Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea Magistrale in Matematica

A framework for risk analysis in automotive cybersecurity

Tesi di Laurea Magistrale

Relatore:
Chiar.mo Prof.
Stefano Pagliarani

Correlatore:
Dott.ssa
Anastasia Cornelio

Presentata da:
Alessandro Sforza

IV Sessione
Anno Accademico 2020/2021

Introduzione

Questa trattazione affronta il problema della cybersecurity nel settore automotive dal punto di vista dell'analisi delle minacce e la valutazione del rischio (TARA, dall'inglese 'Threat Analysis and Risk Assessment'). La questione centrale che motiva la tesi è quella dell'accettabilità dei rischi, fondamentale per prendere una decisione sulle soluzioni di sicurezza da implementare. A tal fine, sviluppiamo un framework quantitativo nel quale prendiamo in input i risultati della valutazione del rischio e definiamo delle misure di diversi aspetti di una possibile risposta al rischio; sfruttiamo quindi la naturale presenza di trade-off (costo contro efficacia) per formulare il problema come un'ottimizzazione multi-obiettivo. Infine, sviluppiamo un modello stocastico dell'evoluzione futura dei fattori di rischio mediante il potente strumento di modellazione rappresentato dalle catene di Markov; adattiamo le formulazioni dei problemi di ottimizzazione a questo contesto non deterministico.
La tesi è il frutto di una collaborazione con la Vehicle Electrification division di Marelli (in particolare con il Cybersecurity Team di Bologna) e ciò ha permesso durante l'intero lavoro di considerare una particolare istanza del problema, derivante da una vera TARA, in modo da testare sia il framework deterministico che quello stocastico in un'applicazione del mondo reale. La collaborazione spiega anche il motivo per cui spesso nella tesi si assume il punto di vista di un tier-1 supplier; tuttavia, le analisi svolte si possono adattare ad un qualsiasi altro livello della supply chain.

Nel Capitolo 1, introduciamo brevemente le motivazioni alla base del problema della cybersecurity in ambito automotive ed esempi di possibili

attacchi e minacce per gli utenti della strada. Presentiamo alcune delle norme e normative emerse negli ultimi anni in questo campo con particolare riguardo alla ISO/SAE 21434 [7], che è la più recente e rappresenta una base importante per tutta la tesi. Chiudiamo con un elenco e una possibile classificazione allo stato attuale delle soluzioni di cybersecurity.

Nel capitolo 2, riassumiamo due tecniche di valutazione del rischio, contenute in EVITA [8] e ISO/SAE 21434: mostriamo somiglianze e differenze e, soprattutto, stabiliamo una serie di definizioni e termini che vengono utilizzati in tutto tutto il lavoro. Si passa quindi alle idee principali della tesi: il punto di vista dell'analisi costo-efficacia, la formulazione matematica del problema, la definizione della riduzione del rischio, della soddisfazione del cliente e delle funzioni obiettivo che si vogliono ottimizzare; finalmente applichiamo queste idee nella suddetta applicazione del mondo reale, dove siamo in grado di validare alcune delle intuizioni che il decisore già aveva.

Nel capitolo 3, riconosciamo che i rischi sono soggetti al cambiamento e cerchiamo di modellare la loro evoluzione nel tempo con l'aiuto delle catene di Markov. Le quantità introdotte nel Capitolo 2 diventano quindi processi stocastici di cui siamo in grado di calcolare i valori attesi mediante simulazione delle catene di Markov. Dobbiamo tener conto della stocasticità; in particolare, modelliamo l'avversione del decisore verso l'incertezza mediante funzioni di utilità e analisi di eventi estremi; ciò porta a nuove formulazioni del problema di ottimizzazione. Alla fine, torniamo all'applicazione del mondo reale e vediamo che tipo di conclusioni possiamo trarre dai risultati ottenuti.

Introduction

This work addresses the problem of automotive cybersecurity from the point of view of Threat Analysis and Risk Assessment (TARA). The central question that motivates the thesis is the one about the acceptability of risk, which is vital in taking a decision about the implementation of cybersecurity solutions. For this purpose, we develop a quantitative framework in which we take in input the results of risk assessment and define measures of various facets of a possible risk response; we then exploit the natural presence of trade-offs (cost versus effectiveness) to formulate the problem as a multi-objective optimization. Finally, we develop a stochastic model of the future evolution of the risk factors, by means of Markov chains; we adapt the formulations of the optimization problems to this non-deterministic context.
The thesis is the result of a collaboration with the Vehicle Electrification division of Marelli, in particular with the Cybersecurity team based in Bologna; this allowed us to consider a particular instance of the problem, deriving from a real TARA, in order to test both the deterministic and the stochastic framework in a real world application. The collaboration also explains why in the work we often assume the point of view of a tier-1 supplier; however, the analyses performed can be adapted to any other level of the supply chain.

In Chapter 1, we briefly introduce the motivations behind the cybersecurity problem in the automotive field and examples of possible attacks and threats for road users. We present then some of the standards and regulations that emerged in the last few years in this field with particular regard of ISO/SAE 21434 [7], which is the most recent and represents an important

foundation for all the thesis. We close with a list and a possible classification of cybersecurity solutions to the present day.

In Chapter 2, we summarize two risk assessment techniques, contained in EVITA [8] and ISO/SAE 21434: we show similiraties and differences and most importantly establish a series of definitions and terms which are used throughout all the work. We then move on to the main ideas of this work: the cost-effectiveness point of view, the mathematical formulation of the problem, the definition of risk reduction and customer satisfaction and the objective functions we want to optimize; we finally apply these ideas in the aforementioned real world application, where we are able to validate some of the insights that decision maker already had.

In Chapter 3, we acknowledge that risks are subject to change and try to model their evolution with the aid of Markov chains. The quantities introduced in Chapter 2 then become stochastic processes whose expected values we are able to compute by simulation of the Markov chains. We need to take into account stochasticity; in particular, we enforce the decision maker aversion towards uncertainty by means of utility functions and analysis of extreme events; this leads to new formulations of the optimization problem. In the end, we return to the real world application and see what kind of conclusions we can draw from the results obtained.

Contents

List of Figures

List of Tables

Chapter 1

Introduction to security in the automotive industry

1.1 Security in modern vehicles

A modern vehicle is equipped with approximately 50/80 independent computers, called Electronic Control Units (ECUs).
Electronics has been a key part of the functioning of vehicles for a while and it has increasingly controlled many useful operations. For example, the physical inputs, as those triggered when the driver pushes on the brake pedal, are not mechanically transformed to their result, rather some ECUs mediate this operation. ECUs are also used to enforce helpful safety relevant features: for example Advanced Driver-Assistance Systems (ADAS), which are groups of electronic technologies that assist drivers in driving and parking functions. Other than safety, the presence of ECUs on a vehicle allows for more 'hedonistic' features, for example the infotainment system that can be found nowadays in many cars.

ECUs are highly interconnected: they receive inputs from sensors, exchange data with actuators and communicate over one or more internal network buses, namely Local Interconnect Network (LIN) bus, Controller Area Network (CAN) bus, FlexRay and only recently Ethernet.

For many years, these networks were isolated from the outside. Over the last years this situation suddenly changed: modern vehicles offer broad attack surfaces and access points, both wired and wireless (for example Wi-Fi, Bluetooth, cellular). Furthermore, there is always the possibility that the user modifies the conditions of the vehicle after it has been produced. The so-called *automotive aftermarket* is the secondary market of the automotive industry, concerned with the manufacturing, remanufacturing, distribution, retailing, and installation of all vehicle parts, chemicals, equipment, and accessories, after the sale. The parts, accessories, etc. for sale may not come from the OEM (Original Equipment Manufacturer). Important, and also potentially dangerous, examples are Bluetooth or WiFi OBD adapter and scan tools, but in general the variety of parts for nearly all vehicle makes and models is almost unlimited.

The automotive industry was not really prepared: the architecture of these systems was never designed having in mind the possibility of connection to the outside. On the CAN-bus, for example, ECUs send CAN packets, broadcast to all components on the bus; each component decides whether a given message is intended for itself or not. Furthermore, the CAN-bus, unlike Ethernet, is not meant to be equipped with any authentication protocol. This means that a malicious attacker who finds a way to breach and gains control of just a single ECU can then control many other functions and components across the vehicle. Along with architecture limits, a general lack of cybersecurity culture caused the emergence of many vulnerabilities. As reported in [2], most of the components of the network may become target of attacks: in-vehicle devices, ECUs, sensors and actuators, safety critical and non-safety critical applications running on in-vehicle devices, communication links between all these components.

The emergence of these kind of threats has sparked the introduction of cybersecurity in the automotive field. Cybersecurity will be even more vital if we consider the direction of technological progress: according to a report from Juniper Research published in 2018, connected cars (via telematics or

by in-vehicle apps) are expected to increase to 775 million by 2023, rising from 330 million vehicles in 2018. As we are inevitably going towards more and more connections between vehicles, connections with the environment outside vehicles, all operating in the context of intelligent transportation systems and smart cities, cybersecurity is a more and more stringent need.

We now give a brief perspective on some of the groundbreaking events in the young history of automotive hacking. Our main source is [3], which appears on the website of Miller and Valasek (the authors of the notorious 2015 Jeep Cherokee hack described below). Among the earliest research in this field, in 2010 a group of security researchers demonstrated that they were able to gain control of an entire car system under the condition of having prior physical access to the ECU [4]; for example, they were able to control the display on the speedometer, kill the engine, affect braking functions. The article was widely criticized because the threat model with prior physical access was considered unrealistic.
In a follow-up paper [5], they showed how physical access is not necessary and documented the possibilities of both short and long range wireless access. They were able to get code execution on the vehicle through Bluetooth or the telematics unit and consequently to inject CAN messages and compromise functionalities as in the previous article.
In 2015, Miller and Valasek [3] were able compromise the head unit of a Jeep Cherokee model by exploiting a vulnerability accessible through the Internet. They were able to get code running on the head unit and to reprogram the firmware of another processor of that unit. At this point, in the same way as above, they were able to inject CAN messages. The interesting fact is that no user interaction was required and the attack could be extended, much like a common computer virus, to all the fleet of Jeep Cherokee in the US; FCA had to issue a safety recall to update the software in almost 1.4 million vulnerable vehicles, with a huge cost for the company and bother for customers.
Other attacks have been documented in the last years and they all show a

similar chain of exploits as the hack recounted above. The first step in the chain is the remote attack, which can differ on the distance, the need of any user interaction and required equipment. After executing code on some internal component, the second step is to send messages to the vehicle's critical ECUs and this requires additional work: the components connected to the outside are often isolated from the safety critical ones and the attacker has to reprogram a device that acts as a gateway. In the final step, the attacker send CAN messages to control operative functionalities of the vehicle.
In 2015/2016, attacks of this nature conducted on the Tesla Model S resulted in the first proactive mass Over The Air (OTA) security update of vulnerable vehicles; the ability to safely perform this kind of software update is considered one of the key challenges in the future of automotive cybersecurity.

1.2 New regulations and activities

The introduction of new technologies and functionalities in modern vehicles has determined an increasing need in security, as for example safety and privacy rise when the right security measures are applied. New regulations have appeared in order to ensure cybersecurity in this field.
The most recent example of this kind of regulation is UN R155 [6], which came into force in January 2021 and whose requirements must be fulfilled for the homologation of any vehicle produced starting from July 2024. The recipients of the regulation are carmakers (who are responsible for the homologation of vehicles), however the regulation indirectly affects suppliers because it clearly states that carmakers need to deal with risks related to the suppliers. The requirements in UN R155 are commonly overturned to suppliers in the form of specifications about processes that involve product development and post-development. In this sense, cybersecurity risk management involves the whole automotive supply chain.
OEMs must establish and certify a Cybersecurity Management System (CSMS), which ensures security is adequately considered during development, produc-

tion and post production phases. For each vehicle type, then, OEMs must prove that they are able to manage vehicle cyber risks including supplier related risks, secure vehicles by design to mitigate risks along the value chain, detect and respond to security incidents across the whole vehicle fleet.

Risk management comprises risk assessment, which consists in identifying the threats and vulnerabilities to which the vehicle is subject, and its treatment through the identification of appropriate mitigations. One of the most recent contribution in this regard is ISO/SAE 21434 standard. The reader is referenced to Section 2.2 for details. OEMs typically require suppliers to implement ISO/SAE 21434 in order to demonstrate compliance with UN R155 requirement about the management of supplier-related risk.
The cybersecurity risk management of an organization described in ISO/SAE 21434 applies throughout all lifecycle phases as illustrated in Figure 1.1.

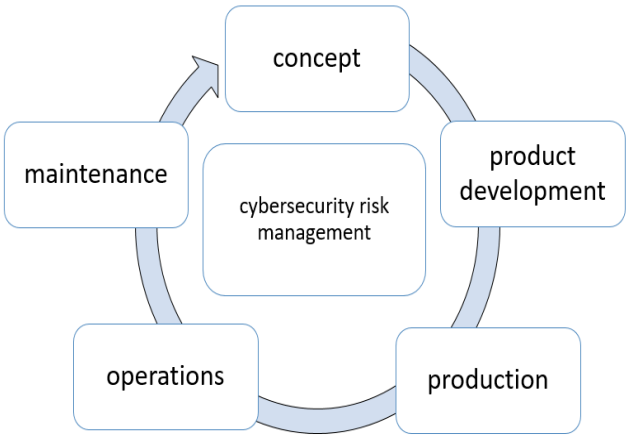


Figure 1.1: Overall cybersecurity risk management in [7]

In particular, in the product development phase, cybersecurity activities are performed iteratively until no further refinements of cybersecurity controls are needed. The cybersecurity specifications are defined and confirmed through verification activities for the fulfilment of the cybersecurity concept.

Figure 1.2 illustrates an example of how a V-model-based workflow can be iteratively applied at three levels, i. e. item, component and sub-component

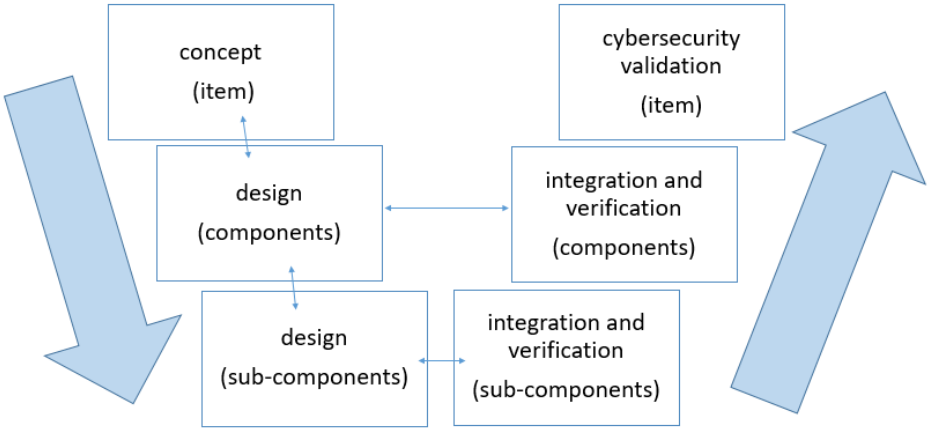


Figure 1.2: Example of product development V-model in [7]

level. This means that, at each level, the entire V-cycle must be applied, starting with the requirements analysis and architectural design of the left-hand side and going on with the integration and verification of the right-hand side. Horizontal bi-directional arrows depict verifications of the implemented and integrated component against its cybersecurity specifications while vertical bi-directional arrows depict verifications against the cybersecurity specification from a higher level of architectural abstraction during design.

1.3 Examples of cybersecurity solutions

In this section, we briefly report and categorize some automotive cybersecurity solutions that can be implemented in vehicles. We present them at a high level of abstraction, knowing that during product development the solutions are refined into detailed technical requirements. These solutions appear at three levels: vehicle, ECU and infrastructure level.

At vehicle level, the introduction of the central gateway ECU is the fundamental architectural evolution: this ECU separates trusted domain from untrusted and is equipped to be attack resistant, with dedicated hardware (HSMs, Hardware Security Modules), cryptographic functions and firewall

policies.
At ECU level, specific on-board functions have been adopted, both hardware and software, to enforce secure communication, integrity of ECU software and data, access control on the network. We give a possible classification:

- **access regulation**, functions regulating accesses from external entities and their associated roles (Secure Diagnostics granted through authentication of the diagnostic tool (via OEM server), Secure Debug Access through password securely stored;
- **intrusion detection**, functions inspecting communications in order to detect anomalies or intrusions (firewall to filter messages according to whitelist/blacklist or rule-based techniques and Intrusion Detection Systems);
- **logging**, functions providing secure log of sensitive data preserving confidentiality and authenticity;
- **network security**, protocols providing reliable communications (Secure On-Board Communication) in order to prevent from spoofing on CAN network;
- **software authentication**, functions that ascertain reliable and authentic software running on the ECU, during updates (Authenticated Software Update) and even at each boot via hash functions for example (Authenticated Boot);
- **services**, like memory protection or certificate and key managers.

Finally, at infrastructure level, we have some off-board measures that permit the functioning of on-board solutions:

- **Public Key Infrastructure (PKI)** consists of hardware, software, policies and standards that manage creation, administration, distribution and revocation of digital certificates and private keys used for ECUs communications;

- **Firmware Over The Air (FOTA) infrastructure** manages software packages installations and updates, which are dangerous operations and must be carefully performed;
- **security on cloud infrastructure** absorbs the task of collecting all the data coming from vehicles, concerning cybersecurity events in order to analyze them with statistical and machine learning strategies to detect intrusions, incidents, attacks.

Chapter 2

Risk assessment and treatment

A valid framework for risk assessment in automotive is contained in the EVITA project. Here we report a brief summary of the main aspects, the reader is referred to [8] for details. There are other methodologies (in [2] it is possible to find examples, analyses and comparisons) but in terms of style and contents the techniques used in available examples closely resemble EVITA. EVITA is an old project (dating back to 2009) which is freely available online and we will introduce for its undoubted historical meaning. ISO/SAE 21434 is a more recent standard (see [7]); our choice is to introduce EVITA, which shares important traits with ISO/SAE 21434 in terms of risk analysis and highlight some of the differences in Section 2.2. ISO/SAE 21434 will then be the reference model in all the examples.

2.1 Risk assessment: EVITA framework

In EVITA, threat identification and modeling is conducted using the approach of *attack trees*, which are related to fault trees commonly used for safety hazards. The root of an attack tree (Level 0) is an abstract *attack goal* that gives the attacker a benefit of some kind. Its child nodes (Level 1) represent different *attack objectives* that could satisfy this attack goal. The attack objectives may be further decomposed into a number of *attack*

methods that could be employed to achieve the attack objective. Each attack method will in turn be based on a logical combination (AND/OR) of attacks against one or more *assets* populating the lowest levels of the attack tree. These are described here as *asset attacks*, and are the terminal nodes of the tree.

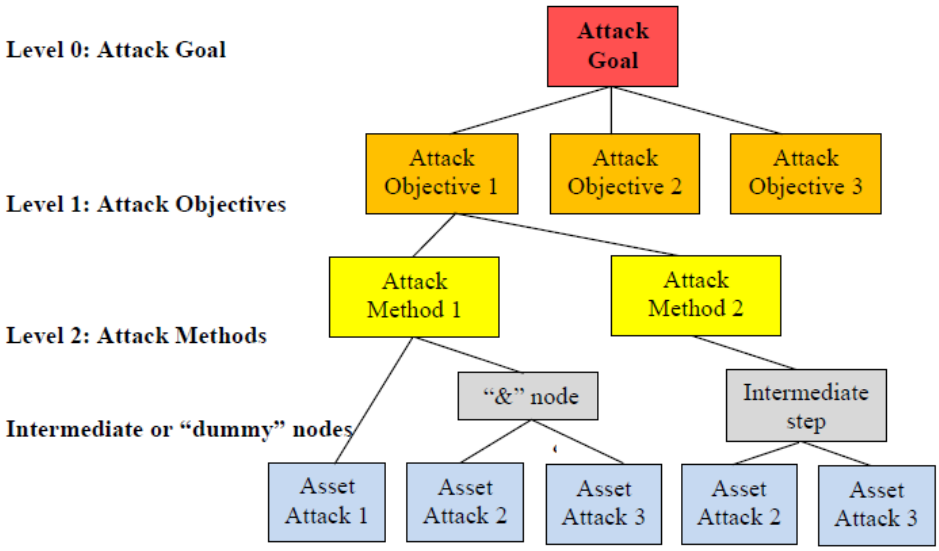


Figure 2.1: General attack tree structure in EVITA

Having established a threat model, EVITA defines the risk of an attack as a function of the possible severity of the attack for the stakeholders and the estimated probability of occurrence of a successful attack.

Severity is estimated at Level 1 (attack objective) of the attack tree and is a vector of four components (see [8, Table 4]):

- **safety**: physical injuries that might be sustained by persons;
- **privacy**: identification and tracking of vehicles or individuals;
- **financial**: financial losses that may be experienced by individuals or ITS operators;
- **operational**: interference with vehicle systems and functions that do not impact on functional safety.

Likelihood of occurrence is estimated at terminal nodes of the attack tree and then combined following the logic (AND/OR) of the attack tree. It is defined through the estimation of attack potential, a measure of the minimum effort to be expended in an attack to be successful. The following factors are considered:

- **Elapsed Time**: total amount of time taken by an attacker to identify that a particular potential vulnerability may exist, to develop an attack method and to sustain effort required mounting the attack.

- **Specialist Expertise**: required level of general knowledge of the underlying principles, product types or attack methods.

- **Knowledge of the system under investigation**: specific expertise in relation to the system under investigation. Though it is related to general expertise, it is distinct from that.

- **Window of opportunity**: amounts of access to a system required to identify and exploit vulnerabilities, it may increase the likelihood of detection of the attack.

- **IT hardware/software or other equipment**: equipment required to identify and exploit vulnerability.

For each asset attack all these factor are given a value according to a scale (see [8, Table 5]) and then summed to get attack potential, which is transformed into attack probability (integer on a scale from 1 to 5). If an attack method can be implemented using any one of a number of asset attacks (OR relationship) the combined attack probability is taken to be the highest of the attack probabilities. Where the attack method requires a conjunction of asset attacks (AND relationship), the combined attack probability is taken to be the lowest of the attack probabilities associated with the contributing asset attacks.

The risk level (a vector of four integer components, one for every severity aspect) is determined from the severity associated with the attack objective and the combined attack probability associated with a particular attack method. These are mapped to the risk using a *risk graph* approach (see [8, Table 9, Table 11]). In particular, for safety-related attack objective (non-zero safety component in severity), there is an additional parameter determined at Level 1, *controllability*: it represents the potential for the driver to influence the severity of the outcome. Safety related risk behaves differently according to this value (less controllable threats have higher risk outcomes).

2.2 Risk assessment: ISO/SAE 21434

ISO/SAE 21434 specifies engineering requirements for cybersecurity risk management regarding concept, product development, production, operation, maintenance and decommissioning of electrical and electronic systems in road vehicles, including their components and interfaces. This standard describes cybersecurity engineering from the perspective of a single item: an item comprises all electronic equipment and software (i.e. its components) on a vehicle that is involved in the realization of a specific functionality at vehicle level and it interacts with its operational environment. This is one of the most evident difference with EVITA, where the system under investigation is the whole automotive on-board network. In this sense, ISO/SAE 21434 provides a more flexible framework which can be tailored to accommodate the needs of a specific situation.

In the phase of Threat Analysis and Risk Assessment (TARA), the main differences between [7] and [8] are in the terms and definitions (attack probability and severity in [8] become respectively feasibility and impact in [7], for example), in the way the attack potential and risk values are discretized and in the use of different risk matrices in the risk graph approach. The reader is referred to Section 2.1 and to Clause 15, Annex F, Annex G of [7] for a comparison.

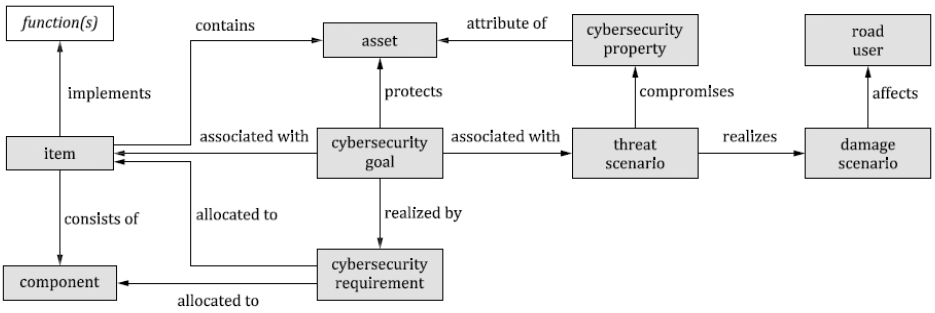


Figure 2.2: Relationship between terms and definitions in [7]

Following this standard, TARA comprises the following steps:

- asset identification: it is the identification of the assets, that is, the components of the system which, losing a property of cybersecurity, such as integrity, confidentiality and availability, generate a condition of the vehicle or one of its functionality that can damage a road user;
- threat identification: it is the identification of threats, that is, the causes of impairment of the asset that allow the realization of the condition of damage to the road user;
- impact estimate: similarly to EVITA, it is the evaluation of the consequences on the road user that would occur following the realization of the damage condition with regard to safety, privacy, the financial aspect and the operational aspect;
- attack tree analysis: is the description of the steps that allow the implementation of the attack, that is the condition of damage to the road user;
- vulnerability identification: it is the identification of system vulnerabilities that can be exploited to traverse an attack tree and then carry out an attack;
- feasibility estimate: it is the assessment of the effort required to carry out the attack, taking into account aspects such as the expertise or

equipment required, the distance from the vehicle, the availability of the necessary information;

- risk calculation: ISO/SAE 21434 gives an example where feasibility is obtained by aggregating attack potential and discretized through [7, Table G.7] into 4 levels (in EVITA there were 5); the maximum value of impact among the safety, privacy, financial, operational fields is retained; impact and feasibility are combined through [7, Table H.8] to obtain risk on a 5 level scale (in EVITA there were 7/8 levels). ISO/SAE 21434 still gives organizations the possibility of tailoring this procedure.

Risk treatment provides that for each risk associated with a threat, a treatment option is chosen from the following possibilities:

- risk avoidance: for example by eliminating the source that generates the threats;
- risk reduction: defining controls that lower the feasibility of the attack associated with the threat;
- risk sharing or transfer: for example through the stipulation of an insurance;
- risk acceptance: its value is deemed acceptable.

2.3 Cost-effective risk treatment

2.3.1 General ideas

Taking as input the results of risk assessment, during risk treatment decision makers are faced with the problem of allocating a finite amount of resources in order to identify an adequate risk response. It is useful (and not unrealistic) to assume that an organization has developed a general risk management strategy. This strategy makes explicit the assumptions, constraints,

risk tolerances and priorities/trade-offs used within organizations for making investment and operational decisions.

The scope of this work is to develop a quantitative structure for integrating the results of risk assessment in the decision making process. The idea is to define quantities that measure various facets of a possible risk response and optimize some criteria based on these quantities. Since the structure of the problem naturally determines the presence of trade-offs and constraints, the spontaneous formulation is a multi-objective optimization.
In this way, we think we are able to address the question of risk *acceptance* and inform decisions about it. In a general scenario, an organization can respond to risk in different ways: acceptance, avoidance, mitigation, sharing, transfer or a combination of the above. However, in our application, we take decisions only for what concerns mitigation and acceptance, as these are the options that truly affect risks. A standard idea, in this situation, is to look at the risk assessment results and define a threshold: all the risks under the threshold are accepted, all the others must be mitigated. The appropriate course of action is determined according to a unique global criterion applied to one threat (and related risks) at a time. While it is easy to understand and apply in practice, the definition of the threshold can sometimes be arbitrary or follow criteria that are not system-specific; this kind of method does not really consider all the information we have on the system as a whole and ignores further aspects of the problem, for example costs of implementation. Facing a difficult decision with only one parameter, one degree of freedom, can be limiting.
Conversely, in our setting a single risk deriving from a single threat is deemed acceptable if the optimum solution tells us that we do not have to do anything against it. We are just assuming to have some criteria of optimality. One could argue that these criteria are as arbitrary as the one above, and this is the truth; still, it is certain that we are capturing more about the complex behaviour of risks and we are betting that this can be more informative for decision makers than just fixing an acceptance threshold. We face the

problem providing more flexibility to the decision maker, more room to play with different parameters.

2.3.2 Use cases

ISO/SAE 21434 defines requirements in a generic manner such that it can be applied to a variety of items and components, with the possibility of carrying out a reuse analysis or integrating out-of-context and off-the-shelf components. The standard applies to all tiers of the supply chain and prescribes that if a product is developed (at any tier), the organization must go through a cybersecurtity concept phase, described in [7], Clause 9. This phase involves consideration of vehicle level functionality, as implemented in items. The item and its operational environment are identified as an "Item definition", which forms the basis for the subsequent activities. This clause also specifies cybersecurity goals for the item which are the highest level of requirements. The cybersecurity concept consists of cybersecurity requirements and requirements on the operational environment, both of which are derived from the cybersecurity goals and based on a comprehensive view of the item. In this phase, threats are defined, related risks are evaluated and appropriate cybersecurity controls are identified.

The optimization framework is thought to have a similar degree of adaptability as ISO/SAE 21434, so that many possible use cases can be addressed. In this work, we had in mind three use cases:

1. **product development**: OEM or supplier analyzes possible choices of cybersecurity controls from the point of view of cost-effectiveness;

2. **supplier out-of-context product development**: an organization develops a product prior to engagement or commercial agreement with a customer; additional information on cybersecurity requirements by customers for previously required and similar products can be used to create a measure of *customer satisfaction* (see Section 2.4).

3. **supplier product development with customer requirements**: security controls specified by the customer are assumed to be already in the risk treatment, decisions are made on additional controls. Customer satisfaction can be tailored to fit this use case: we can either eliminate it or use requirements made only by other customers to weigh how they might be interested in the product.

However, adaptability is really the essence of the multi-objective optimization method: for example, if another use case arises it is possible to introduce new objective functions that measure new facets of that use case. Furthermore, this modus operandi could be adapted to help decision making in other phases of the product development cycle, other than concept phase (for example, the weakness and vulnerability analysis during design and testing phases).

2.4 Mathematical formulation

In this section we introduce definitions and objects that are intended for a static analysis, but in Chapter 3, where we develop a dynamic analysis, these same objects become the initial values of stochastic processes. For this reason, some of the notations include a subscript 0 to indicate the initial time.

2.4.1 Inputs

Adopting definitions and terms from ISO/SAE 21434, we have a finite number of *threat scenarios* and *security controls* against these threat scenarios. The respective sets are TS and SC; we indicate the generic element of TS with i and the generic element of SC with j, so that we often write $i \in TS$ and $j \in SC$.
There is a map from threat scenarios to security controls that mitigate them, i.e. $M : TS \to \mathcal{P}(SC)$:

$$M(i) := \{j \in SC \mid j \text{ mitigates } i\}.$$

From the output of risk assessment, conducted for example via ISO/SAE 21434, we retain the following data:

- for each i in TS, I^i is the maximum value among the fields that compose the impact vector (safety, financial, operational, privacy);
- for each i in TS, $\mathbf{F}_0^i = (f_0^{i,1}, f_0^{i,2}, f_0^{i,3}, f_0^{i,4}, f_0^{i,5})$ is the feasibility vector composed of expertise, knowledge of the item, equipment, window of opportunity and elapsed time required to realize threat scenario i ;
- for each j in SC, $\mathbf{G}_0^j = (g_0^{j,1}, g_0^{j,2}, g_0^{j,3}, g_0^{j,4}, g_0^{j,5})$ is the feasibility vector composed of expertise, knowledge of the item, equipment, window of opportunity and elapsed time required for the step of the attack path that bypasses the security control j .

ISO/SAE 21434 describes the procedure to obtain risk from impact scalar I and feasibility vector $\mathbf{F}$. This function $r = r(I, \mathbf{F})$ is briefly described in Table 2.1 with its thresholds and cutoffs, while the reader is referred to [7] for all the explanations.

	$\sum_{h=1}^{5} f^h$			
	$[0, 13]$	$[14, 19]$	$[20, 24]$	$[25, 57]$
I=0	1	1	1	1
I=1	3	2	2	1
I=2	4	3	2	1
I=3	5	4	3	2

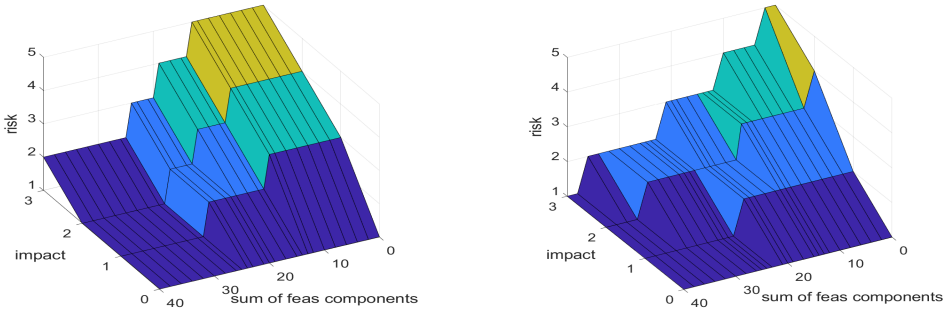
Table 2.1: ISO/SAE 21434 risk function

According to ISO/SAE 21434, however, risk values may also be determined by a risk formula defined by the organization. A customized formula we will adopt is the following:

$$r(I, \mathbf{F}) = round\left(1 + I \cdot \left(\frac{57 - \sum_{h=1}^{5} f^h}{57}\right)^2\right) \tag{2.4.1}$$

where $\mathbf{F} = (f^1, f^2, f^3, f^4, f^5)$ is the feasibility vector, 57 is the maximum possible sum of the components of $\mathbf{F}$ according to ISO/SAE 21434 (so that $(57 - \sum_{h=1}^{5} f^h)/57$ is a normalized value between 0 and 1) and the operator *round* rounds a real number to the nearest integer.

These two risk function share two features: they take values on integers from 1 to 5 and they are increasing in I and decreasing in each component of $\mathbf{F}$. The motivation for the introduction of this new function can be seen in



(a) ISO/SAE 21434 (b) Customized

Figure 2.3: Risk functions comparison

Figure 2.3: customized risk function is more varied in the region where the sum of feasibility components is higher than 25; thus, it differentiates risks more than ISO/SAE 21434 risk function, which has a flatter profile in that region.

Whatever the risk function r we decide to adopt, we now give some definitions. For each $i \in TS$, we define

$$R^i := r(I^i, \mathbf{F}_0^i)$$

to be the risk of threat scenario i.

For each $i \in TS, j \in SC$ we can compute $R^{i \leftarrow j}$ as the risk of threat scenario i after the implementation of the security control j. In our application, security controls influence feasibility of the attack paths and have no effect on the impact of the damage scenario; the new feasibility is obtained by the analysis of the attack paths: we consider the feasibility of the attack path

plus a step that bypasses the security control by taking the feasibility vector between $\mathbf{F}_0^i$ and $\mathbf{G}_0^j$ that maximizes the sum of its components:

$$R^{i\leftarrow j} := \begin{cases} r(I^i, \arg\max_{\mathbf{X}\in\{\mathbf{F}_0^i,\mathbf{G}_0^j\}} \sum_{h=1}^{5} x^h) & \text{if } j \in M(i) \\ R_i = r(I^i, \mathbf{F}_0^i) & \text{if } j \notin M(i) \end{cases} \tag{2.4.2}$$

By definition, R^i and $R^{i\leftarrow j}$ are integer from 1 to 5. We will clearly have that $1 \leq R^{i\leftarrow j} \leq R^i$, so that security measures never increase risk and in general they do not eliminate it completely.

For each $j \in SC$, the cost of implementation is $w_j \in \mathbb{R}$, measured in hours of effort required for the implementation.

In the use cases 1 and 2 from Subsection 2.3.2, we consider the problem from the eye of the supplier and we assume to have data on security requirements made by OEMs in order to construct a measure of customer satisfaction. We have a mapping that links each customer with their security requirements. We can translate this map into a map from customers to security controls that match requests $CustReq : Cust \to \mathcal{P}(SC)$:

$$CustReq(k) := \{j \in SC \mid j \text{ matches } k\text{'s requests}\}$$

for each customer $k \in Cust$, where $Cust$ is a set composed of OEMs and k is the symbol for the generic element in $Cust$.
A decision is a choice of a subset sca of SC (an element of $\mathcal{P}(SC)$), which we will refer to as a *security control alternative*.

2.4.2 Objective functions

We present here the objective functions that we intend to optimize.

Definition 2.1. The *average risk reduction* (we will often refer to it simply as *risk reduction*) after implementation of sca is:

$$\Delta Risk(sca) := \frac{1}{|TS|} \sum_{i\in TS} \left(R^i - \min_{j\in sca} R^{i\leftarrow j}\right)$$

It is a global measure of the performance of the *sca* on the whole system. This formula comes from the assumption that the risk of a threat scenario i after the implementation of a security control alternative *sca* is determined by the most effective of the security controls in *sca*. We take the average over the n threat scenarios so that the resulting value is normalized and is on the scale prescribed by ISO/SAE 21434. If a decision maker considers some threat scenarios more important than others, it is possible to introduce non-uniform weights in the sum over TS. We consider the simpler uniform case, but all the properties we show next generalize to all choices.
Clearly this objective function is meant to be maximized.
It is straightforward to see that we have the following monotonicity property.

Theorem 2.4.1 (monotonicity)**.** *Assume* $sca_1, sca_2 \in \mathcal{P}(SC), sca_1 \subseteq sca_2$. *Then*

$$\Delta Risk(sca_1) \leq \Delta Risk(sca_2)$$

Another property is the subadditivity.

Theorem 2.4.2 (subadditivity)**.** *For all* $sca_1, sca_2 \in \mathcal{P}(SC)$

$$\Delta Risk(sca_1 \cup sca_2) \leq \Delta Risk(sca_1) + \Delta Risk(sca_2).$$

Proof. We can fix a threat scenario i and compare the i-th term of the sum in $\Delta Risk(sca_1 \cup sca_2)$ with the i-th in $\Delta Risk(sca_1) + \Delta Risk(sca_2)$: we show that

$$R^i - \min_{j \in sca_1 \cup sca_2} R^{i \leftarrow j} \leq (R^i - \min_{j \in sca_1} R^{i \leftarrow j}) + (R^i - \min_{j \in sca_2} R^{i \leftarrow j}).$$

We need to focus on the set of security controls that realize the minimum:

$$ArgMin(i) := \{\overline{j} \in sca_1 \cup sca_2 \mid R^{i \leftarrow \overline{j}} = \min_{j \in sca_1 \cup sca_2} R^{i \leftarrow j}\}.$$

Let $\overline{j}$ be an element of $ArgMin(i)$, $\overline{j}$ must be in sca_1 or in sca_2: suppose $\overline{j} \in sca_1$, for example. Then

$$R^i - \min_{j \in sca_1 \cup sca_2} R^{i \leftarrow j} = R^{i \leftarrow \overline{j}} = R^i - \min_{j \in sca_1} R^{i \leftarrow j}$$

and the fact that $R^i - \min_{j \in sca_2} R^{i \leftarrow j} \geq 0$ gives us the result. If $\overline{j} \in sca_2$, we proceed in the same way. □

The interesting fact here is that, even for sca_1, sca_2 such that $sca_1 \cap sca_2 = \emptyset$, it can happen that

$$\Delta Risk(sca_1 \cup sca_2) < \Delta Risk(sca_1) + \Delta Risk(sca_2)$$

because the same threat scenario can be mitigated by a control in sca_1 and one in sca_2 and there is no benefit from this cooperation in $\Delta Risk$. This situation is frequent in any application, as in ours (see Section 2.5).

Taking a closer look at the proof above, we notice that it is possible to strengthen the result: $\Delta Risk$ satisfies the so called *submodularity* property.

Theorem 2.4.3 (submodularity)**.** *For all* $sca_1, sca_2 \in \mathcal{P}(SC)$

$$\Delta Risk(sca_1 \cup sca_2) + \Delta Risk(sca_1 \cap sca_2) \leq \Delta Risk(sca_1) + \Delta Risk(sca_2).$$

Proof. For each $i \in TS$, we show that:

$$(R^i - \min_{j \in sca_1 \cup sca_2} R^{i \leftarrow j}) + (R^i - \min_{j \in sca_1 \cap sca_2} R^{i \leftarrow j}) \leq (R^i - \min_{j \in sca_1} R^{i \leftarrow j}) + (R^i - \min_{j \in sca_2} R^{i \leftarrow j}).$$

As above, let $\overline{j}$ be an element of $ArgMin(i)$: suppose $\overline{j} \in sca_1$, for example. Then

$$R^i - \min_{j \in sca_1 \cup sca_2} R^{i \leftarrow j} = R^{i \leftarrow \overline{j}} = R^i - \min_{j \in sca_1} R^{i \leftarrow j}$$

and in general

$$R^i - \min_{j \in sca_1 \cap sca_2} R^{i \leftarrow j} \leq R^i - \min_{j \in sca_2} R^{i \leftarrow j}$$

because of monotonicity ($sca_1 \cap sca_2 \subset sca_2$). □

Submodularity is an interesting property, which deserves a little digression. We introduce a more general framework in the next definition.

Definition 2.2 (submodular set function)**.** Let X be a set and $v : \mathcal{P}(X) \to \mathbb{R}$ a function. This kind of function is often referred as *set function*. v is said to be *submodular* if

$$\forall S, T \subseteq X, v(S) + v(T) \geq v(S \cup T) + v(S \cap T).$$

Notice that a non-negative submodular function is also a subadditive function, but a subadditive function need not be submodular. Submodular functions have a natural characterization which makes them suitable for many applications, as stated by the following result.

Theorem 2.4.4 (characterization of submodularity)**.** *Let X be a finite set and $v : \mathcal{P}(X) \to \mathbb{R}$ a function. These conditions are equivalent:*

1. *v is submodular;*
2. *$\forall S, T \subseteq X$ with $S \subseteq T$ and $\forall x \in X \setminus T$ we have that $v(T \cup \{x\}) - v(T) \leq v(S \cup \{x\}) - v(S)$;*
3. *$\forall S \subseteq T \subseteq X$ and $U \subseteq X \setminus T$ we have that*

$$v(T \cup U) - v(T) \leq v(S \cup U) - v(S).$$

Proof. 1 $\implies$ 2) Condition 2 is an immediate instantiation of Condition 1.
2 $\implies$ 3) Let $U = \{u_1, \ldots, u_n\}$.

$$\begin{aligned} & v(T \cup U) - v(T) \\ = v(T \cup U) & - \sum_{i=1}^{n} \big(v(T \cup \{u_1, \ldots, u_i\}) - v(T \cup \{u_1, \ldots, u_i\})\big) - v(T) \\ & = \sum_{i=1}^{n} \big(v(T \cup \{u_1, \ldots, u_i\}) - v(T \cup \{u_1, \ldots, u_{i-1}\}\big) \\ \overset{(2)}{\leq} & \sum_{i=1}^{n} \big(v(S \cup \{u_1, \ldots, u_i\}) - v(S \cup \{u_1, \ldots, u_{i-1}\}\big) = v(S \cup U) - v(S) \end{aligned}$$

where clearly $v(T \cup \{u_1, \ldots, u_{i-1}\}) = v(T)$ for $i = 1$.
3 $\implies$ 1) Let $S, T \subseteq X$ where we assume $S \neq T$. We define $S' = S \cap T$,

$U = S \setminus T$, $T' = T$. Then by Condition 3

$$\begin{aligned} & v(S' \cup U) - v(S') \geq v(T' \cup U) - v(T') \\ \overset{\text{def. } S',T',U}{\iff} & v((S \cap T) \cup (S \setminus T)) - v(S \cap T) \geq v(T \cup (S \setminus T)) - v(T) \\ \iff & v(S) - v(S \cap T) \geq v(S \cup T) - v(T); \end{aligned}$$

by rearranging terms we obtain

$$v(S) + v(T) \geq v(S \cup T) + v(S \cap T)$$

□

Condition 2 is the so called *diminishing returns* property: the difference in the incremental value of the function that a single element makes when added to an input set decreases as the size of the input set increases. In our application, this condition informally states that it gets more difficult to have a gain in $\Delta Risk$ by adding controls as the number of already present controls increases.
Condition 3 is the *group diminishing returns* property, it does not add any particular meaning but it is useful in the proof.

Notice that if X is not assumed to be finite, the conditions are not equivalent. As a counterexample, let $v(S) = 1$ if S is finite and $v(S) = 0$ if S is not finite: v satisfies the diminishing return condition 2, but is not submodular (take S, T infinite sets, with finite intersection).

Returning to the mathematical formulation of our optimization problem, we define another quantity that is involved in the objective functions.

Definition 2.3. The *average customer satisfaction* (we will often refer to it simply as *customer satisfaction*) for *sca* is:

$$CustSat(sca) := \frac{1}{|Cust|}\left(\sum_{k \in Cust} \Delta Risk(sca \cap CustReq(k))\right)$$

where we have $\Delta Risk(J) = 0$ if $J = \emptyset$.

The formula looks involved but it can be interpreted as a projection of the risk reduction onto the customer requirements. The main advantage of this formula is that the defined quantity is comparable to the risk reduction, so the two can be summed to have a single objective function. By comparable we mean that average risk reduction and average customer satisfaction are on the same scale (as one is defined through the other) and that they are in the following relation, which is an immediate consequence of the monotonicity property in Theorem 2.4.1:

$$CustSat(sca) \leq \Delta Risk(sca),\ \forall sca. \tag{2.4.3}$$

$CustSat$ inherits all the properties of $\Delta Risk$, as they are preserved under linear combinations and under the projection onto $CustReq(k)$. In particular, $CustSat$ is monotone and submodular.
Obviously, the customer satisfaction objective function is meant to be maximized.

Definition 2.4. The *cost* of an *sca* is:

$$Cost(sca) := \sum_{j \in sca} w_j$$

Clearly, cost is intended to be minimized.

2.4.3 Optimization problems

With these definitions, we have defined a flexible framework where we can formulate different problems. A multi-objective optimization problem can be formulated as

$$\min(u_1(\vec{x}), \ldots, u_k(\vec{x})) \text{ s.t. } \vec{x} \in X$$

where $k \geq 1$ is the number of objectives ($k \geq 2$ when the problem is truly multi-objective), $\vec{u} : X \to \mathbb{R}^k, \vec{u}(\vec{x}) = (u_1(\vec{x}), \ldots, u_k(\vec{x}))$ is the vector-valued objective function and X is the feasible set. This is without loss of generality as maximizing $u_i(\vec{x})$ is equivalent to minimizing $-u_i(\vec{x})$.

In multi-objective optimization, there does not typically exist a feasible solution that optimizes all objective functions simultaneously. Therefore, we are interested in solutions that can not be improved in any of the objectives without degrading at least one of the other objectives, *Pareto optimal* solutions.

Definition 2.5. A feasible solution $\vec{x}_1 \in X$ is said to *Pareto dominate* another solution $\vec{x}_2 \in X$ if $u_i(\vec{x}_1) \leq u_i(\vec{x}_2), \forall i$ and there exists j such that $u_j(\vec{x}_1) < u_j(\vec{x}_2)$.
A solution $\vec{x} \in X$ is called *Pareto optimal* if there does not exist another solution that dominates it. The set of Pareto optimal outcomes is often called the *Pareto front* or *Pareto boundary*.

In our application, here are some possible formulations.

Problem 2.1 (unconstrained multi-objective)**.** We want to maximize the sum of risk reduction and customer satisfaction while minimizing cost:

$$\max_{sca \in \mathcal{P}(SC)} \Delta Risk(sca) + CustSat(sca), \min_{sca \in \mathcal{P}(SC)} Cost(sca).$$

This can be generalized to include a multiplicative constant α that regulates the importance that decision maker gives to customer satisfaction:

$$\max_{sca \in \mathcal{P}(SC)} \Delta Risk + \alpha\, CustSat(sca), \min_{sca \in \mathcal{P}(SC)} Cost(sca).$$

$\alpha = 1$ is the choice we will usually adopt.
$\alpha = 0$ can be adopted for the third use case in Subsection 2.3.2, when the supplier is developing the product for a single customer, in order to eliminate the contribution of customer satisfaction.
A decision maker that wants to approximately put the same weight on risk reduction and customer satisfaction can fix $\alpha = \frac{\Delta Risk(SC)}{CustSat(SC)}$, which is greater than 1 because of (2.4.3). This value of α is precisely intended to balance the inequality in (2.4.3), at least in the case of a full decision $sca = SC$.
Because of (2.4.3), the objective function given by the sum of risk reduction and customer satisfaction can be interpreted as a perturbation of the risk reduction that takes into account the presence of customers.

We can then lay out a constrained version.

Problem 2.2 (constrained)**.** Let C be a determined budgetary capacity. We want to maximize the sum of risk reduction and customer satisfaction under the budget constraint:

$$\max_{sca \in \mathcal{P}(SC)} \Delta Risk + \alpha\, CustSat(sca) \text{ s.t. } Cost(sca) \leq C$$

We impose a constraint on cost and avoid a constraint on risk reduction for two reasons: first, it is reasonable to assume that an organization has a good estimate on the amount of money that can be allocated to risk treatment; on the other hand, as we explained in Subsection 2.3.1, we want to face the risk acceptance problem providing more flexibility and a constraint on risk reduction would decrease it.

2.5 Example of application: static version

The use case under analysis in the example is the second use case of Subsection 2.3.2: a tier 1 supplier is developing an out-of-context product for its market composed of various customers. As the example arises from a real threat analysis and risk assessment, we do not give all the information at TARA level. In particular, we omit the details of the attack paths (which would also be too long to report) and threat scenarios; we refer to threat scenarios simply with their IDs ('TS001','TS002',etc.). We omit names of customers when reporting their requirements.
We outline the main aspects of this use-case adopting both ISO/SAE 21434 and 2.4.1 risk functions.

All computations are performed using MATLAB.

2.5.1 Inputs

In this example, we have a set TS composed of 9 threat scenarios, a set SC composed of 6 security controls and a set $Cust$ composed of 5 customers.

There are $2^{|SC|} - 1 = 2^6 - 1 = 64 - 1 = 63$ possible decisions sca. Clearly, decisions such that $|sca| < 4$ are not really effective against risk and can not be considered in a serious decision making process, but we report results for them anyway. With reference to the security solutions examples introduced in Section 1.3, the available security controls in this example are: Secure On-Board Communication, Secure Diagnostics, Secure Debug Access, Authenticated Boot, Authenticated Software Update, Firewall.
In Table 2.2 and Table 2.3, we can see how security controls affect risk value (through feasibility) of each threat scenario summarized by the $R^{i \leftarrow j}$ matrix, obtained through formula 2.4.2 respectively with ISO/SAE 21434 risk function and (2.4.1) risk function. We only report the $R^{i \leftarrow j}$ matrices without giving the details of the function M, the impacts and feasibility vectors that determine this result.
In this example, costs of implementation are in Table 2.4 while customers' requests are given by the $CustReq$ function summarized in Table 2.5, where 'x' indicates that the customer on the row requires the security solution on the column.

	sec oc	sec diag	sec debug acc	auth boot	auth sw update	firewall
TS004	1	4	4	4	4	4
TS005	4	1	4	4	4	4
TS006	3	3	2	2	3	3
TS007	5	2	5	5	5	5
TS008	4	2	4	2	2	4
TS009	4	4	4	4	4	3
TS010	3	3	3	2	3	3
TS011	4	2	4	2	2	4
TS014	3	3	2	3	3	3

Table 2.2: $R^{i \leftarrow j}$ with ISO/SAE 21434 risk function

	sec oc	sec diag	sec debug acc	auth boot	auth sw update	firewall
TS004	1	3	3	3	3	3
TS005	3	2	3	3	3	3
TS006	2	2	2	1	2	2
TS007	4	2	4	4	4	4
TS008	4	2	4	2	2	3
TS009	3	3	3	3	3	2
TS010	2	2	2	2	1	2
TS011	3	2	3	1	1	3
TS014	3	3	2	3	3	3

Table 2.3: $R^{i \leftarrow j}$ with customized risk function

security control	cost (hrs of effort)
Secure On-Board Communication	840
Secure Diagnostics	200
Secure Debug Access	280
Authenticated Boot	680
Authenticated Software Update	920
Firewall	160

Table 2.4: Estimated costs of implementation measured in hours of effort

	sec oc	sec diag	sec debug acc	auth boot	auth sw update	firewall
Cust 1		x	x	x	x	x
Cust 2		x			x	
Cust 3		x	x	x	x	x
Cust 4		x	x		x	
Cust 5			x		x	

Table 2.5: Customers' requests

2.5.2 Optimization results: static version

As in the example there are not too many threat scenarios or security controls, the unconstrained multi-objective formulation in Problem 2.1 is sufficient: maximizing the combination of risk reduction and customer satisfaction while minimizing cost. A constraint on cost would fit better the case of a more wide range of decisions.

Here we explain details and legend of the figures, valid hereafter in this work:

- '1' denotes that the *sca* contains Secure On-Board Communication, '2' Secure Diagnostics, '3' Secure Debug Access, '4' Authenticated Boot, '5' Authenticated Software Update, '6' Firewall;

- yellow line connects data points on the Pareto Front, where any of the two objective functions can not be improved without degrading the other;

- data points highlighted with cyan squares are four 'special' security control alternatives: $\{2, 3, 5\}$ is the *sca* which contains the most requested security controls according to Table 2.5 (Secure Diagnostics, Secure Debug Access and Authenticated Software Update); starting from this one we have $\{2, 3, 4, 5\}$, $\{2, 3, 5, 6\}$ and $\{2, 3, 4, 5, 6\}$ which respectively add Authenticated Boot, Firewall and both; the decision maker knows in advance that these are practical and suitable to be implemented in this use case, so that we could seek to validate their goodness;

- 'max risk left' is an interesting discrete value that is not among the objective functions involved in the optimization but still is reported in the figure; for each *sca*, we assign a different colour according to this value:

$$\max_{i \in TS} \min_{j \in sca} R^{i \leftarrow j}.$$

We report in Figure 2.4 the results for $\alpha = 1$ and in Figure 2.5 for $\alpha = \frac{\Delta Risk(SC)}{CustSat(SC)} = \frac{1.8889}{1.2444} = 1.5179$, obtained adopting ISO/SAE 21434 risk function.
In Figure 2.6, we report the results for $\alpha = 1$ and in Figure 2.7 for $\alpha = \frac{\Delta Risk(SC)}{CustSat(SC)} = \frac{1.4444}{0.9333} = 1.5179$, obtained adopting customized risk function.

We begin our analysis with some remarks on single security solution. Secure On-Board Communication has an ambivalent nature. No customer requires this solution yet (see Table 2.5) because it is known that it is really difficult to implement it in practice: it involves more than one ECU (at least the one transmitting the authenticated and/or encrypted message and all the the ECUs receiving that message); on top of that, it relies upon the complex mechanisms of key negotiation and update between all involved ECUs. However, despite its high cost (see Table 2.4), in this experiment Secure On-Board Communication is often present on the Pareto frontier (as it can be seen in the Figures). This is due to the fact that Secure On-Board Communication is the only solution that mitigates threat scenario 'TS004'; its risk before mitigation is 4 for ISO/SAE 21434 risk function and 3 for customized risk function and after the implementation of Secure On-Board Communication it becomes respectively 2 and 1 (see Tables 2.2, 2.3), which represents an important risk reduction. What we want to prove, both with quantitative and qualitative arguments, is that we can avoid the implementation of Secure On-Board Communication and still achieve a good performance of the risk response; in a certain sense this means that the risk of threat scenario 'TS004' is acceptable, and risk acceptability is the central question we are addressing in this work.

All the the decisions that do not include Secure Diagnostics (numbered as '2' in the figures) have the higher 'max risk left' (5 for ISO/SAE 21434, 4 for customized risk function); that is because Secure Diagnostics is the only solution that mitigates 'TS007' which has maximum risk. Then, Secure Diagnostics is one of the most required solution by customers (see Table 2.5) and has a low cost (200 hrs of effort). In general, the absence of Secure

Diagnostics determines a worst result in $\Delta Risk + \alpha \cdot CustSat$ (it is more evident with ISO/SAE 21434 than with customized risk function, but true for both). All this evidence states that Secure Diagnostic must be present in any meaningful decision sca.

As mentioned before, $sca = \{2, 3, 5\}$ has a special role because it is composed by the most requested security controls in Table 2.5. Assuming $\{2, 3, 5\}$ as a starting point, we analyze possible courses of action that do not include Secure On-Board Communication (1). We introduce the following metric, which will guide us in the analysis.
For compactness, we introduce this notation:

$$\Sigma^{\alpha}(sca) := \Delta Risk(sca) + \alpha \cdot CustSat(sca).$$

Definition 2.6 (cost-effectiveness)**.** Let $sca_1, sca_2 \in \mathcal{P}(SC)$ be decisions such that $sca_1 \cap sca_2 = \emptyset$. Then the *cost-effectiveness* Γ of adding sca_2 to sca_1 is

$$\Gamma(sca_1 \leftarrow sca_2) := \frac{Cost(sca_1 \cup sca_2) - Cost(sca_1)}{(\Sigma^{\alpha}(sca_1 \cup sca_2) - \Sigma^{\alpha}(sca_1)) \cdot 10} = \frac{Cost(sca_2)}{(\Sigma^{\alpha}(sca_1 \cup sca_2) - \Sigma^{\alpha}(sca_1)) \cdot 10}$$

$\Gamma(sca_1 \leftarrow sca_2)$ measures the hours of effort required to obtain a 0.1 gain in $\Delta Risk + \alpha \cdot CustSat$ by implementing sca_2 in addition to sca_1. In this sense, lower values of Γ are to be preferred.

Proposition 2.5.1. *Γ is increasing in the first argument, meaning that for sca_1, sca_2, sca_3 such that $sca_1 \subset sca_2, sca_3 \cap sca_2 = \emptyset$ we have*

$$\Gamma(sca_1 \leftarrow sca_3) \leq \Gamma(sca_2 \leftarrow sca_3).$$

Proof. Σ^{α} is submodular because it is a linear combination of $\Delta Risk$, which is submodular (Theorem 2.4.3). Then Σ^{α} satisfies the diminishing return condition from Theorem 2.4.4 and it means that

$$\Sigma^{\alpha}(sca_1 \cup sca_3) - \Sigma^{\alpha}(sca_1) \geq \Sigma^{\alpha}(sca_2 \cup sca_3) - \Sigma^{\alpha}(sca_2).$$

These gains are in the denominator of $\Gamma(sca_1 \leftarrow sca_3), \Gamma(sca_2 \leftarrow sca_3)$, hence the result. □

In Figures 2.4, 2.5, 2.6, 2.7, the quantity $\Gamma(sca_1 \leftarrow sca_2)$ represents the slope (divided by 10) of the line that connects the data points

$$(Cost(sca_1), \Sigma^\alpha(sca_1)) \; ; \; (Cost(sca_1 \cup sca_2), \Sigma^\alpha(sca_1 \cup sca_2)).$$

In Table 2.6 we report the cost-effectiveness of adding $\{4\}, \{6\}, \{4,6\}$ to $\{2,3,5\}$ (the possible courses of action starting from $\{2,3,5\}$).

	ISO/SAE 21434		customized	
	$\alpha = 1$	$\alpha = 1.5179$	$\alpha = 1$	$\alpha = 1.5476$
$\Gamma(\{2,3,5\} \leftarrow \{4\})$	437	381	219	189
$\Gamma(\{2,3,5\} \leftarrow \{6\})$	103	90	103	89
$\Gamma(\{2,3,5\} \leftarrow \{4,6\})$	270	235	180	156
$\Gamma(\{2,3,5,6\} \leftarrow \{4\})$	437	381	219	189

Table 2.6: Cost-effectiveness analysis

From the table we observe that

$$\Gamma(\{2,3,5\} \leftarrow \{4\}) > \Gamma(\{2,3,5\} \leftarrow \{4,6\}) > \Gamma(\{2,3,5\} \leftarrow \{6\})$$

whatever the risk function and α we choose. Therefore, adding Firewall (6) to $\{2,3,5\}$ looks like the most cost-effective course of action. However, at least qualitatively, we have to account the fact that Σ^α has the diminishing return property; it informally means that it gets more difficult to have a gain in Σ^α as the value of Σ^α increases. In this sense, the results for $\Gamma(\{2,3,5\} \leftarrow \{4,6\})$ do not look so bad, considering $\Sigma^\alpha(sca = \{2,3,4,5,6\})$ is sensibly greater than $\Sigma^\alpha(sca = \{2,3,5,6\})$ (as it can be seen in the Figures). Observing that

$$\Gamma(\{2,3,5\} \leftarrow \{4\}) = \Gamma(\{2,3,5,6\} \leftarrow \{4\})$$

whatever the risk function and α we choose, we highlight that Authenticated Boot (4) and Firewall (6) have no overlap in their effect on our objective Σ^α:

when added to $\{2, 3, 5\}$, the gains that they bring are complementary. We conclude that Authenticated Boot (4) and Firewall (6) can be implemented together on top of the most requested solutions with a good cost-effectiveness.

We observe in conclusion that the considerations above are true whatever the risk function and α, but their validity is amplified when we choose $\alpha = \frac{\Delta Risk(SC)}{CustSat(SC)}$ and customized risk function. As we can see in Figure 2.7, $sca = \{2, 3, 4, 5, 6\}$ reaches the Pareto optimality with these choices.

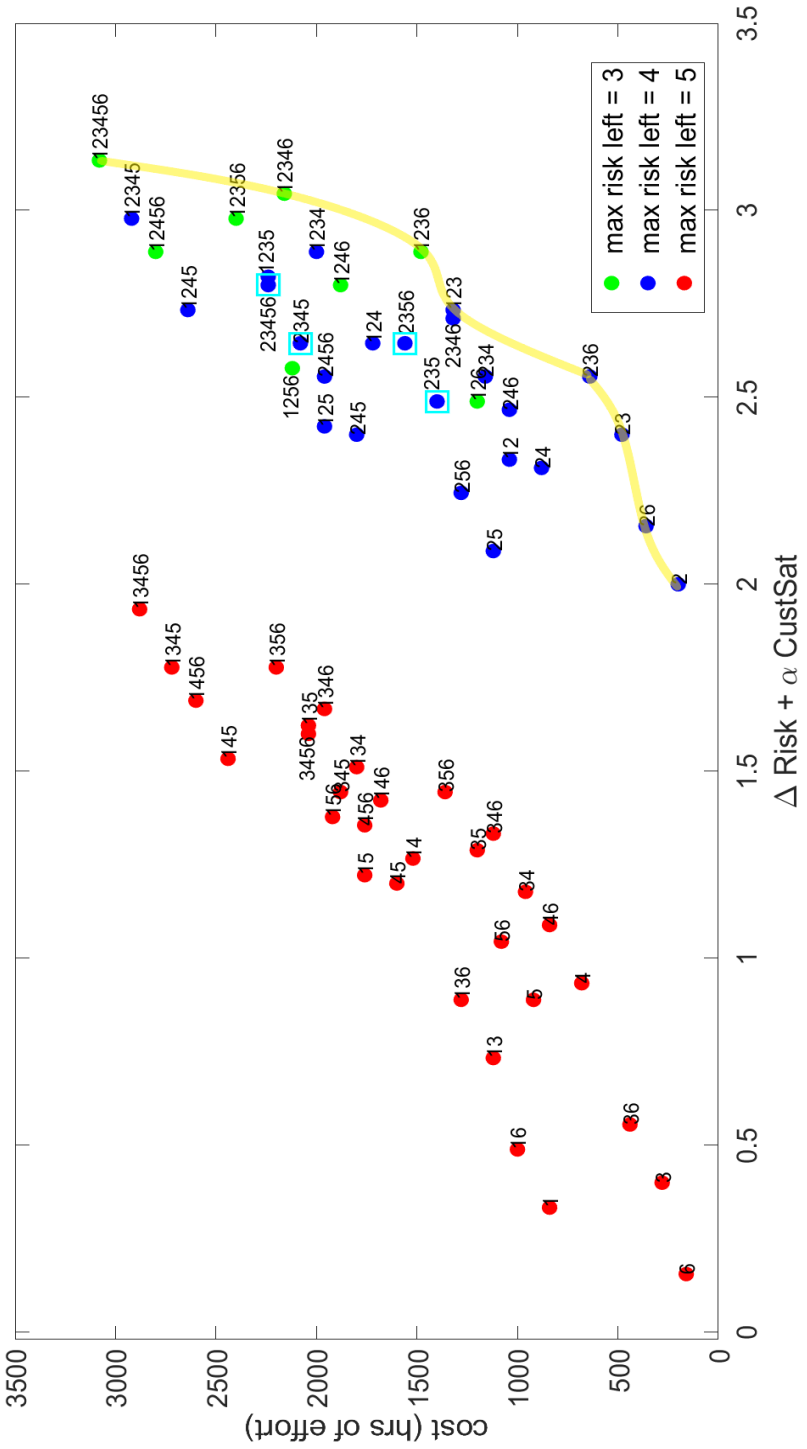


Figure 2.4: Results with ISO risk function, $\alpha = 1$

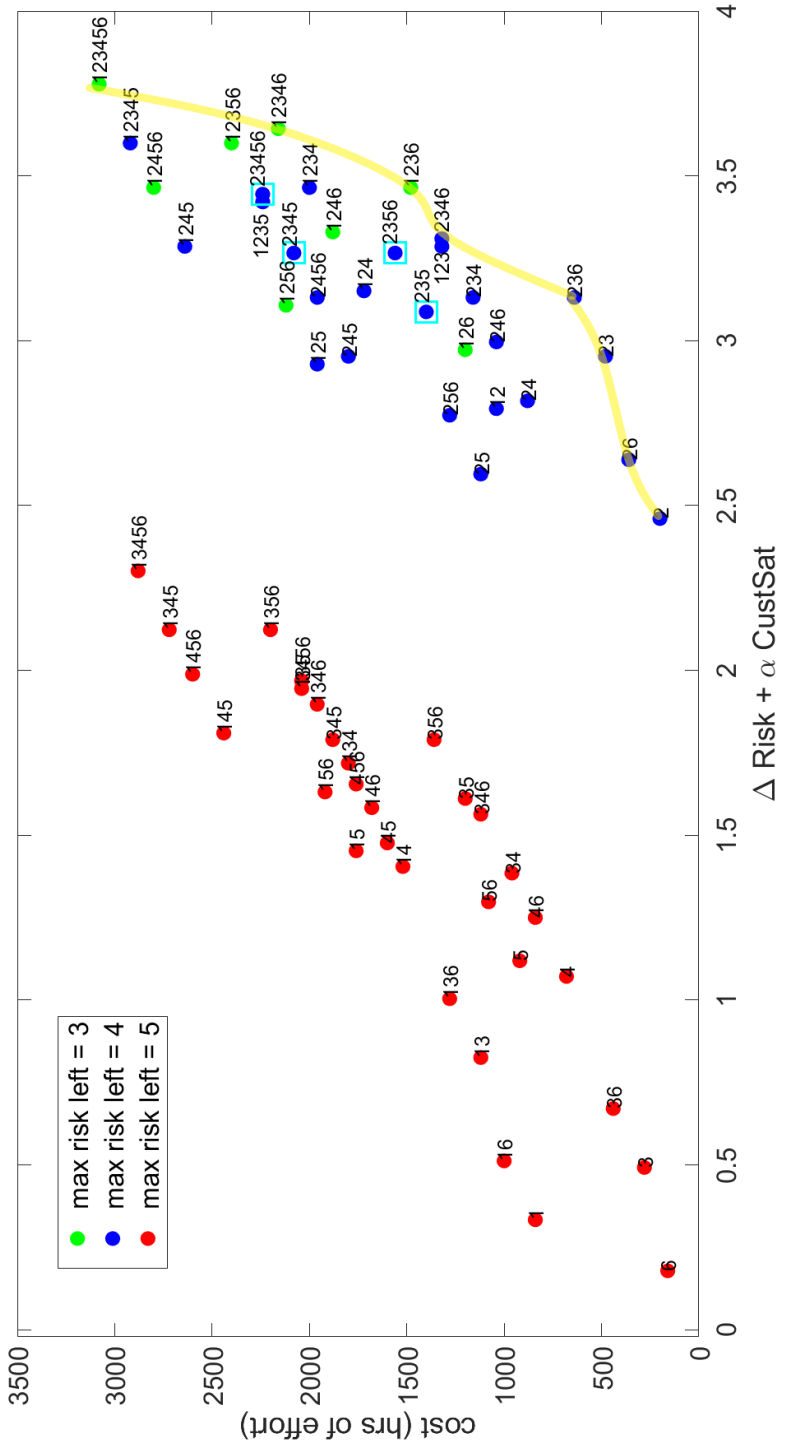


Figure 2.5: Results with ISO risk function, $\alpha = 1.5179$

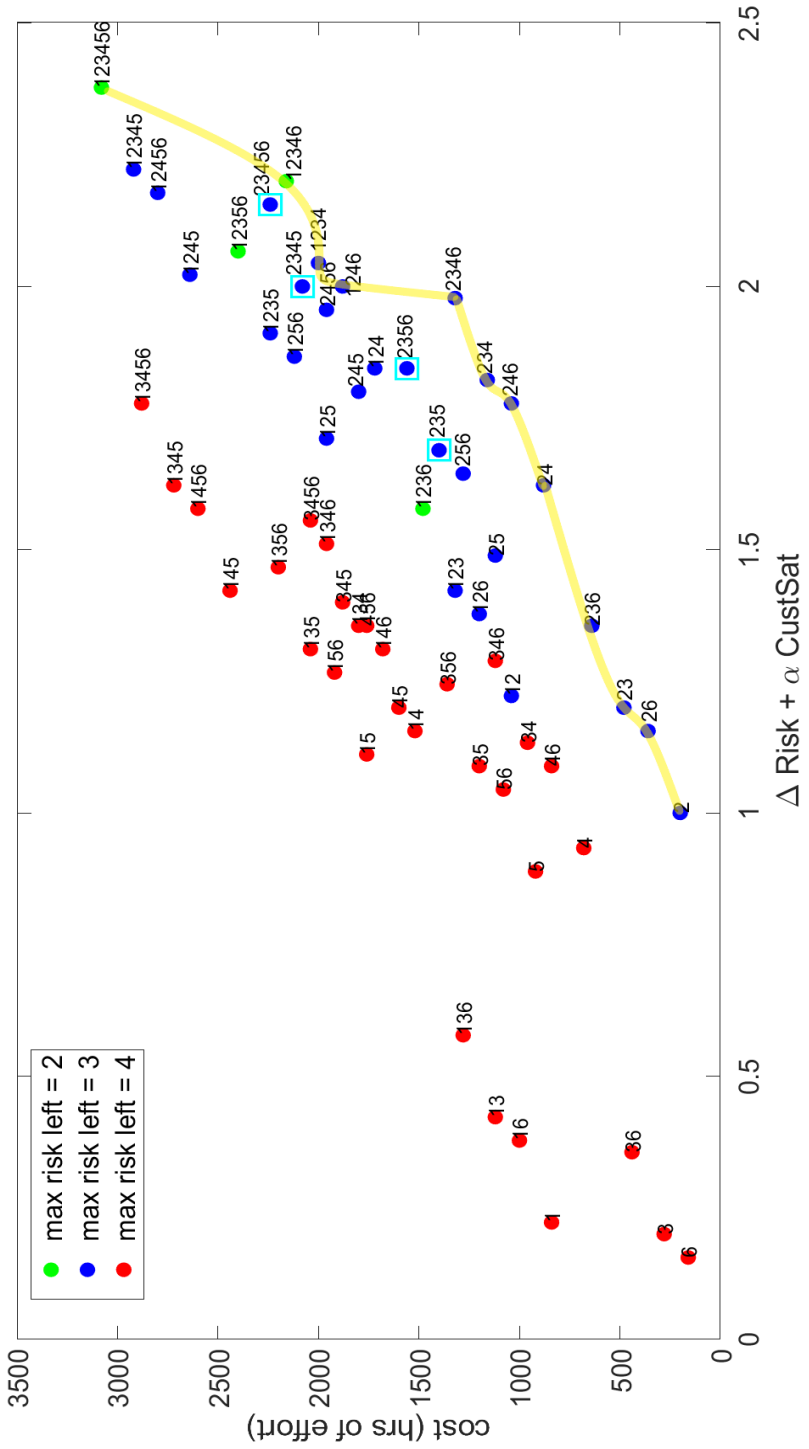


Figure 2.6: Results with customized risk function, $\alpha = 1$

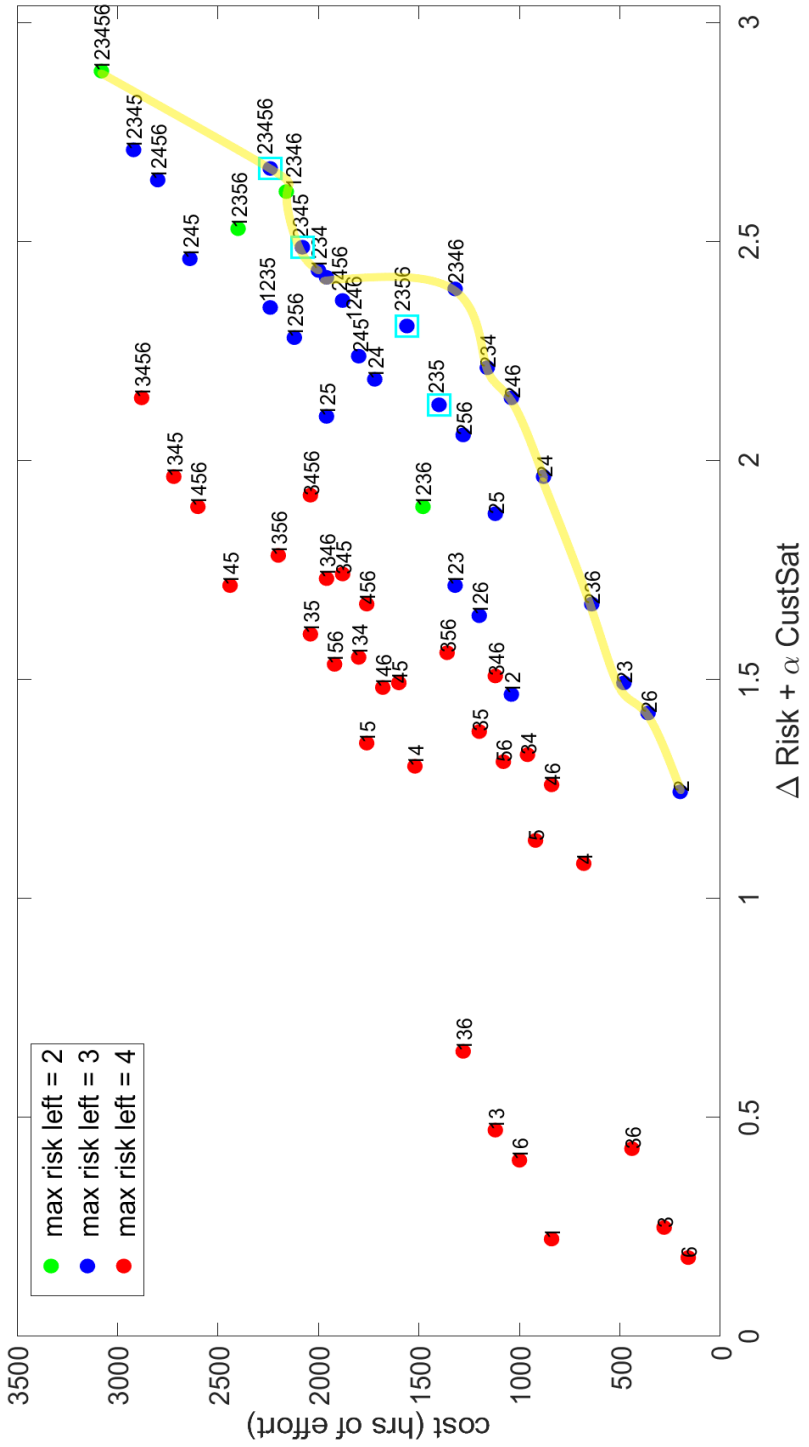


Figure 2.7: Results with customized risk function, $\alpha = 1.5476$

Chapter 3

Dynamic evolution of risk

3.1 Practical observations on risk evolution

Risk assessment is conducted considering available information on threats and security solutions at the time of development. However, the average lifespan of a vehicle is 10/15 years during which many aspects considered into the assessment can, and in fact will, change: technology improves, malicious attackers gain expertise about the system they are attacking, vulnerabilities are discovered, attack methods become well known, knowledge under non-disclosure agreement may become less secret due to data leakage and all the other infinite possibilities.

The aim of this chapter, clearly, is not predicting future. The idea is to introduce a non-deterministic model that takes into consideration that risk evolves and tries to capture some basic intuitions that experts have on how this evolution looks like. Estimations of future risk through this dynamic model and other derived quantities can then be plugged in the optimization framework introduced in Section 2.4 or in other types of analysis.

Risk of a threat scenario is obtained from the impact of the associated damage scenarios and the attack feasibility of the associated attack paths, as we detailed in Sections 2.1 and 2.2. It is reasonable to assume that the impact value stays constant, as the effects produced by the realization of an

attack are always the same. Conversely, we want to model the evolution of the fields determining the feasibility, as they are the ones that are clearly influenced by ever-changing external factors like the ones mentioned before.

The mathematical tool we have decided to adopt for this purpose is a well-known and deeply studied object: *discrete-time Markov chains*. We give a brief presentation of the first simple definitions, results and formalism for Markov chains in Appendix A.

One could ask the rationale of this particular modeling tool in this particular application. After all, we could have modeled just the distribution of the random variable of interest at final time, without needing to introduce processes and Markov dynamics. The motivations of this choice are to be found in the use case examples. In the first place, when it came to estimating the distributions according to which feasibility values evolved, it has immediately become clear that it was easier and more solid to estimate a 5 years transition rather than a 15 years one, so that we needed to model entire processes rather than just final random variables. This choice offers also the advantage of letting us compute interesting quantities at intermediate times, which is not possible if we model only a random variable. It could be interesting considering that common contracts in the automotive industry (between parties at different levels of the supply chain) are structured with an initial period of maintenance and an optional period to be renegotiated after 3/5/6 years. After these considerations, a Markov type dynamic is the most natural choice in absence of further information.

3.2 Markov chains and expected values

3.2.1 Definitions

We adopt definitions, terms and formalism from Appendix A.

For $k \in \{1, \dots, 5\}$, $P_k = ((p_k)_{h,l})_{h,l=0}^{|S_k|-1}$ is the transition matrix for the k-th field of feasibility and its dimension is $|S_k| \times |S_k|$. P_k is the same for all threat scenarios and security controls.

$\delta(f_0^{i,k})$ is the initial distribution of the k-th field for a given threat scenario $i \in TS$ ($\delta(g_0^{j,k})$ for a given security control $j \in SC$): it is a row vector all centered in the initial estimate given by the risk assessment (so it is a Dirac δ). This assumption is natural and simple but can be debated: one of the advantages of the setup is that it is possible to introduce some degree of uncertainty by changing the initial distribution if one thinks that the estimation of the initial value has some margin of error.

As we will detail in the example, the transition matrices P_k are not full because we assume that the processes have a decreasing direction and some transitions are not allowed. In particular the transition matrices have a diagonal block structure (resulting from transitions that are not allowed) and are lower triangular (resulting from the decreasing direction). Consequently, in the application many of the transition probabilities involved are null and we will use this sparsity to simplify computations.

For each $i \in TS$, we define five stochastic processes on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. These represent the five feasibility fields for threat scenario i from initial time $n = 0$ to final time $N > 0, N \in \mathbb{N}$. With the notation introduced after Theorem A.2.1, we give the following definitions:

- $(F_n^{i,1})_{n=0}^N$ is the specialist expertise, takes integer values in $S_1 = [0, 8]$ and is a $Markov(\delta(f_0^{i,1}), P_1)$ chain;
- $(F_n^{i,2})_{n=0}^N$ is the knowledge of the item, takes integer values in $S_2 = [0, 11]$ and is a $Markov(\delta(f_0^{i,2}), P_2)$ chain;
- $(F_n^{i,3})_{n=0}^N$ is the equipment, takes integer values in $S_3 = [0, 9]$ and is a $Markov(\delta(f_0^{i,3}), P_3)$ chain;
- $(F_n^{i,4})_{n=0}^N$ is the window of opportunity, takes integer values in $S_4 = [0, 10]$ and is a $Markov(\delta(f_0^{4,1}), P_4)$ chain;
- $(F_n^{i,5})_{n=0}^N$ is the elapsed time, takes integer values in $S_5 = [0, 19]$ and is a $Markov(\delta(f_0^{i,5}), P_5)$ chain;

where $(f_0^{i,1}, f_0^{i,2}, f_0^{i,3}, f_0^{i,4}, f_0^{i,5})$ is the output of the risk assessment as in Subsection 2.4.1. We use the notation $\mathbf{F}_n^i = (F_n^{i,1}, F_n^{i,2}, F_n^{i,3}, F_n^{i,4}, F_n^{i,5})$ for the whole random vector.

Similarly, for each $j \in SC$, we define five stochastic processes on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. These represent the five feasibility fields for the step of the attack path that bypasses security control j from initial time $n = 0$ to final time $N > 0$:

- $(G_n^{j,1})_{n=0}^N$ is the specialist expertise and takes integer values in $S_1 = [0, 8]$ and is a $Markov(\delta(g_0^{j,1}), P_1)$ chain;
- $(G_n^{j,2})_{n=0}^N$ is the knowledge of the item, takes integer values in $S_2 = [0, 11]$ and is a $Markov(\delta(g_0^{j,2}), P_2)$ chain;
- $(G_n^{j,3})_{n=0}^N$ is the equipment, takes integer values in $S_3 = [0, 9]$ and is a $Markov(\delta(g_0^{j,3}), P_3)$ chain;
- $(G_n^{j,4})_{n=0}^N$ is the window of opportunity, takes integer values in $S_4 = [0, 10]$ and is a $Markov(\delta(g_0^{j,4}), P_4)$ chain;
- $(G_n^{j,5})_{n=0}^N$ is the elapsed time, takes integer values in $S_5 = [0, 19]$ and is a $Markov(\delta(g_0^{j,5}), P_5)$ chain;

where $(g_0^{j,1}, g_0^{j,2}, f_0^{j,3}, f_0^{j,4}, f_0^{j,5})$ is the output of the risk assessment, as in Subsection 2.4.1. We use the notation $\mathbf{G}_n^i = (G_n^{i,1}, G_n^{i,2}, G_n^{i,3}, G_n^{i,4}, G_n^{i,5})$ for the whole random vector.

We assume the following independence property.

Proposition 3.2.1. *For all $i \in TS, j \in SC$, the defined processes are assumed to be **independent** in $(\Omega, \mathcal{F}, \mathbb{P})$.*

3.2.2 Expected values and risk aversion

In this new setup, for each $i \in TS, j \in SC$, R^i and $R^{i \leftarrow j}$ become the initial values of stochastic processes $(R_n^i)_{n=0}^N$ and $(R_n^{i \leftarrow j})_{n=0}^N$, as they are functions

of feasibility vectors, which we now modeled as random. We directly obtain expected values of these quantities, for example $\mathbb{E}\big[R_n^i\big] = \mathbb{E}\big[r(I^i, \mathbf{F}_n^i)\big]$ (we drop for a moment the superscript $i \in TS$ for simplicity, i is fixed):

$$\begin{aligned}
\mathbb{E}\big[R_n\big] = \mathbb{E}\big[r(I, \mathbf{F}_n)\big] &= \sum_{\vec{f}=(f^1,f^2,f^3,f^4,f^5)} r(I,\vec{f})\mathbb{P}(\mathbf{F}_n = \vec{f}) \\
&\overset{3.2.1}{=} \sum_{\vec{f}} r(I,\vec{f})\mathbb{P}(F_n^1 = f^1)\mathbb{P}(F_n^2 = f^2)\mathbb{P}(F_n^3 = f^3)\mathbb{P}(F_n^4 = f^4)\mathbb{P}(F_n^5 = f^5) \\
&= \sum_{\vec{f}} r(I,\vec{f})(\delta(f_0^1)P_1^n)_{f^1}(\delta(f_0^2)P_2^n)_{f^2}(\delta(f_0^3)P_3^n)_{f^3}(\delta(f_0^4)P_4^n)_{f^4}(\delta(f_0^5)P_5^n)_{f^5} \\
&= \sum_{\vec{f}} r(I,\vec{f})p^n_{f_0^1 f^1}p^n_{f_0^2 f^2}p^n_{f_0^3 f^3}p^n_{f_0^4 f^4}p^n_{f_0^5 f^5} \qquad (3.2.1)
\end{aligned}$$

where $p^n_{f_0^k f^k}$ is the element on the f_0^k-th row and f^k-th column of P_k^n. In the third line we used the n-step probability result in A.2.2; in the fourth line the definition of Dirac's δ.
Aforementioned sparsity of transition matrices luckily simplifies this computation as many of the transition probabilities in the last line are null.
Proceeding in a similar way by exploiting independence in Proposition 3.2.1, we are able compute $\mathbb{E}\big[R_n^{i\leftarrow j}\big]$. However, calculations get more complicated as $R_n^{i\leftarrow j}$ is a function of both $\mathbf{F}_n^i$ and $\mathbf{G}_n^j$, so that the number of possible realizations we need to include in the sum squares.

More importantly, for each $sca \in \mathcal{P}(SC)$ and $n \in [1, N] \cap \mathbb{N}$, we are interested in the quantities

$$\Delta Risk_n(sca), CustSat_n(sca)$$

where, in the same way as above for R^i and $R^{i\leftarrow j}$, $\Delta Risk(sca), CustSat(sca)$ from Definition 2.1 and Definition 2.3 become initial values for the stochastic process $(\Delta Risk_n(sca))_{n=0}^N$, $(CustSat_n(sca))_{n=0}^N$.
As we have introduced a non-deterministic framework, we have to establish an *attitude towards risk*, where risk is intended in the sense of the stochasticity of the quantity of interest. We have to consider that the simple expected

value is not representative enough of a distribution. We want to model a risk adverse decision maker. In economics and finance, risk aversion explains the inclination of people to prefer an outcome with low uncertainty to an outcome with high uncertainty, even if the expected return of the latter is equal to or higher than the expected return of the more certain outcome. For example, a risk adverse investor, faced with the choice of putting money into a bank account with a low but almost certain interest rate or buying a stock which could have high or null returns, prefers the first option. In our case, returns are represented by a combination of $\Delta Risk_n(sca)$ and $CustSat_n(sca)$. In this kind of setting we employ two techniques (mutually exclusive in their application) in order to enforce risk aversion:

- we model a *utility function* of the quantity of interest;
- we account for *extreme events* in the tail of the distribution of the quantity of interest.

These concepts and the related choices we make in our application deserve a little digression.

Expected utility theory

Utility functions are a tool for modeling risk attitudes. These functions are defined are only up to positive affine transformation: adding a constant or multiplying by a positive constant do not affect the conclusions we draw with our analysis.

Definition 3.1 (utility function)**.** A *utility function* of a risk adverse decision maker is a function $u : \mathbb{R}_{\geq 0} \to \mathbb{R}$ which is non decreasing and concave.

Utility functions are only non decreasing, concavity characterizes risk aversion.
In our application, we opt for the *isoelastic* utility function:

$$u(c) := \begin{cases} \frac{c^{1-\eta}}{1-\eta} & \text{for } \eta \geq 0, \eta \neq 1 \\ \log(c) & \text{for } \eta = 1. \end{cases} \tag{3.2.2}$$

Affine transformations do not affect decision making and the term -1 in the numerator is included just to establish the limit

$$\lim_{\eta \to 1} \frac{c^{1-\eta}}{1-\eta} = \log(c).$$

This class of utility functions has the following property, referred as *constant relative risk aversion*: for all $c \geq 0$

$$-\frac{c \cdot u''(c)}{u'(c)} = \eta.$$

This intuitively means that the risk aversion of the isoelastic utility takes into account the scale of c.
In the applied example of Section 3.3, we choose $\eta = \frac{1}{2}$ and $\eta = 1$ (which implies $u = \log$).

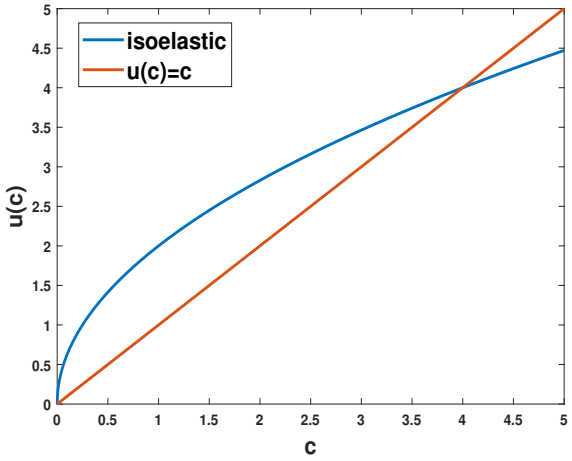


Figure 3.1: Plot of isoelastic utility for $\eta = \frac{1}{2}$

In our analysis we are then led to the following multi-objective optimization problem.

Problem 3.1 (expected utility)**.** We fix a time $n \in [1, N] \cap \mathbb{N}$ and a utility function u. We want to maximize the expected utility of the sum of risk reduction and customer satisfaction at time n while minimizing cost:

$$\max_{sca \in \mathcal{P}(SC)} \mathbb{E}\big[u(\Delta Risk_n(sca) + CustSat_n(sca))\big], \min_{sca \in \mathcal{P}(SC)} Cost(sca).$$

While we have addressed the issue of attitude towards uncertainty, this formulation has its downfalls; the most evident is that the utility u takes an input with a precise meaning (in our case the sum of risk reduction and customer satisfaction) and produces an output which could take any value in $\mathbb{R}$. We are accounting for uncertainty but we are losing in capacity of interpretation of results: expected utility can be used only to identify optimal decision, but we can not use these expected values in a quantitative cost-effectiveness analysis as we did in Subsection 2.5.2.

Evaluation of extreme events

In financial risk management, a quite old and simple idea is Roy's safety-first criterion. It is a criterion of selection of a portfolio based on minimizing the probability of the portfolio's return falling below a minimum desired threshold. With a slight variation of this idea, we can think of maximizing expected return subject to the constraint that the probability of the return falling below the threshold must be less that a certain safety level.
We mimic these techniques in our context. We consider the probability of the sum of risk reduction and customer satisfaction falling below a threshold:

$$\mathbb{P}(\Delta Risk_n(sca) + CustSat_n(sca) \leq \lambda).$$

The threshold λ must be in some way dependent of the *sca* under consideration. We choose

$$\lambda := C \cdot \mathbb{E}\big[\Delta Risk_n(sca) + CustSat_n(sca)\big]$$

where $C \in (0, 1]$ represents the kind of deviation from the expected value that the decision maker is worried about. In the applied case we set for example $C = 0.9$ which means that the decision maker considers a 10% deviation from the expected value as an undesired outcome.
We reach a formulation of the problem where we avoid setting a safety level (as suggested by the aforementioned criterion) and we just include the probability as an objective function that we want to minimize.

Problem 3.2 (evaluation of extreme events)**.** Consider the following multi objective optimization problem:

- $\max_{sca\in\mathcal{P}(SC)} \mathbb{E}\big[\Delta Risk_n(sca) + CustSat_n(sca)\big]$;
- $\min_{sca\in\mathcal{P}(SC)} Cost(sca)$;
- $\min_{sca\in\mathcal{P}(SC)} \mathbb{P}(\Delta Risk_n(sca) + CustSat_n(sca) \leq \lambda)$

The advantage of this formulation is that the expected value remains an interpretable quantity, there is no transformation as in the expected utility method. The disadvantage can be found in the fact that the conclusions drawn from this method highly depend on the parameter λ (so in our case C); the decision maker must have a good idea of what kind of deviations are considered undesirable.

3.2.3 Monte Carlo simulation and details

In both methods introduced above, we need to compute the expected values

$$\mathbb{E}\big[u(\Delta Risk_n(sca) + CustSat_n(sca))\big]$$

$$\mathbb{E}\big[\Delta Risk_n(sca) + CustSat_n(sca)\big]$$

$$\mathbb{P}(\Delta Risk_n(sca) + CustSat_n(sca) \leq \lambda) = \mathbb{E}\big[\mathbb{1}_{(\Delta Risk_n(sca)+CustSat_n(sca)\leq\lambda)}\big]$$

and this task presents some challenges. We can not simply reduce ourselves to the computation of $\mathbb{E}[R_n^i]$ and $\mathbb{E}[R_n^{i\leftarrow j}]$, because of the presence of the utility function u or simply the minimization $\min_{j\in sca}$ in Definition 2.1: any hope of exploiting linearity of the $\mathbb{E}$ operator is disrupted. An exact calculation would be more difficult than the one in (3.2.1), as there are too many underlying processes to consider that generate too many possible realizations: $\Delta Risk_n(sca)$ and $CustSat_n(sca)$ are function of the whole set of random variables $\{\mathbf{F}_n^i\}_{i\in TS}, \{\mathbf{G}_n^j\}_{j\in SC}$. Even sparsity is not really of any help.
We opt for a *Monte Carlo simulation* of the Markov chains. For each process

(there are $5 \cdot (|SC| + |TS|)$ of them), we sample trajectories from their distribution and compute $\Delta Risk_n(sca) + CustSat_n(sca)$ on those trajectories. By iterating this simulation we estimate the expected value, computing averages and invoking the strong law of large numbers . We remark how the independence assumption in Proposition 3.2.1 simplifies sampling because we only need the distributions of each Markov chain (entirely described by initial distributions and transition matrices in the definitions above, see Theorem A.2.1) and do not have to model a joint distribution.

One can generate an N-step path of a Markov chain with given transition probabilities and initial distribution as in the end of Appendix A. We put into practice this idea in MATLAB through a function that samples from the appropriate distribution the next state x of a Markov chain given its actual state x_0 and its transition matrix P. We use a simple counter to implement the sampling, following the pseudo-code in Algorithm 1. Iterating this procedure N times, we simulate an N-step path.

Algorithm 1 sampling next state from the chain distribution

```
procedure NEXTSTATE(P, x_0)
    cdf ← cumsum(P(x_0 + 1, :));        ▷ cumsum computes the CDF
    ctr ← 1;
    u ← rand;                           ▷ rand samples a number from U[0, 1]
    while u > cdf(ctr) do
        ctr ← ctr + 1;
    end while
    x ← ctr − 1;                        ▷ ctr initialized as 1, space state starts from 0
    return x
end procedure
```

With the same simulation strategy, we are able to expected values of any function involving the quantity

$$\Delta Risk_n(sca) + \alpha \cdot CustSat_n(sca),\ \alpha = \frac{\Delta Risk_n(SC)}{CustSat_n(SC)}$$

and repeat the analyses proposed in the previous Section for this choice of α.

A feature of the Monte Carlo method is that the sample size of the simulation can be chosen via the theory of confidence intervals in order to attain a given accuracy. Here we give a brief presentation of the results we need.
In probability theory the central limit theorem (CLT) establishes that, under the right assumptions, when i.i.d. random variables are averaged, their properly normalized average tends toward a normal distribution even if the original variables themselves are not normally distributed. More formally:

Theorem 3.2.2 (Central limit)**.** *If $(X_n)_{n\in\mathbb{N}}$ is a sequence of i.i.d. random variables in $L^2(\Omega,\mathbb{P})$ with $\mathbb{E}\big[X_i\big] = \mu$ and $Var\big[X_i\big] = \sigma^2$, then as $n \to \infty$*

$$\frac{M_n - \mu}{\frac{\sigma}{\sqrt{n}}} \xrightarrow{d} Z \sim \mathcal{N}(0,1)$$

where $M_n := \frac{1}{n}\sum_{k=1}^{n} X_k$

Proof. It follows directly from Lévy's continuity theorem and analysis of the characteristic function. We refer to [11] for all the details of the proof. □

The central limit theorem can be formulated in this way:

$$M_n \simeq \mu + \frac{\sigma}{\sqrt{n}} Z \sim \mathcal{N}\big(\mu, \frac{\sigma^2}{n}\big), \text{ for } n \gg 1$$

as a detail of the convergence result in the law of the large numbers ($\simeq$ means that the two variables have approximately the same distribution).
An average of the kind of M_n naturally appears in the Monte Carlo method, as the approximation is obtained by averaging all simulations. Under the hypotheses of central limit theorem, we have that

$$\mathbb{P}(|M_n - \mu| \leq \lambda \frac{\sigma}{\sqrt{n}}) = \mathbb{P}(|Z| \leq \lambda) = 2F(\lambda) - 1$$

where F is the CDF of the standard normal distribution

$$F(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy.$$

If we want to approximate μ with M_n, we fix a confidence of $p = 99\%$ (for example) and we get $\lambda = F^{-1}(\frac{p+1}{2}) \approx 2.57$ so that

$$\mathbb{P}(|M_n - \mu| \leq 2.57\frac{\sigma}{\sqrt{n}}) \simeq 99\%.$$

$r_{99} := 2.57\frac{\sigma}{\sqrt{n}}$ is the radius of the 99% confidence interval for μ: it means that the unknown expected value μ belongs to $[M_n - 2.57\frac{\sigma}{\sqrt{n}}, M_n + 2.57\frac{\sigma}{\sqrt{n}}]$ with 99% probability. It is important to notice that in this computation we cannot use true variance σ^2 and true mean μ (which defines σ^2 and is the unknown). We have to resort to the approximation given by the (corrected) sample standard deviation

$$\overline{\sigma}_n = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(X_k - M_n)^2}.$$

In the end it is possible to choose the sample size n in order to shrink the estimated numerical error r_{99} towards the desired accuracy. In some cases, we also monitor the relative numerical error

$$\frac{r_{99}}{|\sum_{k=1}^{n} X_k|}.$$

3.3 Example of application: dynamic version

We take the inputs from the static application (Subsection 2.5.1) as initial states of our simulation.

Among the five fields that define feasibility, we assume that specialist expertise and window of opportunity are deterministic, because they are in a certain sense an intrinsic feature of an attack method. Elapsed time, knowledge of the system under investigation and equipment are the non-deterministic quantities. This assumption can be debated, but if one has another opinion it is easy to make a change within such a flexible framework. Following the scale in [8, Table 5] we assume that these non-deterministic quantities are decreasing, so that the sum of the feasibility components decreases and risk

increases in time.

In the example we consider a horizon of 15 years and a transition time of 5 years. Translating to the formalism introduced in the previous Section we have $n \in \{0, 1, 2, 3\}$, so that $n = 0$ is initial time, $n = 1$ is 5 years, $n = 2$ is 10 years, $n = 3$ is 15 years.

Here is an example of a transition matrix (in this case the matrix P_4 for the equipment field). We remark the diagonal block, lower triangular structure, whose meaning has already been explained.

$$((p_4)_{h,l})_{h,l=0}^{9} = \left(\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & & & & & \\ 0.2 & 0.8 & 0 & 0 & 0 & & & & & \\ 0.1 & 0.3 & 0.6 & 0 & 0 & & & \mathbf{0} & & \\ 0.05 & 0.15 & 0.25 & 0.55 & 0 & & & & & \\ 0.05 & 0.1 & 0.15 & 0.2 & 0.5 & & & & & \\ \hline & & & & & 1 & 0 & 0 & 0 & 0 \\ & & & & & 0.3 & 0.7 & 0 & 0 & 0 \\ & & \mathbf{0} & & & 0.2 & 0.2 & 0.6 & 0 & 0 \\ & & & & & 0 & 0 & 0.2 & 0.8 & 0 \\ & & & & & 0 & 0 & 0.05 & 0.15 & 0.8 \end{array}\right)$$

Transition matrices for knowledge of the item and elapsed time have the same structure, we do not report them here. We notice the diagonal dominance and the increasing probabilities along the rows: that is intended to model a process which tends to stay on its current state and if it moves from it has a higher probability of moving towards near lower states.

We now show results obtained through simulation of the Markov chains (see Subsection 3.2.3) in the 10 years projection ($n = 2$) adopting customized risk function and with particular choices for the parameters of Problem 3.1 ($\eta = \frac{1}{2}$) and Problem 3.2 ($C = 0.9$). Similar conclusions can be drawn for other choices of time (in particular for $n = 3$) and parameters ($\eta = 1 \implies u = \log$ for example); we just want to highlight an example of the kind of comments and insights that can be developed within this framework, avoiding too many repetitions.

For this application, we keep results only for sca such that $|sca| \geq 4$ and for $sca = \{2, 3, 5\}$ which is composed of the security controls most requested by customers.

The number of simulations has been chosen following the theory of confidence intervals (see Subsection 3.2.3). If we denote with $X(sca)$ the random variable whose expected value we want to approximate and with $r_{99}[X(sca)]$ the radius of the 99% confidence interval for that expected value, the number of simulations has been set in order to shrink the maximum absolute and relative radius among all sca considered:

$$R_{99}^{abs}[X] := \max_{\{sca:|sca|\geq 4\}\cup\{2,3,5\}} r_{99}[X(sca)]$$

$$R_{99}^{rel}[X] := \max_{\{sca:|sca|\geq 4\}\cup\{2,3,5\}} \frac{r_{99}[X(sca)]}{|\mathbb{E}[X(sca)]|};$$

the accuracy obtained with 2000 iterations has been deemed to be good enough and in the next sections we discuss all the details.

Results obtained with ISO/SAE 21434 risk function go in the same direction and we do not report them here.

3.3.1 Remarks on expected utility

We report results for Problem 3.1 with $n = 2, \eta = \frac{1}{2}$: in Figure 3.2 α is set to 1, in Figure 3.3 we set α as the random variable $\frac{\Delta Risk_2(SC)}{CustSat_2(SC)}$.

Overall, the scenario is evidently different from the one in the static version. $\{2, 3, 4, 5, 6\}$ is stably present on the Pareto frontier together with $\{2, 3, 4, 5\}$. This is more evident when $\alpha = \frac{\Delta Risk_2(SC)}{CustSat_2(SC)}$. We can not go deeper in this analysis because, as explained before, the presence of the transformation u does not allow any quantitative analysis and expected utility method can be used only to identify a preference order among security control alternatives. For compactness, we reintroduce the notation

$$\Sigma_2^\alpha(sca) := \Delta Risk_2(sca) + \alpha \cdot CustSat_2(sca).$$

In Table 3.1, we observe the accuracy in the approximation of $\mathbb{E}[u(\Sigma_2^\alpha(sca))]$ obtained after 2000 simulation samples, in terms of radius and relative radius

of the confidence. This type of accuracy certifies that computational error does not affect the insights provided above.

	$\alpha = 1$	$\alpha = \frac{\Delta Risk_2(SC)}{CustSat_2(SC)}$
$R_{99}^{abs}[u(\Sigma_2^\alpha)]$	0.0157	0.0170
$R_{99}^{rel}[u(\Sigma_2^\alpha)]$	0.0068	0.0067

Table 3.1: 99% confidence intervals for expected utility with 2000 simulations

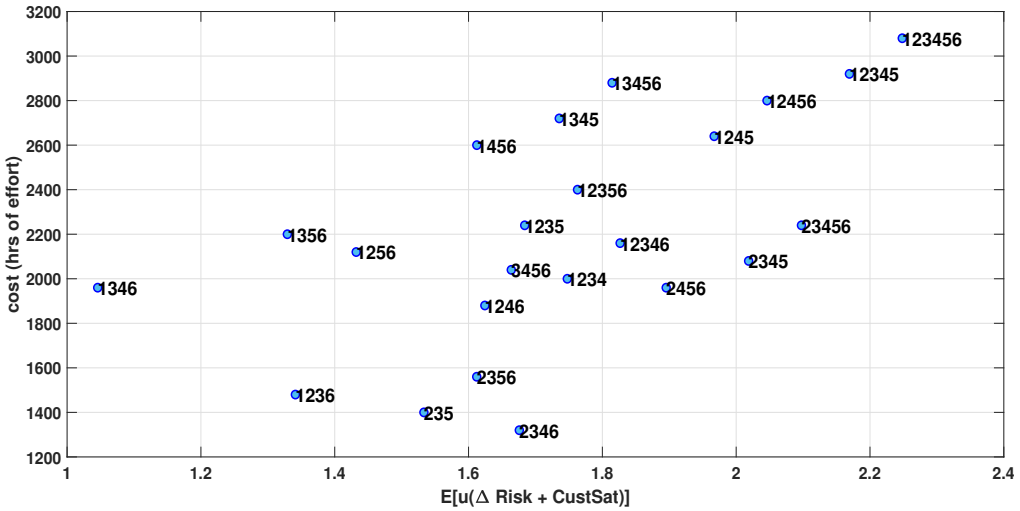


Figure 3.2: Results for expected utility: 10 years projection, customized risk function, $\alpha = 1$, $\eta = \frac{1}{2}$

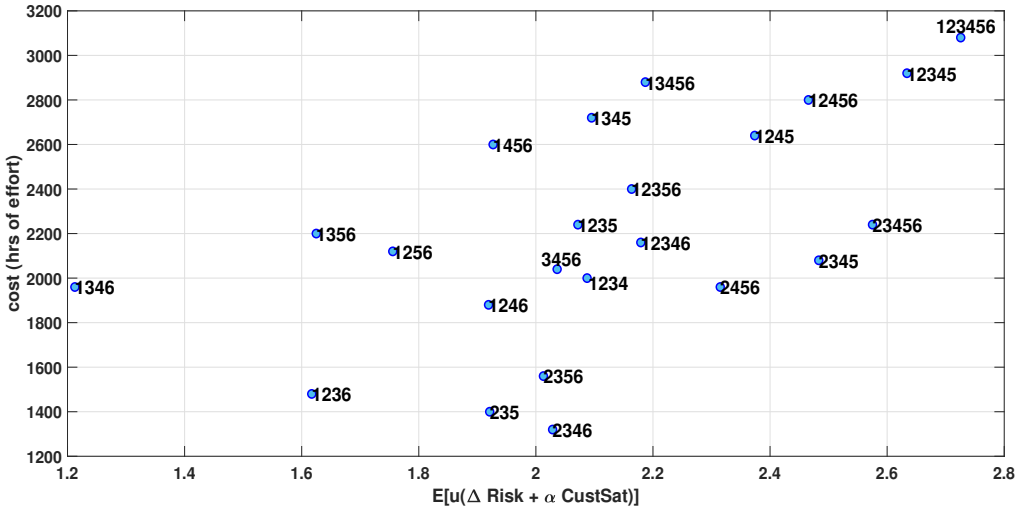


Figure 3.3: Results for expected utility: 10 years projection, customized risk function, $\alpha = \frac{\Delta Risk_2(SC)}{CustSat_2(SC)}$, $\eta = \frac{1}{2}$

3.3.2 Remarks on evaluation of extreme events

We report results for Problem 3.2 with $n = 2, C = 0.9$: in Figure 3.4 α is set to 1, in Figure 3.5 we set α as the random variable $\frac{\Delta Risk_2(SC)}{CustSat_2(SC)}$. We used a gradient of colour in order to visualize $\mathbb{P}(\Sigma_2^\alpha(sca)) \leq \lambda)$, where $\lambda = C \cdot \mathbb{E}[\Sigma_2^\alpha(sca)]$: from blue (10% of going below threshold) to yellow (over 30%).

The choice $C = 0.9$ is motivated by decision maker preference: for $C = 0.75$ the probabilities tend to become trivial (almost no outcome is under the threshold) and $C > 1$ does not really make sense in representing 'extreme events'; $C = 0.9$ seems like a good compromise.

With regard to the error in the approximation of expected values, we need to make a distinction. Looking at Table 3.2, the maximum radius of the 99% confidence intervals, both absolute and relative, for the expected value of $\Sigma_2^\alpha(sca)$ show that 2000 simulation samples ensure a good approximation. In contrast, as it is possible to see in Table 3.3, the error in estimating $\mathbb{P}(\Sigma_2^\alpha(sca)) \leq \lambda)$ is more impactful: a spread of almost $\pm 3\%$ in a quantity that ranges from 10% to 35% can not be ignored. As we would need a huge

number of iterations to shrink these values (about 10000 to achieve a $\pm 1\%$ spread), we opt for just accounting for this error and being cautious in our analysis when this probability is involved.

Again, even with the measure of the simple expected value without utility, $\{2, 3, 4, 5, 6\}$ is present on the Pareto frontier. We notice that $\{2, 3, 4, 5\}$ achieves a much better performance than $\{2, 3, 5, 6\}$ in terms of Σ_2^α. These facts strengthen in a future projection what we already said in Section 2.5. In Table 3.4, we compare the three *sca* composed of 5 security controls that have the best Σ_2^α. We argue that $\{2, 3, 4, 5, 6\}$ is the most cost-effective: only $\{1, 2, 3, 4, 5\}$ has a better Σ_2^α, but it also costs about 700 hours of effort more; $\{2, 3, 4, 5, 6\}$ furthermore proves to always be among the *sca* with small $\mathbb{P}(\Sigma_2^\alpha(sca)) \leq \lambda)$, with for example a mere 8.75% of going under the threshold when $\alpha = \frac{\Delta Risk_2(SC)}{CustSat_2(SC)}$.

	$\alpha = 1$	$\alpha = \frac{\Delta Risk_2(SC)}{CustSat_2(SC)}$
$R_{99}^{abs}[\Sigma_2^\alpha]$	0.0182	0.0217
$R_{99}^{rel}[\Sigma_2^\alpha]$	0.0136	0.0134

Table 3.2: 99% confidence intervals for $\mathbb{E}[\Sigma_2^\alpha(sca))]$ with 2000 simulations

	$\alpha = 1$	$\alpha = \frac{\Delta Risk_2(SC)}{CustSat_2(SC)}$
$R_{99}^{abs}[\mathbb{1}_{\Sigma_2^\alpha \leq \lambda}]$	0.0270	0.0268

Table 3.3: 99% confidence intervals for $\mathbb{P}(\Sigma_2^\alpha(sca)) \leq \lambda)$ with 2000 simulations, $\lambda = 0.9 \cdot \mathbb{E}[\Sigma_2^\alpha(sca)]$

	$\mathbb{P}(\Sigma_2^\alpha(sca)) \leq \lambda)$		$\mathbb{E}[\Sigma_2^\alpha(sca)]$	
	$\alpha = 1$	$\alpha = \frac{\Delta Risk_2(SC)}{CustSat_2(SC)}$	$\alpha = 1$	$\alpha = \frac{\Delta Risk_2(SC)}{CustSat_2(SC)}$
$sca = \{1,2,3,4,5\}$	14.7%	10.5%	2.169	2.634
$sca = \{1,2,4,5,6\}$	18.4%	16.7%	2.046	2.466
$sca = \{2,3,4,5,6\}$	14.9%	8.75%	2.098	2.575

Table 3.4: $\mathbb{P}(\Sigma_2^\alpha(sca)) \leq \lambda)$ and $\mathbb{E}[\Sigma_2^\alpha(sca)]$ for some interesting sca, $\lambda = 0.9 \cdot \mathbb{E}[\Sigma_2^\alpha(sca)]$

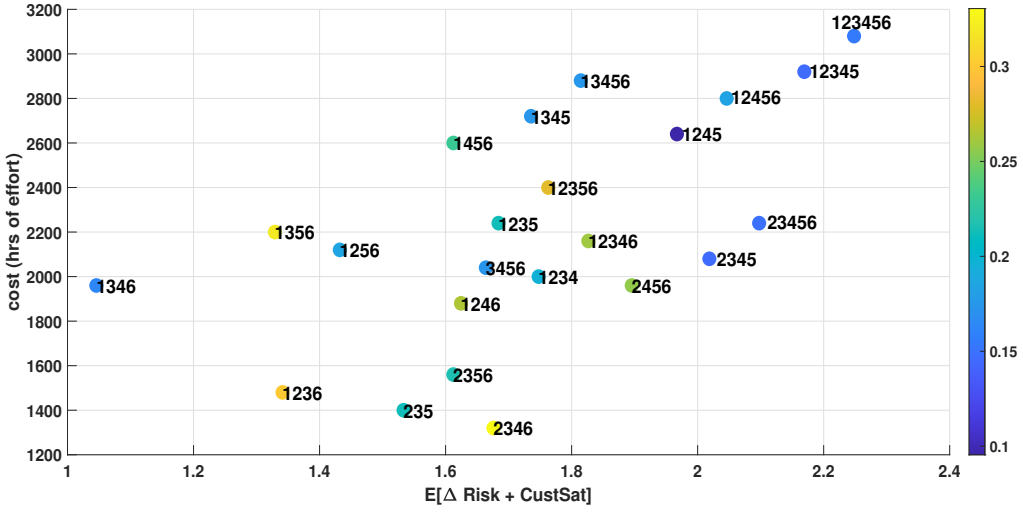


Figure 3.4: Results for extreme events analysis: 10 years projection, customized risk function, $\alpha = 1$, $C = 0.9$

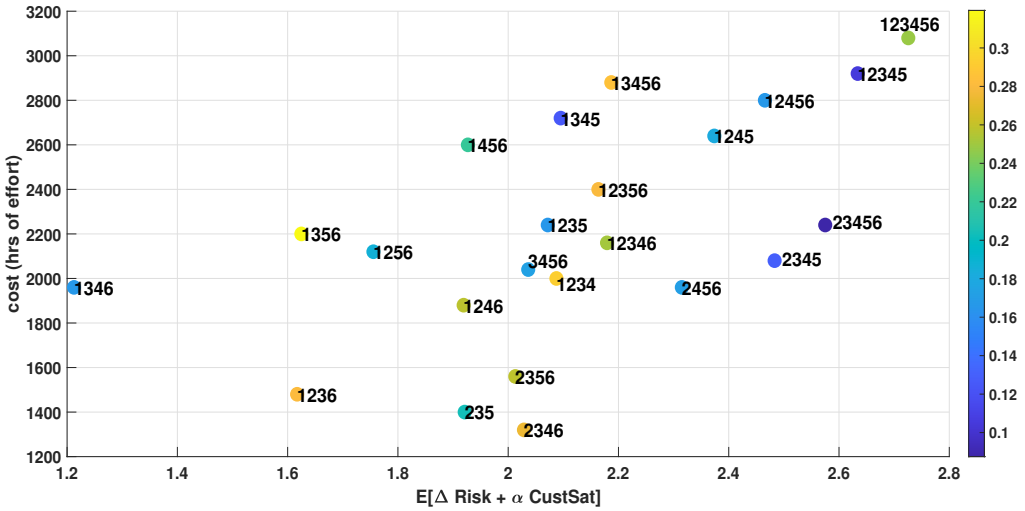


Figure 3.5: Results for extreme events analysis: 10 years projection, customized risk function, $\alpha = \frac{\Delta Risk_2(SC)}{CustSat_2(SC)}$, $C = 0.9$

Conclusions

At the end of this work, we can draw some final insights about what has been done and what can be done in the future.
We have formalized the problem of decision making in cybersecurity and built a framework for the evaluation of possible courses of action. The tool can be useful in different moments of the product development process: from concept phase, where the security controls to be implemented are selected according to the results of risk analysis; through vulnerability analysis during design and testing, where further solutions may be added in order to fulfil the defined cybersecurity specifications; to the post-production phase, in the renegotiation of the maintenance contracts with suppliers for example. Throughout the thesis, we applied all this theory in a particular instance placed in the concept phase, a small instance with 9 threat scenarios and 6 security controls; in this situation, our tool was mainly valuable in the direction of validating some intuitions that the decision maker already had on that instance. In a larger instance in terms of threat scenarios (large $|TS|$) or in terms of security controls (large $|SC|$) or in both, we think that with some tweaks the framework remains effectual and crucially it could be used also for the identification of the right decision. In the case of large $|SC|$, however, we would have to address two needs:

- a more efficient calculation of the objective function $\Delta Risk(sca)$ for all sca, as at the moment $\mathcal{O}(2^{|SC|})$ operations are required;
- the employment of parallel computing in the simulation of Markov chains, to avoid long running times due to the computational burden.

This kind of instance is not unrealistic, considering that one could define the set of security controls SC with more and more refinement, from abstract macro categories of security controls like in our example to concrete specific requirements, and this would cause the number of security controls $|SC|$ to grow considerably.

On the stochastic modelling of the risk evolution, in the end, it is evident that there is still a lot of room for improvement. Markov chains have been regarded as the right tool also considering the available level of evidence on the phenomenon. It is a quite basic and off-the-shelf model, but the degree of sophistication can be increased ad libitum to fit new evidence, only if and when new evidence arrives.

Appendix A

Discrete-time Markov chains

In this chapter, we give some basic definitions, notations and results on discrete-time Markov chains, a tool we applied in modeling the evolution of risk. We do not develop a detailed treatment of this subject, which is infinitely vast, but we present the simple material that we strictly need in Chapter 3.

The references for this chapter are [12, ch. 2] and [13, ch. 1].

A.1 Markov property

Sequences of independent and identically distributed random variables are not always interesting as stochastic models because they have a predictable and simple behavior. In order to introduce more variability, one can allow for some probabilistic dependence on the past. A limited amount of memory (only up to the previous state) suffices to produce a great diversity of behaviors.

Definition A.1. A sequence $(X_n)_{n\in\mathbb{N}}$ of random variables on $(\Omega, \mathcal{F}, \mathbb{P})$ with values in a set E is called a *discrete-time stochastic process* with *state space* E. We assume the state space to be countable and denote its elements with i, j, k. When $X_n = i$ the process is said to *visit state* i *at time* n.

Definition A.2. Let $(X_n)_{n\in\mathbb{N}}$ be a discrete-time stochastic process with state space E. If for all integers $n \geq 0$ and all states $i_0, i_1, \ldots, i_{n-1}, i, j$ such that

$$\mathbb{P}(X_n = i, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) > 0$$

we have that

$$\mathbb{P}(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) = \mathbb{P}(X_{n+1} = j | X_n = i)$$

$$\mathbb{P}(X_{n+1} = j | X_n = i) = p_{ij}$$

the stochastic process is called a homogeneous Markov chain.

First equation is the Markov property, which is a loss of memory property that indicates a dependence on the past only through the previous state. The second equation is the time-homogeneity condition, which states that transitions are independent of time n. The term 'homogeneous' is often dropped and we simply say Markov chain.

Definition A.3. The matrix $P = (p_{ij})_{i,j\in E}$ where

$$p_{ij} = \mathbb{P}(X_{n+1} = j | X_n = i)$$

is the *transition matrix* of the Markov chain.

Because its entries are probabilities, the transition matrix is a stochastic matrix in the sense that for all $i, j \in E$

$$p_{ij} \geq 0, \sum_{k\in E} p_{ik} = 1.$$

A.2 Distribution of a Markov chain

The random variable X_0 is called the initial state and its probability distribution $\lambda \in [0, 1]^n$ (conventionally a row vector)

$$\lambda(i) := \mathbb{P}(X_0 = i), i \in E$$

is the *intial distribution*. From the chain rule we have

$$\begin{aligned}&\mathbb{P}(X_0 = i_0, X_1 = i_1, \ldots, X_k = i_k)\\&= \mathbb{P}(X_0 = i_0)\mathbb{P}(X_1 = i_1|X_0 = i_0)\cdots\mathbb{P}(X_k = i_k|X_{k-1} = i_{k-1}, \ldots, X_0 = i_0)\end{aligned} \tag{A.2.1}$$

With the Markov property in Definition A.2, we obtain the *distribution* of the Markov chain

$$\mathbb{P}(X_0 = i_0, X_1 = i_1, \ldots, X_k = i_k) = \lambda(i_0)p_{i_0 i_1}\cdots p_{i_{k-1}i_k}$$

and we get the following result.

Theorem A.2.1. *The distribution of a Markov chain is determined by its initial distribution and its transition matrix.*

We denote a Markov chain with initial distribution λ and transition matrix P with $Markov(\lambda, P)$. We discuss the existence and the construction of a Markov chain given an initial distribution and a transition matrix in the next Section.
Many probabilities for the Markov chain can be expressed conveniently in terms of the transition matrix P and its n-th product $P^n, n \geq 0$. Let p_{ij}^n denote the (i, j)-th entry of P^n. By the definition of matrix multiplication and the law of total probability

$$p_{ij}^n = \sum_{i_1,\ldots,i_{n-1}\in E^{n-1}} p_{ii_1}p_{i_1i_2}\ldots p_{i_{n-1}j} = \mathbb{P}(X_n = j|X_0 = i);$$

$$\mathbb{P}(X_n = j) = (\lambda P^n)_j. \tag{A.2.2}$$

The property of multiplication of matrices $P^{m+n} = P^m P^n$ yields the Champman-Kolmogorov equation

$$p_{ij}^{m+n} = \sum_{k\in E} p_{ik}^m p_{kj}^n,$$

which states that the probability the chain moves from i to j in $m + n$ steps is equal to the probability that it moves from i to any $k \in E$ in m steps, and then it moves from k to j in n more steps.

A.3 Markov recurrences

Many Markov chains receive a natural description in terms of a recurrence equation driven by white noise.

Proposition A.3.1. *Let $(X_n)_{n\in\mathbb{N}}$ be a stochastic process with state space E of the form*

$$X_{n+1} = f(X_n, Y_{n+1})$$

where $(Y_n)_{n\in\mathbb{N}}$ are i.i.d. random variables on a general space E' that are independent of X_0 and $f : E \times E' \to E$.
Then X_n is a Markov chain with transition probabilities $p_{ij} = \mathbb{P}(f(i, Y_1) = j)$

Proof. By the recursive definition of the process we have that

$$\mathbb{P}(X_{n+1} = i|X_n = i, X_{n-1}, \dots, X_0) = \mathbb{P}(f(i, Y_{n+1}) = j|X_n = i, X_{n-1}, \dots, X_0).$$

Then Y_{n+1} is independent of $(X_0, \dots, X_n)$ because this vector is a function of $(X_0, Y_1, \dots, Y_n)$ through f, which implies:

$$\mathbb{P}(f(i, Y_{n+1}) = j|X_n = i, X_{n-1}, \dots, X_0) = \mathbb{P}(f(i, Y_{n+1}) = j).$$

In the end, Y_1 and Y_{n+1} have the same distribution:

$$\mathbb{P}(f(i, Y_{n+1}) = j) = \mathbb{P}(f(i, Y_1) = j) = p_{ij}.$$

□

This result is useful for identifying stochastic processes that are Markov chains. We now establish that any Markov chain can be constructed as in Proposition A.3.1. We need this fact

Remark 1 (Uniform representation of a random variable). Let λ be a probability measure on $E = \{0, 1, \dots\}$, let U be a random variable uniformly distributed on $[0, 1]$ and $X = h(U)$ where

$$h(u) = j \text{ if } u \in I_j \text{ for some } j \in E$$

and $I_j = [\sum_{k=0}^{j-1} \lambda_k, \sum_{k=0}^{j} \lambda_k)$. Then

$$\mathbb{P}(X = j) = \mathbb{P}(h(U) = j) = \mathbb{P}(U \in I_j) = \lambda_j$$

Theorem A.3.2 (Construction of Markov chains)**.** *Let $P = (p_{ij})_{i,j\in E}$ be a stochastic matrix and λ a probability measure on $E = \{0, 1, \dots\}$. Let $U_0, U_1, \dots$ i.i.d. random variables, $U_i \simeq U[0,1]$. Define $X_0 = h(U_0)$ (where h is as in Remark 1) and $X_{n+1} = f(X_n, U_{n+1})$ where for each $i \in E$*

$$f(i,u) = j \text{ if } u \in I_j \text{ for some } j \in E$$

and $I_{ij} = [\sum_{k=0}^{j-1} p_{ik}, \sum_{k=0}^{j} p_{ik})$. Then $(X_n)_{n\in\mathbb{N}}$ is a Markov chain with initial distribution λ and transition matrix P.

Proof. By Remark 1, X_0 has distribution λ. By Proposition A.3.1, $(X_n)_{n\in\mathbb{N}}$ is a Markov chain with transition probabilities

$$\mathbb{P}(f(i,U_1) = j) = \mathbb{P}(U_1 \in I_{ij}) = p_{ij}.$$

□

An immediate consequence is that there exists a Markov chain associated with any stochastic matrix.

This artificial representation is useful for simulating small Markov chains and we will exploit it in the Monte Carlo method, presented in Section 3.2 and applied in the dynamic use case in Section 3.3. The procedure is contained in Algorithm 2 below.

Algorithm 1 in Section 3.2 is an implementation of the artificial recurrence function f with a counter, adapted to our use case. In the application, the initial state of the chain is deterministic (initial distribution is a Dirac's δ) so we do not need the sample u_0 and the function h.

Algorithm 2 N-path sampling

procedure N-PATHSAMPLER(f, h) ▷ f, h defined above

$u_o, \ldots, u_N \leftarrow rand(N+1)$ ▷ $rand(N+1)$ samples from $U([0,1]^{N+1})$

$i_0 \leftarrow h(u_0)$

$m \leftarrow 1$

while $m < N+1$ **do**

$i_m \leftarrow f(i_{m-1}, u_m)$

$m \leftarrow m+1$

end while

return $i_0, \ldots, i_N$

end procedure

Bibliography

[1] Ring, Martin and Dürrwang, Jürgen and Sommer, Florian and Kriesten, Reiner, *Survey on vehicular attacks - building a vulnerability database*, 208-212, 10.1109/ICVES.2015.7396919, 2015

[2] Prakash Kadhirvelan, Sathya and Söderberg-Rivkin, Andrew, *Threat Modelling and Risk Assessment Within Vehicular Systems*, August 2014.

[3] Miller, Charlie and Valasek, Chris, *Securing Self-Driving Cars (one company at a time)*, http://illmatics.com/carhacking.html, 2018.

[4] Checkoway, Stephen et al., *Experimental Security Analysis of a Modern Automobile*, 2010.

[5] Checkoway et al., *Comprehensive Experimental Analyses of Automotive Attack Surfaces*, 2011.

[6] UNECE, *UN Regulation No. 155 - Cyber security and cyber security management system*, https://unece.org/transport/documents/2021/03/standards/un-regulation-no-155-cyber-security-and-cyber-security, 2021.

[7] ISO (International Organization for Standardization)/ SAE International, *ISO/SAE 21434, Road vehicles — Cybersecurity engineering*, https://www.iso.org/standard/70918.html, 2021.

[8] E-safety vehicle intrusion protected applications, *Deliverable D2.3: Security requirements for automotive on-board networks based on dark-side scenarios*, 2009.

[9] VDA QMC Working Group 13 / Automotive SIG, *Automotive SPICE Process Assessment / Reference Model*, 2017.

[10] SAE International, *J3061, Cybersecurity Guidebook for Cyber-Physical Vehicle Systems*, 2016.

[11] Pascucci, Andrea, *Teoria della Probabilità: Variabili aleatorie e distribuzioni*, Springer, 2020.

[12] Brémaud,Pierre, *Markov Chains: Gibbs Fields, Monte Carlo Simulation and Queues*, Second Edition, Springer, 2020.

[13] Serfozo, Richard, *Basics of Applied Stochastic Processes*, Springer, 2009.