhu, Z., Chen, B., Reniers, G., **Zhang, L.,** Qiu, S., & Qiu, X. (2017). Playing chemical plant environmental protection games with historical monitoring data. *International journal of environmental research and public health*, *14*(10), 1155.
14. Wang, B., Wu, C., Huang, L., **Zhang, L.,** Kang, L., & Gao, K. (2018). Prevention and control of major accidents (MAs) and particularly serious accidents (PSAs) in the industrial domain in China: Current status, recent efforts and future prospects. *Process Safety and Environmental Protection*.
15. Wang, B., Wu, C., Reniers, G., Huang, L., Kang, L., & **Zhang, L.** (2018). The future of hazardous chemical safety in China: Opportunities, problems, challenges and tasks. *Science of The Total Environment*, *643*, 1-11.
16. Chen, C., Reniers, G., & **Zhang, L.** (2018). An innovative methodology for quickly modeling the spatial-temporal evolution of domino accidents triggered by fire. *Journal of Loss Prevention in the Process Industries*, *54*, 312-324.

Book Chapter

1. **Zhang, L.,** & Reniers, G. (2018). Applying Game Theory for adversarial risk analysis in chemical plants. In *Security risk assessment in the chemical and process industry/Reniers, Genserik [edit.]; et al.* (pp. 110-130).

Monograph

1. **Zhang, L.,** & Reniers, G. (2018). *Game Theory for Managing Security in Chemical Industrial Areas*. Springer.

Conference Paper

1. **Zhang, L.,** & Reniers, G. L. L. (2018). Optimizing security patrolling scheduling in chemical industrial parks by using game theory. In *Safety and reliability: safe societies in a changing world: proceedings of ESREL 2018, June 17-21, 2018, Trondheim, Norway/Haugen, Stein [edit.]* (pp. 3001-3006).

Conference Presentations

1. ARA4CI@Leiden
2. GDRR2017@Madrid
3. ESREL2018@Trondheim
4. CISAP2018@Milano
5. International Symposium of the MKO Process Safety Center@Taxis

Acknowledgement

Life can be long but only several steps are critical. Initiating the idea of studying abroad, passing the IELTS test, obtaining the offer letter from TUDelft, getting the approval from China Scholarship Council, arriving Schipol airport, first meeting Genserik at the corridor, and many other things, are unforgettable moments in my life. I am so thankful to have these people and these moments in my life.

Genserik Reniers, a person I did not know until 2013 however afterwards who has been playing and will continuously play a vital role in my life. I can still remember all the details about our first skype meeting in the winter of 2013 at 10 pm (Beijing time, therefore 3 pm CET). The moment in October 2014 that you stood by my office and asked “are you Laobing?” is like just happened yesterday. Super-efficient, industrious, patient, open-minded, handsome, and knowledgeable, are those words suitable to describe you. Thank you for allowing me to knock your door at any time when I have questions, for always replying any of my requests within a short time, for returning my writings (proposals/papers/books) with enormous revisions and still saying “good job”, for introducing me experts related to my research topic etc. I would not only learn from your expertise but also learn how you manage time and deal with complex tasks, how you stay with people, how you look at things and so on.

Never forget where you started. Special thanks should be addressed to Prof. Xiaogang Qiu, Dr. Bin Chen, Dr. Gang Guo, and Dr. Hong Duan, who were the supervisor team of my master degree and introduced me to the academic word. Thanks to Xiaogang for organizing the team of large-scale social simulation, from where I learned how to conduct scientific research and how to cooperate with colleagues to solve scientific as well as engineering problems. Without Xiaogang’s support, I would never have the opportunity to study abroad. Thanks to Bin, Gang, and Hong for all those interesting discussions on meta-modeling, on building a computer simulation engine, and on computer simulation in general. The 15th floor of the Tech Building in NUDT will be a place that I will never forget.

Prof. Wout Dullaert, Prof. Pieter van Gelder, Prof. Bartel Van de Walle, Prof. Stefan Pickl, Prof. Fubao Zhou, and Prof. David Rios Insua, thank you for agreeing to be part of my defence committee. Your insightful comments and expertise improve the quality of the thesis. Pieter, thank you also for your great management works for the safety science group as well as for your technical support on my mathematical questions. Special thanks also to Prof. Alexander Verbraeck, thank you for your great help in my research, from my master degree period to my PhD.

Colleagues from the safety science group have always been kind and helpful. Peter, a retired pilot, the grandfather of several kids, a researcher who studies the foundation of safety science, thank you for your help during these years. Paul, thank you for introducing us the history of safety science as well as of the Netherlands. I enjoyed a lot from the talks with you. Pei-hui and Zarah, thank you for telling me how to deal with some daily issues of living in the Netherlands. Thanks to Gabriele, Nima, and Wolter, for your help on my research. Yuling, Yamin, Pengfei, Chao, Xin, Jie, Jianfeng, Kunru, Jerry, Yunxiao, Xuchao, Yaping, Saba, Francesca, Valeria, Federica, Amirali, Behnaz, Zahra, and Jos, I am so lucky to have all of you in my life. Monique and Astrid, thank you for your great help during these years.

Friends make life worth living. Xu, a friend as well as a supervisor of mine, thank you for all your guidance and for the discussions we had. Jiapeng, Yuan, Yamin, and Linying, thank you for your help during the moment that I needed the most in these four years. Mingxin, Zhangling, Yun, Baohong, Yuewen, and many others, thanks to your friendship. Zhen, thank you for all your help on the administrative works in NUDT.

Great thanks to China Scholarship Council (CSC), who funded my PhD study. CSC enables me to enter a new world. I would also like to thank the LDE Centre for Safety and Security for their financial support during my extension in the last period of my PhD. Jos, Werner, Kathy, Dick, Alexander, and Paul, thank you for participating in the interviews for evaluating my models. Thank you for your kind hospitality when I visited your sites.

The family is always the power source. Thanks to my parents and parents-in-law for unconditionally supporting my PhD research. Xingxing, my lovely wife, is the most beautiful girl in the world and is becoming a strong mother. Thank you for accompanying me during the last twelve years, witnessing my growth from a high-school boy to a man with a PhD degree. Xinhe, my little boy, thanks for your cries in the midnight, reminding me to read more papers and be more productive. Your coming makes me re-think about some fundamental philosophical questions, such as the balance of work and live, the meaning of life, and the world peace etc.

Laobing Zhang

Delft, August 2018