UNIONE EUROPEA
Fondo Sociale Europeo

Ministero dell'Università
e della Ricerca

PON
RICERCA
E INNOVAZIONE
2014 - 2020

REACT EU

La borsa di dottorato è stata cofinanziata con risorse del
Programma Operativo Nazionale Ricerca e Innovazione 2014-2020, risorse FSE REACT-EU
Azione IV.4 “Dottorati e contratti di ricerca su tematiche dell’innovazione”
e Azione IV.5 “Dottorati su tematiche Green”

Impact Measures for Business Process Compliance: Analyzing Impact Compliance and Declarative Specifications

Tewabe Chekole Workneh

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Supervisor:

Prof. Matteo Cristani (Università degli Studi di Verona, Italy)

Co-Supervisor:

Prof. Pietro Sala (Università degli Studi di Verona, Italy)

Sept 2024

UNIVERSITÀ DEGLI STUDI DI VERONA
DEPARTMENT OF COMPUTER SCIENCE
GRADUATE SCHOOL OF SCIENCES ENGINEERING MEDICINE
DOCTORAL PROGRAM IN COMPUTER SCIENCE

WITH THE FINANCIAL CONTRIBUTION OF
National Operational Programme for Research and Innovation 2014-2020

Cycle XXXVII Year 2024

TITLE OF THE DOCTORAL THESIS

Impact Measures for Business Process Compliance: Analyzing Impact Compliance and Declarative Specifications

S.S.D INF/01

SUPERVISOR:
Prof. Matteo Cristani, Università degli Studi di Verona, Italy

CO-SUPERVISOR:
Prof. Pietro Sala, Università degli Studi di Verona, Italy

COORDINATOR:
Prof. Ferdinando Cicalese, Università degli Studi di Verona, Italy

Doctoral Student: Tewabe Chekole Workneh

Signature: ____

To Tata, Kal, and Abigu. To Jesus and Mary.
(I've never seen the righteous forsaken or their seed begging for bread.)

Keywords

Business Process, Business Process Compliance, Impact compliance, monotonic Reasoning, Graph Theory

Abstract

A *business* is an organization that engages in *commercial*, *industrial*, or *professional* activity on a systematic and predefined manner. Key aspects of the business include *profit motive*, *exchange of goods and services*, *organization and management*, and *risk*.

A business's objective is typically to supply *goods or services* to consumers or to other businesses in order to make a profit or to achieve a given non-profit purpose. Businesses can range from *small-scale enterprises* like neighborhood shops to *massive multinational corporations* but they are distinguished from occasional or unorganized endeavors for their scope, that is to organize their own activities on a specified way, in many cases formally defined, *a priori* with respect to the activities. In other terms, businesses have their own predefined *processes*, or *workflows*.

A process is a set of *events*, *procedures*, or *operations* that are performed in admissible systematic orders to reach a specified conclusion or purpose. In industrial and agrictulural production, for instance, a process is the activity of converting *inputs* (such as raw materials, data, or resources) into *outputs* (finished goods, results, or solutions) using a set of structured procedures. Key elements of a process include *sequence*, *inputs and outputs*, *transformation*, *repetition*, and *control*.

A business process is a comprehensive formal model that represents the behavior of ***complex systems***, both existing and newly designed ones. These models capture the interactions and synchronizations between humans and machines, making them crucial for understanding and optimizing the ***dynamic interplay*** within business operations.

Compliance refers to adherence to laws, regulations, guidelines, and specifications relevant to organization operations. It ensures that businesses operate within a given

normative background and meet the standards set by regulatory bodies, industry best practices, and internal policies. Ensuring compliance means that business processes are aligned with regulatory requirements, organizational goals, and both soft and hard rules imposed on the processes themselves. Compliance is essential for maintaining legal and ethical integrity, minimizing risks, and enhancing the reputation of the organization. Without compliance warranties, a business may incur penalties from regulatory bodies or have adverse effects in the future, whether economically, socially, or environmentally. Business processes must adhere to various forms of compliance, in the realm of practical business life including:

a ***General Regulatory Compliance***: This ensures business processes conform to external authorities' laws, regulations, and standards. It involves adhering to legal requirements that govern the industry or sector in which the organization operates.

b ***Goal Compliance***: This aligns business processes with predefined organizational goals and objectives. It ensures that the processes support the strategic aims and targets of the organization.

c ***Impact Compliance***: This ensures that the impacts of business processes remain within acceptable thresholds, thereby minimizing negative effects of the execution of the process. It involves assessing and managing the potential effects of processes on various aspects of the business.

d ***Environmental Compliance***: This ensures that the impacts of business processes do not exceed constraints set from an environmental perspective. It involves adherence to environmental regulations and practices that mitigate adverse environmental effects. Adhering to these compliance requirements helps ensure that business processes operate effectively within legal and ethical boundaries while supporting organizational objectives and minimizing negative impacts on society and the environment.

This dissertation primarily focuses on ***impact compliance*** and ***environmental compliance***, that I shall show are analogous in technical terms. The purpose, on a generic viewpoint, is to minimize the emission of Carbon dioxide (CO_2) but can also be used to regulate other pollutant substances. It delves into the definition of impact compliance, its integration with existing compliance concepts, and its evaluation from multiple perspectives.

When we discuss Compliance, there are three perspectives of compliance i.e. corrective, detective, and preventive. All these three perspective base themselves on the detective dimension, for no correction or prevention can be performed without leaning out the detection of (potential) uncompliances.

Corrective measures are intended to limit the extent of any consequences caused by non-compliant situations. They are reactive and aim to address and mitigate issues after they have occurred. Examples include ***manual audits***, which are periodic reviews conducted by individuals to identify and rectify compliance issues, and ***automated detections***, which are systems and tools that automatically detect deviations from compliance standards.

Detective measures aim to identify non-compliant situations "after-the-fact". They are essential for recognizing and addressing issues that have already impacted the business process. These measures also include manual audits and automated detections, similar to corrective measures.

Preventive measures embed compliance into the business processes from the outset, a concept known as ***"Compliance by Design"***. The objective is to proactively prevent non-compliance by designing processes that inherently adhere to compliance requirements.

The dissertation explores the synthesis of business processes through ***declarative specifications***, which provide a structured framework for defining and enforcing compliance. It investigates how ***impact compliance*** can be effectively integrated with regulatory and goal compliance, ensuring a holistic approach to business process management.

Business Process Compliance (BPC) is a collection of methodologies used to evaluate business processes to ensure they adhere to specified constraints. These constraints, which may be imposed due to regulatory requirements or organizational goals, are essential for maintaining the integrity and reliability of business operations. *Business Process*

Compliance (BPC) methods assess whether an execution trace exists—a sequence of actions within the process—that violates any of the imposed constraints. Conversely, this evaluation can also involve superimposing a set of constraints on the process and then assessing all possible executions for compliance. This dual approach to *Business Process Compliance (BPC)* is highly relevant in real-world applications, particularly in the realm of regulatory compliance. Businesses must verify their processes against a normative framework, which includes not only stringent regulations but also softer guidelines, product specifications, and standards. This verification ensures that processes meet the expectations set by regulatory bodies and the business owners themselves.

In this dissertation, I introduce a novel type of compliance, termed ***impact compliance***. This concept is designed to evaluate whether a business process adheres to a set of constraints by ensuring that the undesirable effects of executing tasks within the process are maintained below specified limits. *Impact compliance* is showed to be usable for minimizing negative outcomes associated with business operations such as for regulating the emission of pollutant substances including carbon dioxide(CO2), thereby optimizing overall process performance. I demonstrate that under certain structural conditions, the problems associated with checking for compliance are polynomially solvable on deterministic machines. Specifically, the task of determining whether any execution trace violates the constraints is generally NP-complete. However, verifying that all possible executions adhere to the constraints is polynomially solvable, provided the structural conditions are met.

This dissertation also delves into the synthesis of business processes through declarative specifications and examines the significant role of compliance, discussing *impact compliance by design*. I also design a method to compare business processes as a whole in terms of impact, and therefore obtain a notion of *impact similarity* that is in turn shown to be useful for process mining.

By introducing and detailing the concept of impact compliance, I provide a framework for ensuring that business processes not only comply with regulatory and organizational standards but also minimize undesired effects, thereby enhancing operational efficiency

and time Complexity. This research contributes to the field by offering a detailed exploration of impact compliance, proposing innovative methods for integrating impact compliance into business process models, and demonstrating the application of these methods through various case studies and theoretical analyses.

Statement of Originality

I hereby declare that the present thesis is my original work for the completion of a doctoral degree at the **Università degli Studi di Verona**. It has not been previously submitted to any other institution for a degree, diploma, or qualification. I have carefully reviewed the University's policy on research ethics and accept full responsibility for complying with the procedures set forth by the University's Research Ethics Committee (REC). I have thoroughly assessed any potential risks associated with this research and, if applicable, have obtained the necessary ethical and/or safety approval. I acknowledge my responsibilities towards the participants and respect their rights. .

Signature: ____________________ Sept 2024

Acknowledgments

First and foremost, I would like to express my deepest gratitude to the Almighty for His boundless mercy and for granting me the opportunity to embark on and complete this academic journey. I also extend my heartfelt thanks to my wife, **Kalidan Zerihun Berhe**, for her unwavering love and support throughout this process, and to my beloved daughter, **Abigail Tewabe Chekole**, for her positivity and encouragement.

I am profoundly grateful to my family: my mother, **Ethalmahu Mengash**, my sister, **Tenaye Chekole Workneh** (whose steadfast support I will never forget), my twin brother, **Temesegen Chekole Workneh**, and my elder sister, **Serkalem Chekole**, for their constant encouragement, assistance, and unwavering presence. Special thanks go to my mother-in-law, **Bizunesh Asseff Belayneh**, for her weekly calls, which have been a source of continuous encouragement.

I am also deeply indebted to my advisor, **Prof. Matteo Cristani**, whose guidance, expertise, and support have been instrumental in shaping the course of my research. Words cannot fully express the depth of my appreciation. I extend my gratitude to my co-advisor, **Prof. Pietro Sala**, for your clarity, invaluable insights, and mentorship throughout my Ph.D. journey.

Additionally, I would like to thank **CSQA** and all the management members of the organization for their support. A special mention goes to **Dr. Carlo**, the CEO of the company, whose financial backing during my Ph.D. studies has been essential for the success of this work.

Finally, my sincere appreciation goes to the **University of Verona** for providing me with this incredible opportunity to pursue my research and academic development.

Table of Contents

Lists of acronyms

AI Artificial Intelligence

BPM Business Process Management

List of Figures

List of Tables

List of Algorithms

Chapter One

Introduction and Background

1.1 Introduction

The heart of every thriving city is the core of modern commerce. Towering skyscrapers symbolize human ambition, while businesses of all sizes contribute to the vitality of this urban jungle. Whether multinational corporations or local enterprises, they showcase the spirit of ambition, creativity, and resilience. These companies provide the foundation for economic growth, innovation, and community development. A business process is a series of actions and events to achieve a specific outcome. Each activity in this process takes time and involves multiple steps. For instance, making a pizza requires mixing ingredients according to a particular timeline. The final product is created by following the correct sequence(ensuring technical correctness and model consistency) of steps—mixing the dough, preparing the toppings, and baking the pizza. However, if the steps are not followed in order, the result may not be a pizza at all.

On the other hand, an event is a specific point in time with no inherent duration. However, its effects may persist for some time, depending on the process. It is a singular occurrence that marks a specific moment within the business process. For example, when the oven reaches the desired temperature for baking the pizza, it is an event. It is a crucial point that signals readiness to proceed with the next step but does not take time to complete.

A task is an easy-to-understand activity that can be thought of as a single work unit. A task has inputs and outputs, just like the entire business process. Before a task can start, a few prerequisites need to be fulfilled in some processes but it's not mandatory. A Business process consists of events, tasks, and decision points, which are instances where important choices are made that affect the future course of the business process. These decisions are typically made by individuals, earlier events, or earlier tasks within the business process For example, while kitchen assistants can decide how to arrange the toppings on a pizza, only the head chef can determine whether the pizza dough is ready for baking.

Business Process Management (BPM) is a process optimization methodology. It provides valuable insights into a business's capabilities and activities, which can be leveraged for continuous improvement. BPM is a holistic approach that treats processes as strategic assets that must be understood, analyzed, and refined to consistently deliver superior products and services to end users. These processes are essential for business operations, as businesses ultimately depend on the efficiency and effectiveness of their processes.

Business Process Management (BPM) methodology or approach to be more effective, business processes must be accurately represented. Typically, professionals, business managers, or process owners prefer simple and easy-to-understand representations. This is where business process modeling technology comes in. It provides a robust framework for modeling, analyzing, improving, and automating organizational activities. The field is well-established and widely adopted in the industry, encompassing various methodologies. These methodologies include graphical modeling languages that facilitate stakeholder understanding, such as EPC [46] and BPMN[1], for formal analysis and automated verification of processes.

The business process is a crucial mechanism at the core of any successful company meaning the business process is a fundamental and essential system within a company that plays a key role in its success. It refers to the structured series of tasks, activities, or workflows designed to achieve specific business goals efficiently and effectively. A well-

---

[1] https://www.bpmn.org/

defined and optimized business process ensures smooth operations, consistency, and the ability to deliver value to customers, which are critical for the success of any organization. It is a complex system that converts intangible concepts into concrete goods and services. Like a living organism's circulatory system.

In today's dynamic and fast-paced business environment, the importance of sound business processes cannot be overstated. Businesses operate in a competitive landscape characterized by rapid technological changes and fluctuating market needs. To thrive, businesses must continuously innovate, adapt, and optimize their operations. This requires a thorough understanding of the entire operation meaning tasks/activities, events, and the ability to efficiently manage them.

Business processes consist of a series of coordinated activities or tasks executed by a company to achieve specific goals or provide value to customers or end users. These Business processes are vital for the functionality and success of the company as they determine how tasks and responsibilities are organized to achieve desired outcomes. They encompass core activities in a company such as production, sales, and customer service, and are essential for optimizing organizational efficiency and effectiveness.

The business process is the foundation for product development, customer value delivery, and strategic decision-making.

Among the various business processes across different sectors, supply chain management serves as an excellent example to illustrate the concept. Supply chain management meticulously plans the flow of goods and materials from suppliers to manufacturers and then to distributors, ensuring that items are delivered to the correct location promptly.

During each phase of the entire business process, technology becomes increasingly essential for streamlining corporate operations. Cutting-edge tools such as enterprise resource planning (ERP), customer relationship management (CRM), and business intelligence software provide real-time data and analytics, enabling businesses to make well-informed decisions. Automation technologies streamline repetitive processes, while artificial intelligence enhances operational efficiency and strategic planning by providing predictive insights.

In this dynamic and intricate ecosystem, success depends on the seamless integration of multiple systems and/or phases of the system. Each phase is interconnected and contributes to the overall effectiveness of the enterprise. Business processes that efficiently manage these procedures not only achieve operational excellence but also drive economic growth and enhance the quality of life for their communities and clients. The relentless pursuit of business process excellence steers companies toward a profitable future in an ever-evolving environment

Beyond the operational effectiveness of business processes, understanding and optimizing them is critical in today's world. This involves adhering to moral and legal requirements, industry best practices, and ethical standards. Achieving sustainable success, maintaining a positive reputation, and mitigating risks all depend on business processes conforming to these criteria. This brings us to the concept of *compliance*, which emphasizes ensuring that business processes or activities are conducted in accordance with specific guidelines to minimize adverse effects on the environment, society, and the organization itself.

There are different types of compliance that a company must consider such as :

1. **Regulatory Compliance:** This involves adhering to external laws and regulations imposed by governmental bodies. It encompasses financial regulations, data protection laws, health and safety standards, and more.

2. **Environmental Compliance:** This refers to complying with environmental laws and regulations to minimize the impact on the environment. It includes meeting emissions standards, following waste disposal regulations, and implementing sustainability practices.

3. **Goal Compliance:** This entails aligning business processes with predefined company goals and objectives. It ensures that the processes support the strategic aims and targets of the organization.

4. **Impact Compliance:** This ensures that the impacts of business processes remain within acceptable thresholds, thus minimizing negative consequences. It involves

assessing and managing the potential effects of business processes on various aspects of the business.

Kiichiro Toyoda, the founder of Toyota Motor Corporation, strongly believed in the philosophy that 'the ideal conditions for making things are created when machines, facilities, and people work together to add value without generating any waste.'[2] Waste, which refers to unwanted or unusable materials or substances discarded after primary use, is often considered worthless, defective, or of no use. This by-product, whether polluting or non-polluting, must be properly managed to avoid negatively impacting the community, nature, and the environment. To manage it effectively, it must first be clearly described and quantified.

A key motivation for focusing on *impact compliance* is to ensure adherence to all legal and regulatory obligations. Governments and regulatory agencies worldwide have established various laws and guidelines to protect the environment, uphold consumer rights, and promote fair labor practices. Non-compliance with these requirements can lead to severe consequences for businesses, including significant fines, legal actions, and reputational harm. By integrating impact compliance into their business processes, organizations can proactively manage and mitigate legal risks while ensuring their activities remain within the boundaries of the law.

Business process compliance can be seen as a set of methodologies used to evaluate processes based on the presence of an execution (or trace) that adheres to imposed constraints. This approach involves both imposing constraints and evaluating all possible executions of the process. This issue is particularly relevant in contexts such as regulatory compliance, where the goal is to verify the process against normative standards, including soft regulations like guidelines and product specifications.

These days businesses are increasingly concerned about environmental sustainability due to the growing recognition of the effects of climate change and environmental degradation. These businesses face pressure to reduce their carbon footprint, manage waste efficiently, and conserve natural resources. *Impact compliance* plays a crucial role in en-

---

[2] https://www.teamguru.com/blog/25-inspirational-business-process-improvement-quotes/1632.

suring that businesses adhere to sustainability guidelines and environmental legislation by assisting them in tracking and managing the environmental impact of their operations. By doing so, companies not only mitigate their environmental impact but also elevate their compliance standards with stakeholders, including investors, customers, and other groups that prioritize sustainability.

The ultimate goal of **impact compliance** is to ensure the long-term viability of a company. In a world where resources are limited and the impacts of corporate activities on society and the environment are closely monitored, adopting sustainable business practices is not just preferable but essential for survival. By incorporating impact compliance into their core strategies, businesses can future-proof their operations, ensuring they remain competitive and viable amidst evolving challenges.

It is also crucial to recognize the profound significance of ensuring compliance. It establishes a solid foundation for ensuring that company operations adhere to ethical, legal, and regulatory requirements while minimizing negative effects on the environment and society. By embracing impact compliance, businesses can achieve operational excellence, mitigate risks, foster innovation, and promote sustainability.

1.2 Background

The last few decades have significantly reshaped the corporate environment due to *globalization*, *technological breakthroughs*, and rising *social demands*. These developments have made it essential for businesses to adapt their operations, and management practices, and ensure compliance with various evolving standards. To appreciate the importance and relevance of compliance in the current corporate landscape, it is necessary to first understand the history of these changes.

Traditionally, the main considerations in corporate processes were *profitability* and *efficiency*. Businesses often paid little attention to the broader *social and environmental* effects of their activities, focusing instead on streamlining operations, cutting costs, and maximizing outputs. In an era of minimal regulatory oversight and low social expectations regarding corporate behavior, this approach was often sufficient.

However, significant changes were brought about by the *industrial and technological revolutions* of the 20th and 21st centuries. The advent of the internet, automation, and advanced data analytics revolutionized business operations, enabling unprecedented levels of scale and efficiency. These developments also highlighted the *interdependence* of global supply chains and markets, demonstrating how decisions related to a single task, event, or activity could have far-reaching consequences

As companies grew in size and influence, governments and international organizations recognized the need for *regulatory frameworks* to govern corporate activities. This led to the establishment of numerous laws and regulations aimed at protecting the environment, ensuring ethical labor practices, and safeguarding consumer rights. Notable regulatory milestones include the adoption of the *International Labour Organization* (ILO) standards, the *European Union's General Data Protection Regulation* (GDPR), and the *U.S. Clean Air and Water Acts*[3].

Simultaneously, there was a growing emphasis on *corporate social responsibility* (CSR) and *ethical business practices*. Organizations like the *United Nations Global Compact* and the *World Business Council for Sustainable Development* (WBCSD) encouraged businesses to adopt sustainable and socially responsible practices. This shift was driven by an increasing awareness of global challenges such as *resource depletion*, *social inequality*, and *climate change*.

Compliance, in its broadest sense, refers to adherence to laws, regulations, guidelines, and *ethical principles*. Initially, the primary goal of compliance efforts was to avoid legal penalties and reduce costs. Companies established compliance departments and implemented processes to ensure that their activities met the minimum legal standards.

Over time, the scope of compliance expanded to include a wider range of standards and requirements. This expansion was driven by several factors, including the growing complexity of regulatory frameworks, the rise of corporate scandals, and increasing stakeholder demands for greater *accountability* and *transparency*. Modern compliance systems now encompass *human rights*, *data privacy*, *anti-corruption measures*, and *environmental*

---

[3] ILO standards focus on labor rights, GDPR addresses data privacy, and the U.S. Clean Air and Water Acts regulate environmental protection.

sustainability.

Business Process Management (BPM) emerged as a structured approach to enhance and optimize organizational operations. *Business Process Management (BPM)* involves the design, analysis, execution, monitoring, and optimization of business processes to meet predefined objectives. It includes a variety of tools and techniques, such as workflow automation, performance measurement, and process modeling.

Business Process Management (BPM) is critical for improving operational efficiency, reducing costs, and enhancing the quality of products and services. By systematically managing their processes, companies can identify bottlenecks, eliminate redundancies, and implement best practices. Moreover, BPM supports compliance efforts by providing a formal framework to ensure that processes adhere to legal and ethical standards.

The integration of technology has transformed both compliance and corporate process management. *Real-time data* and *analytics* provided by *business intelligence tools*, *Customer Relationship Management* (CRM) software, and *Enterprise Resource Planning* (ERP) systems enable businesses to monitor and optimize their operations more effectively.

Automation and *Artificial Intelligence* (AI) further enhance BPM by automating repetitive tasks, predicting potential issues, and providing insights for decision-making. For instance, AI-driven analytics can detect patterns of non-compliance and help businesses take proactive measures to address them. *Cloud computing* makes these technologies accessible and scalable, allowing businesses to adapt to the changing demands of modern enterprises.

The next stage in the evolution of compliance is *impact compliance*, which goes beyond traditional frameworks by focusing on the *actual impacts* of corporate activities on the *environment*, *society*, and the *economy*. This approach acknowledges that business processes must not only comply with regulations but also manage and mitigate the broader effects of their operations.

Impact compliance is based on the understanding that corporate actions have wide-ranging implications. For instance, supply chain practices can affect labor conditions, manufacturing processes can harm the environment, and product usage can impact consumer health and safety. The goal of impact compliance is to align corporate activities with broader societal and environmental goals by ensuring that these impacts remain within acceptable boundaries.

From a strategic perspective, impact compliance offers several advantages. First, it helps businesses navigate the increasingly complex and dynamic regulatory environment. By proactively managing their impacts, companies can reduce risks, avoid penalties, and enhance their reputation.

Second, impact compliance encourages sustainable business practices. It drives companies to adopt *greener technologies*, reduce waste, and improve *resource efficiency*. These practices not only benefit the environment but also reduce costs and improve operational efficiency.

Third, impact compliance meets growing stakeholder expectations. Communities, employees, investors, and customers increasingly demand that businesses operate ethically and transparently. A commitment to impact compliance can foster *stakeholder trust* and *loyalty*.

Finally, impact compliance fosters *innovation*. Businesses are compelled to explore new technologies, processes, and business models to manage and mitigate their impacts. This innovation can lead to the development of sustainable products and services, opening up new market opportunities and enhancing competitive advantage.

In conclusion, the evolution of impact compliance is grounded in the changing corporate environment, the rise of ethical and legal standards, the importance of business process management, and the integration of technology. As organizations face increasing pressure to operate responsibly and sustainably, *impact compliance* emerges as a critical framework for managing the broader implications of business processes. By implementing impact compliance, businesses can ensure regulatory compliance, promote environmental sustainability, mitigate risks, drive operational innovation, and meet stakeholder expectations. This comprehensive approach not only supports corporate success but also contributes to a fairer and more sustainable future.

1.3 Motivation

Impact compliance research and implementation are driven by several factors that underscore the importance of companies conducting their operations more responsibly and comprehensively. These factors include, but are not limited to, *risk management*, *market expectations*, *ethical issues*, *environmental sustainability*, *legal requirements*, and the desire for *innovation* and a *competitive edge*.

Organizations are adopting *impact compliance* primarily due to mounting regulatory pressure on a global scale. Governments and international regulatory bodies are continually introducing new rules and regulations to protect the environment, ensure fair labor practices, defend consumer rights, and promote ethical business conduct. Failing to comply with these regulations can lead to significant fines, legal actions, and reputational damage.

For instance, environmental regulations such as the REACH (Registration, Evaluation, Authorization, and Restriction of Chemicals) directive of the European Union place strict limitations on the use and disposal of hazardous materials[4]. Similarly, data protection legislation like the *General Data Protection Regulation* (GDPR) imposes stringent guidelines for managing personal data. These regulatory frameworks demand robust compliance systems to ensure proactive management of the impacts of business activities and adherence to legal standards.

In today's market, consumers are more discerning and better informed than ever before. They expect the businesses they engage with to be both ethically and environmentally responsible, demanding accountability and transparency. This shift in consumer behavior stems from increasing awareness of global issues like social inequality, climate change, and human rights violations.

Companies that fail to meet these expectations risk losing customers to more socially and environmentally conscious competitors. On the other hand, businesses that demonstrate a commitment to *impact compliance* can enhance their brand perception, foster

---

[4] For more information on the REACH directive, see: `https://en.wikipedia.org/wiki/Registration,_Evaluation,_Authorisation_and_Restriction_of_Chemicals`

customer loyalty, and stand out in a crowded marketplace. For example, companies that adopt sustainable practices and communicate them transparently often receive positive customer feedback, leading to increased market share and profitability.

Ethical concerns and the broader movement of corporate social responsibility (CSR) are major drivers of *impact compliance*. Businesses increasingly recognize their responsibility to contribute positively to society and the environment. This recognition extends beyond the pursuit of profit to include the well-being of employees, local communities, and the planet.

Practices such as fair labor standards, community involvement, and environmental stewardship are becoming integral components of business strategy. By implementing *impact compliance*, companies ensure that their operations align with these ethical principles and CSR objectives. This alignment fulfills moral obligations, strengthens stakeholder trust, and enhances the company's reputation.

Addressing environmental challenges is another key driver of *impact compliance*. Urgent global concerns such as biodiversity loss, pollution, resource depletion, and climate change require immediate action and sustained efforts. Businesses have a critical role in minimizing these environmental impacts through sustainable practices.

Impact compliance encourages businesses to adopt strategies such as reducing carbon emissions, conserving resources, minimizing waste, and supporting renewable energy initiatives. These actions contribute to environmental sustainability and help businesses comply with environmental laws. Additionally, sustainable practices often result in cost savings and operational improvements, benefiting both the environment and the economy.

Effective *risk management* is also a significant motivator for *impact compliance*. Companies face a range of risks related to their operations, including reputational, financial, operational, and legal risks. Non-compliance with regulatory standards can lead to fines, lawsuits, and operational disruptions, while unethical behavior can result in negative publicity and loss of stakeholder trust.

Impact compliance provides a framework for identifying, assessing, and mitigating these risks. By ensuring that their operations meet legal, ethical, and environmental standards, businesses can improve their resilience and protect themselves from potential liabilities. This proactive approach to risk management is crucial for maintaining long-term profitability and business continuity.

Pursuing *impact compliance* can also foster *innovation* and provide a competitive edge. In their efforts to manage and mitigate the effects of their operations, companies often find themselves exploring new technologies, processes, and business models. This innovation can lead to the development of sustainable products and services that meet evolving consumer and market demands.

A growing number of stakeholders—including communities, suppliers, employees, and investors—are demanding that businesses act ethically and transparently. Communities hold companies accountable for their social and environmental impacts, suppliers seek ethical partnerships, employees desire purpose-driven workplaces, and investors look for sustainable investment opportunities.

By implementing *impact compliance*, companies can meet stakeholder expectations and build trust. Demonstrating a commitment to ethical and sustainable operations allows businesses to foster positive relationships with stakeholders, enhance their social license to operate, and create long-term value.

Ultimately, *impact compliance* is driven by the realization that sustainability and long-term business success are deeply interconnected. Companies that neglect their social and environmental responsibilities may see short-term profits but are likely to face significant challenges in the future. Conversely, businesses that integrate *impact compliance* into their core operations can achieve profitability, resilience, and sustainable growth.

By aligning their operations with broader societal and environmental goals, businesses contribute not only to their success but also to the well-being of communities and the planet. *Impact compliance* is driven by a range of factors, including stakeholder trust, legal obligations, market expectations, ethical concerns, environmental sustainability, risk management, and long-term business performance. In an increasingly complex and interconnected world, businesses must adopt a comprehensive approach to compliance that goes beyond mere regulatory adherence. *Impact compliance* provides a framework for managing and mitigating the broader impacts of business operations, ensuring that companies operate ethically and sustainably. This approach not only upholds moral and legal standards but also drives innovation, enhances competitiveness, and contributes to a sustainable future.

1.4 Objectives

1.4.1 General Objective

The primary goal of this research is to explore and define the concept of *impact compliance* within the context of *business process management*, with a particular focus on the *agriculture industry*. This study aims to develop frameworks and methodologies that ensure business operations comply with environmental regulations, which means *compliance checking*, particularly those targeting the reduction of *pollutants* and *carbon dioxide emissions*, thereby promoting sustainable practices.

To achieve this goal, the research will address the following key questions:

1. How can *impact compliance* be effectively defined and conceptualized?
2. What are the current practices in *business process compliance*, and how can they be evaluated?
3. What methodologies can be developed to measure and enforce *impact compliance*?
4. How can *computational methods* be applied to verify *impact compliance*?
5. How can *impact compliance* be integrated into *business process management* to reduce greenhouse gas (GHG) emissions such as carbon di oxide?

1.4.2 Specific Objectives

1. **Define and conceptualize impact compliance**

- To develop a clear and comprehensive definition of *impact compliance*, differentiating it from related concepts such as *regulatory compliance* and *conformance* within the context of business process management.

2. **Establish a theoretical framework for impact compliance**
    - To formulate a theoretical foundation that supports the concept of *impact compliance*, clarifying its significance in business processes.
3. **Evaluate current business process compliance practices**
    - To conduct a thorough literature review of existing methodologies in *business process compliance*, identifying gaps and limitations that can be addressed by the concept of *impact compliance*.
4. **Develop methodologies for measuring and enforcing impact compliance**
    - To design enforcement mechanisms ensuring adherence to established *environmental constraints* within business processes.
5. **Integrate impact compliance into business process management**
    - To provide practical recommendations for companies on incorporating *impact compliance* into their processes, with a special emphasis on its application in the *agriculture sector*.

These objectives aim to establish a comprehensive understanding of *impact compliance* and offer actionable solutions for its effective implementation, particularly focusing on environmental sustainability.

1.5 Significance

Impact compliance is more than just following regulations; it is a revolutionary strategy that can enhance corporate responsibility, promote sustainability, and reshape how businesses operate. *Impact compliance* is a complex topic that affects many areas of a business as well as its larger ecosystem. We detail each of these dimensions below, highlighting the significant ramifications and advantages of implementing *impact compliance* procedures.

Impact compliance promotes an open and accountable culture. Companies gain the trust of various stakeholders, such as investors, employees, customers, and regulators, by upholding legal, ethical, and environmental standards. Long-term success depends on this trust because it raises the company's profile and credibility in the marketplace.

Impact compliance frequently reduces waste and improves resource efficiency. Businesses can reduce operating expenses and boost profitability by implementing sustainable measures, including resource optimization, energy savings, and waste recycling. These methods support environmental sustainability in addition to cost savings.

A strong compliance and sustainability record is becoming more desirable to investors. *Impact compliance* can lead to new funding opportunities and attract socially conscious investors. Businesses that prioritize sustainability are generally seen as lower-risk investments, making them more appealing to a broader range of investors.

Long-term profitability can result from sustainable business practices, which ensure that organizations are better equipped to adapt to changes in the market, regulations, and environmental challenges.

Impact compliance encourages businesses to adopt strategies that reduce their environmental impact. This includes reducing waste generation, conserving energy and water, and lowering greenhouse gas emissions. Businesses combat climate change and environmental degradation by taking proactive measures to manage their environmental effects.

Businesses that adhere to environmental laws and regulations help preserve ecosystems and biodiversity. Through responsible resource management and pollution reduction, businesses can contribute to preserving the environment for future generations.

Impact compliance also ensures that businesses comply with health and safety regulations, protecting both workers and communities. This commitment to safety enhances the overall quality of life and improves public health in the regions where businesses operate.

By ensuring adherence to all applicable legal and regulatory obligations, *impact compliance* helps businesses avoid fines, penalties, and sanctions associated with non-compliance.

Maintaining compliance also enhances a company's reputation with regulatory bodies and reduces the likelihood of legal challenges.

Through the integration of *impact compliance* into their operations, businesses can anticipate and effectively manage potential risks. This proactive approach mitigates risks before they escalate into serious issues, fostering stability and smoother business operations.

Impact compliance aligns with international sustainability goals, such as the United Nations' *Sustainable Development Goals (SDGs)*[5]. Companies that incorporate these objectives into their operations contribute to broader efforts to address global challenges like poverty, inequality, and climate change.

Consumers are becoming more conscious of the societal and environmental impact of their purchases. Businesses that focus on *impact compliance* can meet the growing demand for sustainably and ethically produced goods, thereby increasing their appeal to a broader customer base.

The significance of *impact compliance* is broad and profound. It encompasses fostering environmental stewardship, enhancing corporate accountability, delivering financial and economic benefits, and providing social and community advantages. *Impact compliance* also strengthens competitive advantage, encourages innovation, and offers legal and regulatory benefits. It positions businesses to succeed and adapt in the long term by enabling them to respond to consumer demands and global environmental trends. Adopting *impact compliance* is not only a legal necessity but also a strategic imperative in today's increasingly interconnected and environmentally conscious society. It has the potential to revolutionize enterprises and pave the way for a sustainable future.

1.6 Scope

The purpose of this study is to investigate the concept of *impact compliance*, with a focus on the agricultural industry. This sector is highly interconnected with climate change and represents a major source of carbon dioxide emissions. Although *impact compliance*

---

[5] https://sdgs.un.org/goals

is a significant and pertinent issue in many industries, such as manufacturing, health, and others, this study focuses solely on the agricultural sector. The main objective is to understand how *impact compliance* can be used to mitigate the negative effects that agricultural activities have on the environment.

The agricultural sector plays a crucial role in *food production* and *rural development* and is an essential part of the global economy. However, it also contributes significantly to environmental issues such as *greenhouse gas emissions*, specifically *carbon dioxide (CO2)*, and other pollutants. By focusing on *impact compliance* as a tool to manage and reduce the environmental impact of agricultural operations, this study aims to address these challenges.

This study primarily focuses on *carbon dioxide (CO2)* emissions, which are a prominent result of many agricultural practices, including soil cultivation, livestock rearing, and the use of synthetic fertilizers. *CO2* reduction is vital for achieving *environmental sustainability* in agriculture, as it is one of the primary greenhouse gases contributing to global climate change. The goal of this study is to explore methods for monitoring, recording, and minimizing *CO2* emissions to ensure that the agricultural sector adheres to environmental standards and regulations.

In addition to *CO2*, this study will consider other environmental pollutants, such as the potent greenhouse gas *methane (CH4)* emitted by livestock and *nitrous oxide (N2O)* produced by nitrogen-based fertilizers. The study will evaluate how *impact compliance* can help mitigate these pollutants and improve the overall environmental performance of the agriculture sector.

The study aims to integrate *impact compliance* with *environmental management practices* to ensure that agricultural operations not only meet legal requirements but also contribute positively to environmental protection and enhancement.

By concentrating on these areas, the study provides valuable insights and practical recommendations for improving *impact compliance* in the agricultural industry, ultimately leading to more environmentally friendly and sustainable farming practices.

1.7 Beneficiaries

Impact compliance is a concept that benefits a wide range of stakeholders, including communities, businesses, workers, customers, investors, and the environment. Organizations can generate a chain reaction of favorable outcomes that support long-term success and sustainable development by implementing *impact compliance*.

Companies that follow *impact compliance* establish a reputation for accountability, transparency, and ethical conduct. This enhanced reputation may result in more loyal customers, a more valuable brand, and a competitive advantage in the marketplace. Nowadays, customers are more inclined to support businesses that demonstrate a commitment to social and environmental responsibility.

Impact compliance helps companies identify and mitigate the risks of non-compliance, such as fines, legal ramifications, and damage to their reputation. By taking proactive steps to address compliance issues, businesses can avoid costly disruptions and ensure business continuity.

Impact compliance ensures that products meet ethical, moral, and safety standards. Customers can trust the products they purchase to be reliable, safe, and produced in a socially and environmentally responsible manner.

Consumers benefit from greater transparency and access to information about the products they purchase. Businesses that engage in *impact compliance* often provide detailed information about their sourcing, production processes, and sustainability initiatives, empowering consumers to make informed decisions.

Customers are more likely to trust and remain loyal to companies that demonstrate a commitment to sustainability and ethical behavior. *Impact compliance* ensures that businesses act responsibly and ethically, helping to build this trust.

Impact-compliant businesses contribute to the social and economic advancement of the communities in which they operate. By promoting job creation, supporting local suppliers, and participating in community development programs, businesses can enhance the well-being of local populations.

Environmental protection: *Impact compliance* incentivizes companies to adopt environmentally conscious practices. Reducing waste, conserving natural resources, and improving air and water quality benefit communities by enhancing public health and quality of life.

Impact compliance ensures that corporate operations do not deplete or degrade the environment by encouraging the responsible use of natural resources. Sustainable practices such as resource optimization, recycling, and energy conservation contribute to healthier ecosystems in the long term.

By adhering to environmental laws and guidelines, businesses can significantly reduce their environmental impact. This reduction helps combat climate change and environmental degradation by lowering greenhouse gas emissions, reducing pollution, and minimizing waste.

Impact compliance encourages businesses to adopt strategies that preserve biodiversity and natural habitats. By minimizing their environmental footprint, businesses can contribute to the preservation of global biodiversity and the resilience of ecosystems.

A wide range of stakeholders, including businesses, workers, customers, investors, communities, and the environment, reap substantial benefits from *impact compliance*. Companies gain by enhancing their reputation, reducing risks, improving operational efficiency, and fostering innovation. Employees benefit from safe working conditions, job satisfaction, and opportunities for professional growth. Customers gain access to transparent, high-quality, and safe products. Investors see reduced risks and the generation of long-term value. Communities benefit from social and economic advancement, environmental protection, and corporate support. Finally, the environment benefits from reduced pollution, preserved biodiversity, and sustainable resource management. By embracing *impact compliance*, organizations can achieve sustainable growth and profitability while making a positive and lasting impact on both society and the environment.

1.8 Dissertation Outline:

This dissertation aims to tackle a crucial problem in business process management: can a business process be implemented while maintaining a set of limit constraints for variables that gauge the influence of undesirable effects or *impact*? We call this issue *impact compliance*. This dissertation is structured to systematically investigate this idea and its applications.

Chapter 1: Introduction and Background

The introduction chapter establishes the framework for the dissertation by providing a thorough review of business, business process modeling, impact, and impact compliance. It highlights the motivation for this study by identifying the research gap, the advantages of addressing this topic, and the scope and beneficiaries of the dissertation. This chapter also outlines the main objectives of the dissertation and provides a summary of its overall structure.

Chapter 2: Literature Review

Chapter two provides a comprehensive analysis of the existing research on *impact compliance* and business process compliance. It reviews previous studies, conceptual frameworks, and practical applications, providing readers with a solid conceptual foundation for understanding the current state of knowledge in this field. This review sets the stage for identifying gaps and demonstrating the need for further research.

Chapter 3: Business Process Compliance with Impact Compliance

The third chapter delves into the core concept of *impact compliance*. It emphasizes the importance of ensuring that business processes comply with legal and environmental constraints to mitigate adverse effects. The chapter addresses the complexity of verifying compliance, particularly in relation to *NP-complete* problems, and examines the structural conditions under which compliance-checking problems are solvable in polynomial time. It also explores scenarios where deterministic machines can be used to solve specific compliance problems. The chapter includes case studies to demonstrate the practicality of *impact compliance* approaches and provides methods for quantifying and enforcing constraints.

Chapter 4: An Approach for Computing Similarity Between Business Processes Based on Resource Consumption Impacts

This chapter presents a novel approach to assessing business process similarity by incorporating resource consumption, an often-overlooked factor in traditional methods. Two similarity measures—Modified Cosine Similarity and Euclidean-Based Similarity—are developed by modeling task resource usage as vectors.

An algorithm is introduced to compute resource impacts for BPMN process models in SESE form, evaluating average, minimum, and maximum impacts across process traces. This resource-aware approach enhances process analysis and optimization by providing a more comprehensive similarity assessment.

Chapter 5: Conclusion and Recommendations

The final chapter summarizes the key findings of the dissertation, emphasizing their relevance to business process management. It discusses the limitations of the current research and suggests potential directions for future work. Lastly, the chapter offers reflections and practical recommendations for both practitioners and scholars of business process compliance.

List of Relevant Publications

A portion of the work from this doctoral dissertation has been published in information systems journals. The published articles are listed below.

- **T. C. Workneh**, P. Sala, R. Rizzi, and M. Cristani. *Business Process Compliance with Impact Constraints*. In *Information Systems* , 2024.

Chapter Two

Literature review

This section reviews the literature relevant to the research topic. First, I will refer to the general concepts of conformance and compliance checking then provide a review of those papers that treat the topic of compliance and conformance when some attention is posed to matters related to the impact of a business process (or close concepts), including the analysis of the cost, workload, and time.

I also demonstrate that certain aspects of flow control are relevant to the conceptualization of this research. However, I approach the topic differently, particularly in relation to the ability of flow control to establish unifying semantics—an interpretation that contrasts with the perspective taken in this study

Compliance and conformance many conformance-checking techniques have been proposed over the years, and they mostly focused on control flow aspects as in Tsoury et al.'s study [43]. However data, resources, time, and *impact* are some aspects that have been discussed, for instance, in De Leoni et al.'s paper [12], but at a very general level. In this research, I focus on impact aspects that are, per se, relevant, and that I illustrate as a whole concept that spans the above-listed possible aspects.

At large, conformance is relevant, especially in *process mining* methods, and *compliance* could result from certain dual comparisons typical of this field. In particular, I can define conformance as the problem of deciding if a given sequence of events could be a trace of a designed Business Process. Conversely, when generating a model of a Business Process by observing and summarizing a set of traces, I design a mined process. To prevent overfitting, most mining methods discard deviations—executions that could realistically represent non-conforming cases.

The dual problem is in fact *compliance*. Checking compliance has been proposed as a topic that could be automated since some time [19] and dealt with in a significant number of studies, with a specific emphasis on *regulatory compliance* by many scholars, including some of the authors of this papers [23, 34, 35, 21, 22]. This research introduces the *impact compliance* that could occur when, during regulatory compliance checks, many constraints on the values of measurable variables are superimposed. This is a widespread phenomenon in the current legislation, when we aim at stopping pollution, reducing water consumption, or electricity consumption, and reducing waste generation, or more generally, reducing the carbon footprint of industrial processes. my study concentrates on how to add *numeric* constraints while performing the analysis of a Business Process. This could be considered, to many extents, a technique, thought to solve the issue of applying the constraints to the original graph describing the required behavior of a business process, but it cannot be mapped onto the Argumentation Framework, without stretching the correspondences.

Cost, energy, and time: the green business process management paradigm There are four areas in which I can detect investigations related to this study. One area investigates the notion of *green business process management*, one looks at *cost-aware* business processes, one at *energy-aware* ones, and finally *resource-aware* business processes. Before going into these specialized fields, I need to mention a research area where the notion of effectiveness, cost, time, or other variables to measure the impact of a process has been on the table in many scholars' works: process query languages. In particular, the contribution of Delfmann et al. [14] to the field of Diagramed Query Languages and the specific application to the domain of process query has shown that it is possible to query a `BPMN` for specific patterns, and this can be used for identifying the subgraphs that are costing more to the whole cost (or time) of the execution. Process Query languages [37, 36] are a fruitful research path, and therefore allow empirical choice of better execution patterns. The idea of the research above is to provide also room for a multi-target analysis, and it has been employed for empirical approaches to holistic evaluation of processes in different contexts, including Physics [50] but the most relevant case is the pattern-driven business process (re)design that has used in some cases the ideas of Delfamm et al., in particular in [49] and in [26]. These studies are crystal clear about one point: there is still no formal way to treat the problem of choosing the optimal paths of a business process.

Scholars have used the term ‘Green BPM’ to describe a class of technologies that leverage and extend existing business process management technologies to enable process design, analysis, execution, and monitoring in a manner informed by the carbon footprint of process design and execution, as introduced in the study by Hoesch-Klohe et al. [25]. Nowak and Leymann [33] provide a method to extract environmentally relevant patterns from existing patterns of different domains in the process mining perspective. In a previous investigation, Nowak et al. again [32] employed ecologically sustainable adaptation strategies that are described as green business process patterns to address the issue of reducing the effects of the execution of these processes. I should observe that all these investigations by Nowak and collaborators tend, differently from what has been done in this paper, to focus on methods to *mitigate* the negative effects caused by the execution of a specific task (or by a specific trace), or designing (and in the second paper, also re-designing) the Business Process in a perspective that eliminates the worst effects. This approach does not introduce a notion of constraint but can be considered similar in terms of purpose. However, Nowak’s studies tend to solve the problems from an *instrumental* point of view, by enriching the language for Business Process specification and providing semantics for measuring the impact when possible. In a sense, the premise of our discourse is here.

Some researchers also tried to build frameworks that consider the impact during the *process redesign*. These approaches span from additional information (in a sense that is similar to Nowak’s approach) from methods such as Life Cycle Assessment [27]. One fundamental issue in both these approaches (those based on schemata for design and re-design, I may name *methodological approaches*, and those based on Life Cycle Assessment) a crucial point is to identify some assessment measure, to be applied in design and re-design. There is a specific notion, that environmental performance indicators (EPIs) (as well as other relevant organizational factors related to ES and BPM) have been used by Gohar and Indulska [20]. Some studies have addressed the problems related to how to employ the representation of impact they proposed (for instance by adding information on the impact of tasks, or to traces - two different analyses, specifically conducted in different moments of the design and re-design phases) by employing a process pattern-based approach. This approach consists of providing proven, general solutions to common problems referring to social sustainability as in Schoormann et al. investigation [41]and introducing ecology-oriented Guidelines of Modeling (EGoM) as in the study of Lübbecke et al. [30].

Cost-aware business process investigations envisage a holistic approach to managing the cost of business operations in a structured manner, by making an explicit link between cost and processes in all phases of the business process management life cycle. An example of this approach is in the methodology proposed by Wynn et al. [47]. Cost-aware business process management introduces an important theme that is also decisive in the analysis we are performing in this investigation: the additivity nature of a large family of impacts. Carbon footprint, consumption of scarce resources, and cost are all kinds of impacts of an additive nature. If a task costs one and another task on the same task costs two the total cost of the path has to be computed on the base of a value of three. There can be cases in which I measure also *positive impacts*, that could add to the negative values (the *costs* to use a general term, against the *values* or the gains). In this dissertation, I shall focus on the sole *negative* impact (that we shall denote by positive values, aiming at reducing the total sum).

The approach of Bolsinger et al. [6] has been a somewhat seeding viewpoint, especially in the community of Process Management, and is the result of a field of research named *value-focused process engineering*, that has been the focus of some scholars, including [31] who discussed the risk perspective in value-focused process engineering, and others [39, 40, 11]. The approach they follow has inspired the work of Rosemann as well [38] who employed the ideas of Bolsinger et al. to devise *patterns* to detect improvement space in Business Process Management. The work by Bolsinger et al. discusses, in particular, how to choose a route that is convenient for performing better a given Business Process, among all possible paths. The idea may appear similar, but several points are different. First of all, the approach proposed by Bolsinger et al. is purely empirical and has the sole purpose of defining a method, not an algorithm. Moreover, the proposed method is applicable *only* to scalar labeling and thus constitutes a small portion of the domain dealt with in this research. Finally, the proposed method essentially recomputes the labeling by a rewriting based on a direct exclusion that, while applying with polynomial complexity to the scalar case, is not as efficient as our approach. In Section 3.2 I show that the computational cost of the proper vectorial labeling, which is potentially exponential, actually follows polynomial patterns in several cases. The only natural extension of the approach by Bolsinger et al., indeed taken seriously by Bisogno et al. [5], is to use empirical simulation, and this results in a method that performs nicely, but it is still computationally heavy. In this research, I show that this is not a direct consequence of the structure of the problem and can be solved in polynomial time in several significant configurations.

Energy efficiency goals represent one of these additive impacts, that I aim to reduce the most. The development of green business process systems can only be achieved by recognizing their multi-layer feedback nature, as pointed out by Ardagna et al. [2]. López et al. formalized the energy-aware resource allocation problem, including time-dependent variable costs [29]. Their provision is very complex, but the algorithmic solution that they investigated does not directly solve the problem: it delegates the optimization process to a Pareto-driven auction approach that instantiates the problem and recomputes the fixpoint. This approach has a large number of advantages, but it does not apply to multiple impact measures, in fact, the cost is a metaphor for the consumption of resources and it is a single impact measure process. In further studies, inspired by Ardagna's work, Cappiello et al. [9] introduce an approach for defining energy-aware *adaptive* business process co-design, that in part lies on a similar perspective of López et al. method.

Flow control and flow management: managing resources of Business Processes Resource Business Process Management is a field that in some sense considers the above perspective. Scholars look at resources as a summary metaphor for costs and energy. The mainstream idea is to check the compliance of observed process executions considering data, resources, and control flow as a unified problem Unlike the majority of conformance-checking approaches, the fundamental distinction is not to restrict the focus to the ordering of activities (i.e., control flow). Taghiabadi et al. [42] have provided a general framework to accommodate the above-mentioned complexity of problems. They essentially follow an approach based on Process Mining concepts. The compliance view they adopt is therefore comparative. One process is compliant with another one, for its traces are also traces of the other one.

Cabanillas [7] introduces techniques for resource specification, which rely on a new resource selection language that accommodates the aforementioned notions studied by Taghiabadi et al, and provides an extensible conceptualization for the same conformance or compliance notions. Process model similarity and matching methods are used to address the problem of redundantly modeled processes by Baumann et al. [4]. Cabanillas et al. [8], further on, define RAL (Resource Assignment Language), a domain-specific language explicitly developed to assign resources to the activities of a business process model. Del Rio-Ortega et al. [13] support the definition of resource-aware PPIs (process, performance, indicators) in Business processes enriched with resource information. In general, it makes sense to consider flow as a means to determine which traces are acceptable and which are not. However, the principles underlying all methods in this field, as discussed, for instance, in [1] is that I need a *unifying* interpretation for all constraints. An example of a method like this is the *carbon footprint equivalent* that transforms any environmental impact in a measure on a unified unit. Another typical approach is *monetization*, which transforms any impact into a cost.

Chapter Three

Business Process Compliance with impact constraints

3.1 Introduction

Business Process Compliance is a family of methods to evaluate Business Processes in terms of the existence of *one execution* (one trace) that does not violate constraints superimposed on the process itself. The dual version is formulated as the superimposition of constraints and consequent evaluation of the process for *all the executions*. These problems are relevant to a large part of actual applications, especially those in the context of *regulatory compliance* where we aim at verifying the process against a normative background (including, for instance, soft ones, such as guidelines, product specification, and product standards) or goals fixed by the owner of the process. This dissertation discusses one new type of compliance, that is *impact compliance*, devised to verify when a process respects a set of constraints, to establish that certain amounts, measuring the undesired effects of the tasks executed to implement the process, are *below-given limits*. The current literature on Business Process Management, Business Process Analysis, and Business Process Compliance has not yet addressed this type of compliance checking process. As I demonstrated in the later sections of the chapter, this problem is significant and complex to address. In particular, I showed that the checking problems described above are polynomially solvable on deterministic machines under certain structural conditions. In general, however, the first problem is NP-complete, while the second is polynomially solvable on deterministic machines. I dealt with the following issue: is it possible to execute (in one single case or all possible cases) a Business Process while respecting a set of limit constraints (maximum values) for a given set of variables measuring the impact of undesired effects of the execution of the process itself? We denominate this problem *impact compliance*. This problem has not yet been considered fully. As I will demonstrate in chapter 2, some attempts have been made to deeply understand the problems related to the impact of a process. However, these attempts lack generality, do not systematically address complexity issues, and fail to consider how to evaluate processes across a variety of impact aspects simultaneously—an approach that I argue is essential in several practical cases.

FIRST: Contextualize the problem, provide a running example that will be analyzed at various points throughout this dissertation and introduce some of the basic concepts necessary for the research presented here.

Business processes are the formal representation of workflows, designed to model the behavior of agents (both humans and software) that perform various tasks with the explicit purpose of achieving a specific set of goals. This formalism is intended to manage *precedence* among tasks and evaluate *conditions* that determine whether one path, another, or multiple paths (executed in parallel) are followed during the workflow execution.

The control of execution flows to *guarantee* adherence to a set of constraints—either due to normative requirements or to ensure the attainment of goals defined by the workflow's owner—is the subject of *compliance*. On the other hand, monitoring the execution flow to *verify* that the workflow adheres to a specific model is the subject of *conformance*.

To deepen the analysis of the aforementioned concept, I present a conceptualization in Figure 1 that illustrates a situation where the idea of impact is informally discussed in the context of Business Processes.

Example 1. *To perform a specific type of production, we execute a first task (T1), then* *alternatively a task T2 or a task T3, followed by a task T4, and finally, again alternatively, a task T5, T6, or T7. We know, a priori, that the impact of the tasks on the two aspects is as follows:*

Task	***Water***	**CO_2**	***Task***	***Water***	**CO_2**
T1	*3*	*2*	*T5*	*1*	*3*
T2	*5*	*2*	*T6*	*2*	*2*
T3	*2*	*4*	*T7*	*3*	*1*
T4	*1*	*1*			

T1 consumes 3 lt of water per kg, and emits 1 unit of CO_2*; T2 consumes 5 lt of water per kg, and emits 2 units of* CO_2*; T3 consumes 2 lt of water per kg, and emits 4 units of* CO_2*; T4 consumes 1 lt of water per kg, and emits 1 unit of* CO_2*; T5 consumes 1 lt of water per kg, and emits 3 units of* CO_2*; T6 consumes 2 lt of water per kg, and emits 2 units of* CO_2*; T7 consumes 3 lt of water per kg, and emits 1 unit of* CO_2*; A schema of the business process is presented in Figure 3.1.*

- *T1 consumes 3 lt of water per kg, and emits 1 unit of* CO_2*;*
- *T2 consumes 5 lt of water per kg, and emits 2 units of* CO_2*;*
- *T3 consumes 2 lt of water per kg, and emits 4 units of* CO_2*;*
- *T3 consumes 1 lt of water per kg, and emits 1 unit of* CO_2*;*
- *T4 consumes 1 lt of water per kg, and emits 3 units of* CO_2*;*
- *T5 consumes 2 lt of water per kg, and emits 2 units of* CO_2*;*
- *T6 consumes 3 lt of water per kg, and emits 1 unit of* CO_2*;*

The problem we aim to solve involves finding an optimal execution that minimizes impact. The following set of possibilities (execution traces) is available:

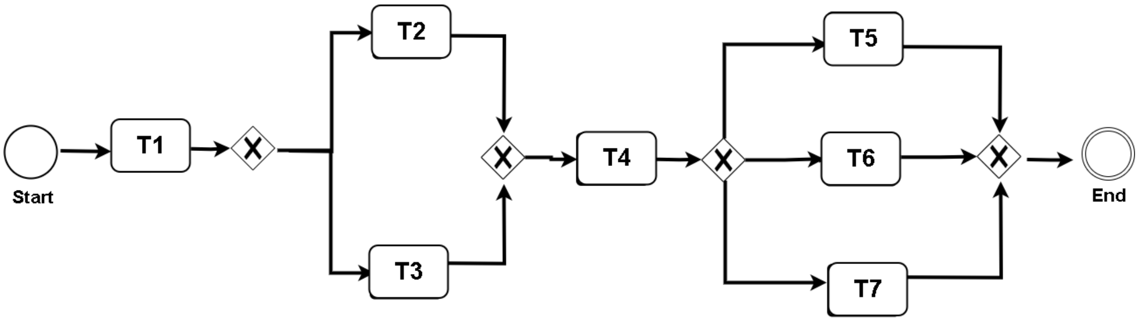


Figure 3.1: An example of a business process in the BPMN format.

1. *T1, T2, T4,T5 consumes 9 lt of water and emits 6 units of* CO_2*;*
2. *T1, T2, T4,T6 consumes 10 lt of water and emits 5 units of* CO_2*;*
3. *T1, T2, T4,T7 consumes 11 lt of water and emits 4 units of* CO_2*;*
4. *T1, T3, T4,T5 consumes 6 lt of water and emits 8 units of* CO_2*;*
5. *T1, T3, T4,T6 consumes 7 lt water and emits 7 units of* CO_2*;*
6. *T1, T3, T4,T7 consumes 8 lt water and emits 6 units of* CO_2*;*

Trace	**Tasks**	**Water**	**CO_2**
TRACE 1	T1, T2, T4, T5	10	8
TRACE 2	T1, T2, T4, T6	11	7
TRACE 3	T1, T2, T4, T7	12	6
TRACE 4	T1, T3, T4, T5	7	10
TRACE 5	T1, T3, T4, T6	8	9
TRACE 6	T1, T3, T4, T7	9	8

Table 3.1: Execution traces with corresponding water and CO_2 values.

TRACE 4 is best in terms of water consumption, while TRACE 3 is best in terms of CO_2 *emissions. If we superimpose a constraint set stating that the maximum we can accept in* *terms of water is 10 liters and in terms of emission is 9 units, the only good traces are 1,5,6. If we superimpose a constraint set that states to reduce the consumption of water to a maximum of 9 and the emissions to a maximum of 9, the only satisfactory traces are 5 and 6.*

The approach adopted in this study is to consider a business process as formalized in Section 3.2.1.

To introduce the concept of Impact-aware `BPMN` and apply the notion of *compliance* to impact, I present an example from the agricultural context. In this domain, the problem of checking impact compliance is *essential* for several reasons, including effectiveness, energy saving, and environmental protection.

Adopting practices that reduce greenhouse gas (GHG) emissions can enable agriculture to play a significant role in the global effort to combat climate change. To implement these actions effectively, it is crucial to understand the global impact of an agricultural process.

In order to exemplify a business process with impact, I consider here the **grape production cycle**, which starts with site selection and finishes with harvesting. The `BPMN` diagram for this process is depicted in Figure 3.2. I concentrate on the effects of methane, water, and carbon dioxide emissions. The purpose of this example is to illustrate a domain of application of the method, I investigate here and show how it would be effective in understanding the actual impact behavior of complex processes. This example is then used to show how the basic tool for algorithmic analysis of processes, the region tree, is devised. I shall not run the algorithm on this example for two reasons. Firstly, the example is devised to present simple, realistic but simplified processes, and this does not give an effective analysis of the behavior, which is instead analyzed in detail both from a theoretical and an experimental viewpoint. Secondly, I chose not to for the sake of conciseness.

- **Carbon Dioxide (CO_2)**: A colorless gas with a sour taste and a slightly harsh smell. It is one of the most significant greenhouse gases associated with agricultural production. As shown in Figure 3.2 and Table 3.2, there are numerous activities/tasks involved in grape production that emit Carbon Dioxide.

- **Water (H_2O)**: An essential component of agricultural productivity. In each growth life cycle, water is necessary to obtain output.

- **Methane (CH_4)**: A hydrocarbon that makes up the majority of natural gas. Since methane is a greenhouse gas (GHG), its presence in the atmosphere impacts the planet's temperature and climate.

The specification provided in Figure 3.2 represents a *simplified* version of a typical grape production process. Specifically, I have intentionally shortened the loops by anticipating decisions during the monitoring phases, eliminating the need for further tests. To ensure reliability, I have assigned the maximum required values for each task, adopting a precautionary approach in evaluating the impact.

While examining Figure 3.2, readers may notice several vertices that are not listed in Table 3.2. These vertices are not included because they are not tasks, and based on the model presented in this dissertation, they do not have associated impact vectors. These vertices are used to manage parallel tasks and alternatives (AND and OR splits and joins). I have implicitly assumed that all splits and joins have zero vector impact.

Each production step is associated with *carbon dioxide emissions*, *water consumption*, and *methane emissions*.

In this context, I focus solely on checking the compliance of a given process against a set of constraints on the maximum impact the process can have.

Looking at Figure 3.2, it is not immediately possible to estimate the actual impact of a single trace. Moreover, the set of traces is quite large (not polynomial in the number of vertices). This raises a natural computational question: Can we determine whether it is possible to execute this process (at least once, or possibly always) within acceptable limits? On the other hand, the dual question is: Can we identify a minimal execution trace?

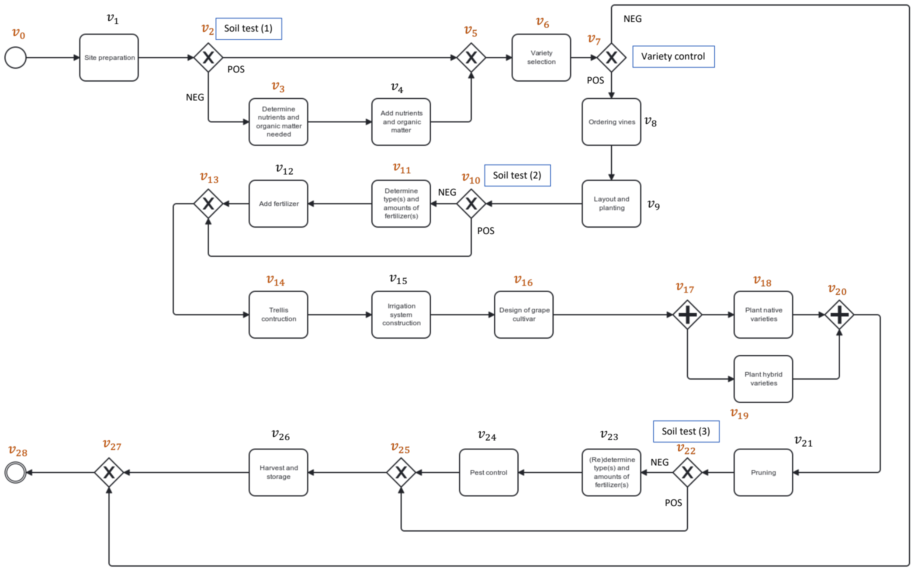


Figure 3.2: BPMN of Grapes Production. In red color the vertices with 0 impact for all components.

Task	Carbon Dioxide Emission	Water Cons.	Methane Emission
v_1	429 tph	0	0
v_4	429 tph	0	10tph
v_8[a]	632 tph	0	0
v_9	429 tph	10 cmph	429 tph
v_{12}	429 tph	10 cmph	14586tph
v_{15}	0	10 cmph	0
v_{21}	0	10 cmph	0
v_{23}	429 tph	10 cmph	0
v_{24}	429 tph	10 cmph	0
v_{26}	429 tph	0	0

Table 3.2: Impact vectors for the tasks with non-zero impact vectors in Figure 3.2.

3.2 Methodological Framework

This section will define and go into the terms that are pertinent to this study. The terms discussed are BPMN, graph representation, and algorithmic treatment of graphs.

3.2.1 Business Process Modeling and Notation

The Business Process Modeling Notation (`BPMN`), an open standard notation for graphical flowcharts that are used to define business process workflows, is a visual modeling language for business analysis applications and establishing enterprise process workflows. All business stakeholders, including business users, business analysts, software engineers, and data architects can readily understand this well-liked and clear image. It has various advantages for business analysts who create and improve the processes and also for managers who monitor and control the processes.

In this dissertation, I consider a small subset of `BPMN`, specifically one that includes only *pools, swimlanes, tasks, starting/ending events, and exclusive/parallel gateways*. However, for the purposes of this work, both pools and swimlanes are used solely as syntactic annotations, and as such, I omit them from the formal notation. A `BPMN` is a labeled directed graph, $G = (V, E, L)$. where $L : V \to \{task, start, end, \textit{Exclusive Split}, \textit{Exclusive Join}, \textit{Parallel Split}, \textit{Parallel Join}\}$, such that for every $v \in V$:

(i) if $L(v) = start$ then for every v' with $L(v') = start$ we have $v = v'$ (*unique start event*), moreover $|\{(v, v') \in E\}| = 1$ (*exactly one outgoing edge from the start event*) and $|\{(v', v) \in E\}| = 0$ (*no incoming edges in the start event*);

(ii) if $L(v) = end$ we have $|\{(v', v) \in E\}| = 1$ (*exactly one incoming edge in an end event*) and $|\{(v, v') \in E\}| = 0$ (*no outgoing edges from an end event*);

(iii) if $L(v) = task$ we have $|\{(v', v) \in E\}| = |\{(v, v') \in E\}| = 1$ (*exactly one incoming/outgoing edge for a task*);

(iv) if $L(v) = \textit{Exclusive Split}$ (resp., $L(v) = \textit{Parallel Split}$) we have $|\{(v', v) \in E\}| = 1$ (*exactly one incoming edge in a split gateway*) and $|\{(v, v') \in E\}| = 2$ (*exactly two outgoing edges from a split gateway*);

(v) if $L(v) = \textit{Exclusive Join}$ (resp., $L(v) = \textit{Parallel Join}$) we have $|\{(v', v) \in E\}| = 2$ (*exactly two incoming edges in a join gateway*) and $|\{(v, v') \in E\}| = 1$ (*exactly one outgoing edge from a join gateway*).

Let us observe that we may have multiple end events.

In the following, we assume our `BPMN` to be structured as *Single Entry Single Exit*

(SESE) Regions [16] that is given a `BPMN` $G = (V, E, L)$ we may create a *set cover* $\mathscr{R} = \{R_1, \dots, R_m\}$ of V, i.e., a set of subsets of V whose union is equal to V itself such that for every $1 \leq i \leq m$:

1 - *disjointness or inclusion* - for every $1 \leq j \leq m$ either $R_i \cap R_j = \emptyset$, $R_i \subseteq R_j$, or $R_i \supseteq R_j$;

2 - *single entry* - there exists at most one $v \in V \setminus R_i$ such that $(v, v') \in E$ with $v' \in R_i$, we will call v' (if any) the *entry-point* of R_i, denoted by $entry(R_i)$;

3 - *single exit* - there exists at most one $v \in V \setminus R_i$ such that $(v', v) \in E$ with $v' \in R_i$, we will call v' (if any) the *exit-point* of R_i, denoted by $exit(R_i)$;

4 - *region maximality* - let $submax(R_i) = \{R_j \in \mathscr{R} : R_j \subset R_i, \forall j'(R_{j'} \subset R_i \rightarrow R_{j'} \subseteq R_j \vee R_{j'} \cap R_j = \emptyset)\}$ the set of all and only maximal proper sub-region of R_i w.r.t. inclusion, then one of the following cases may arise:

   a - *branching case* - $submax(R_i) = \{R_j, R_{j'}\}$, i.e., $|submax(R_i)| = 2$ then, $R_i \setminus (R_j \cup R_{j'}) = \{v, v'\}$ such that $entry(R_i) = v$, $L(v) = \textit{Exclusive Split}$ (resp., $L(v) = \textit{Parallel Split}$), $\{(v, entry(R_j)), (v, entry(R_{j'}))\} \subseteq E$, $exit(R_i) = v'$, $L(v) = \textit{Exclusive Join}$ (resp., $L(v) = \textit{Parallel Join}$), $\{(exit(R_j), v'), (exit(R_{j'}), v')\} \subseteq E$;

   b - *loop case*[1] - $submax(R_i) = \{R_j, R_{j'}\}$, i.e., $|submax(R_i)| = 2$ then, $R_i \setminus (R_j \cup R_{j'}) = \{v, v'\}$ such that $v = entry(R_i)$, $L(v) = \textit{Exclusive Join}$ (resp., $L(v) = \textit{Parallel Join}$), $\{(v, entry(R_j)), (exit(R_{j'}), v)\} \subseteq E$, $v' = exit(R_i)$, $L(v) = \textit{Exclusive Split}$ (resp., $L(v) = \textit{Parallel Split}$), $\{(exit(R_j), v'), (v', entry(R_{j'})\} \subseteq E$, in such a case R_j is called the *forward sub-region* of R_i, denoted by $forward(R_i)$ and $R_{j'}$ is called the *backward sub-region* of R_i, denoted by $backward(R_i)$;

   c - *sequence case* - $submax(R_i) = \{R^i_1, R^i_{m_i}\}$ such that $entry(R_i) = entry(R^i_1)$, $exit(R_i) = exit(R^i_{m_i})$, and for every $1 \leq j < R^i_{m_i}$ we have $(exit(R^i_j), entry(R^i_{j+1})) \in E$.

   Finally, if $L(entry(R_i)) = task$ (resp., $L(exit(R_i)) = task$) and there exists a unique $v \in V \setminus R_i$ such that $(v, entry(R_i)) \in E$ (resp., $(exit(R_i), v) \in E$) we may extend R_i to

---

[1] We must observe here that the loop case might be cumbersome for the general task of determining whether a given trace fits the impact constraint. Indeed, apart from the trivial case of a loop containing only tasks with no impact in any dimension, every loop will, in a finite number of steps, overwhelm one of the components of the input impact vector. This is taken specifically into consideration when we shall formulate the generalization of Problem 1 to Problem 2.

the extended region $R'_i = R_i \cup \{entry(R_i)\}$ (resp., $R'_i = R_i \cup \{exit(R_i)\}$).

Moreover, we assume every loop case to always have both a forward and a backward non-empty region. It is straightforward to write the rules for the cases in which one of them is missing so we omit it, another, less elegant solution, would be to add a dummy task in place of the missing region. We can use the very same arguments to fix the branching cases that feature one empty branch.

From now on we will call a `BPMN` $G = (V, E, L)$ structured in SESE regions simply a *structured* `BPMN`. In a structured `BPMN`, it is easy to see that $\mathscr{R} = \{R_1, \ldots, R_m\}$ is unique and may be organized as a rooted tree $RT = (\mathscr{R}, E)$ such that $(R_i, R_j) \in E$ if and only if $R_j \in submax(R_i)$, we will call such an object the *region tree* of G.

Below I present a diagram denoted as G_1 and G_2. Diagram G_1 is structured in Single Entry Single Exit (SESE) form but Diagram G_2 is not. I assume that the input process is initially provided in Single Entry Single Exit form (SESE) as described in [16].

Informally, a Business Process Model and Notation (`BPMN`) diagram in SESE form adheres to a specific structure. For every gateway-split node, there exists a unique corresponding gateway-join node. This relationship ensures that the region formed by such a pair of gateway nodes has precisely one incoming edge and one exit edge. Conversely, a `BPMN` diagram without SESE form may have a situation where one incoming edge has more than one exit edge. The representation in SESE form is trace equivalent[2], and it can always be obtained from a generic `BPMN` diagram. `BPMN` diagrams, being special cases of Petri nets, possess inherent properties that facilitate such transformations. Moreover, numerous optimized tools exist for addressing the unfolding problem, which allows to transformation of any `BPMN` into equivalent SESE-based `BPMN` representation in polynomial time [18].

3.2.2 Compliance of Business Process

Compliance checking determines when at least one trace (or all traces, in the case of the dual version) of a business process can be executed while adhering to a set of imposed

---

[2] Two `BPMN` diagrams are trace-equivalent if they produce the same traces, see [44].

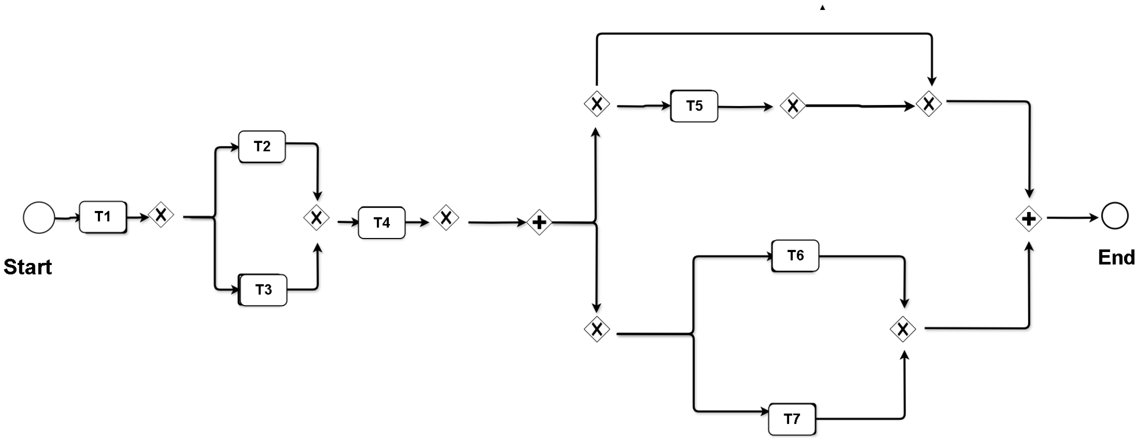


Figure 3.3: G_1 BPMN structure in the Single Entry Single Exit (SESE) form.

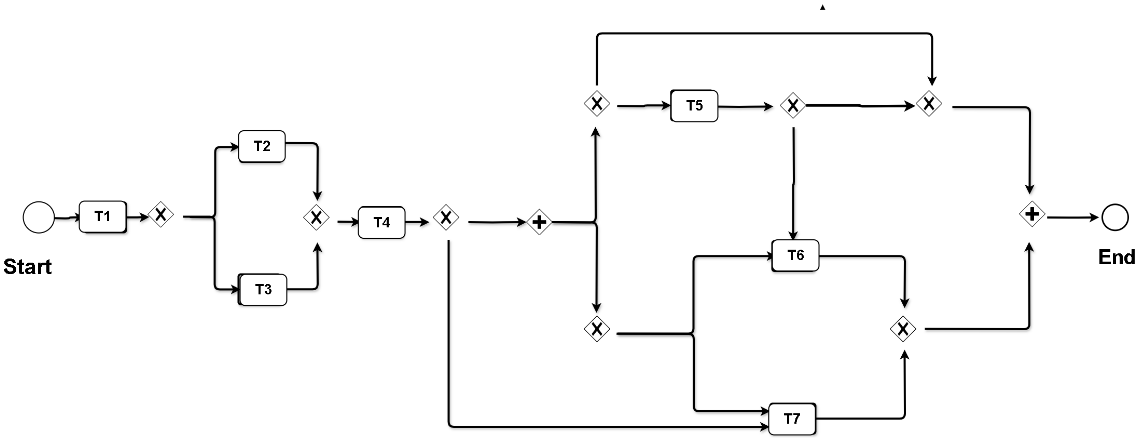


Figure 3.4: G_2 BPMN structure without in Single Entry Single Exit (SESE) form.

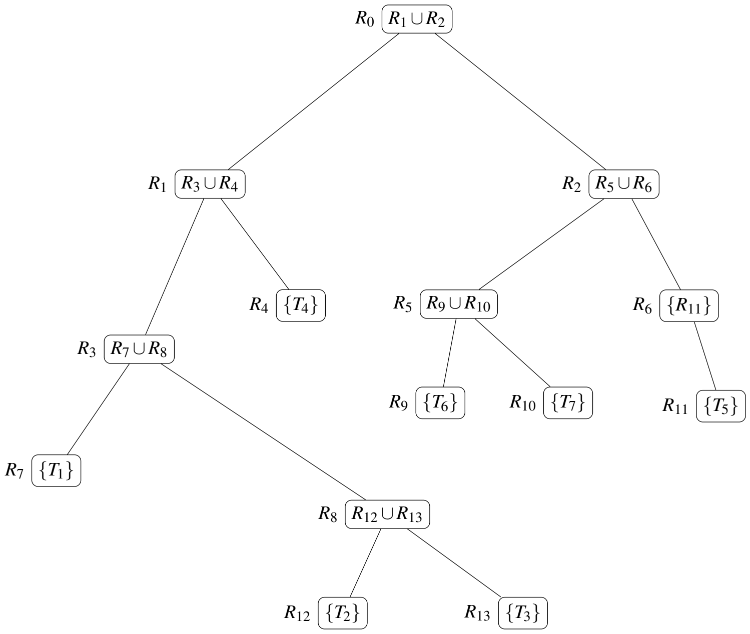


Figure 3.5: The region tree of the BPMN depicted in Figure 3.3

constraints. The side effects of non-compliance, when referred to as impact, are substantial. These procedural inconsistencies may lead to audit issues, legal complications, or violations of other regulatory standards. For example, many legal boundaries are established as limits on emissions or the consumption of scarce resources. Furthermore, numerous standards, such as ISO 14064 (for reporting and limiting the environmental impact of industrial processes), ISO 14083 (which governs emissions in transportation and logistics), and ISO 14067 (which limits product emissions), have significant implications in the marketplace.

Many customers, for legal or practical reasons, may require their supply chain partners to comply with specific standards. Failure to meet these standards may grant the customer the right to terminate the contract. Additionally, process variations leading to non-compliance may result in the overuse of resources or negatively affect the quality of the final product or service. Consequently, most process failures in agricultural or industrial production lead to significant environmental impacts or other detrimental effects, which may also contribute to climate change.

More broadly, there are instances where the impact referred to does not directly relate to emissions or resource consumption, such as in cases involving *economic impact* or *workload impact*. In this study, I adopt an approach that unifies these various forms of impact and refers to them collectively as *impact*. I acknowledge that there may be multiple types of impact, necessitating a complex, simultaneous, multi-dimensional evaluation.

3.3 Representation of Impact and Compliance of Impact-aware `BPMN` Process

Given a `BPMN` $G = (V, E, L)$, an *impact function* $\mathscr{I}$ over G is a function $\mathscr{I} : V \to \mathbb{N}^k$ with $k \geq 1$ such that for every $v \in V$, we have $\mathscr{I}(v) = \overrightarrow{0}$ if $L(v) \neq \texttt{task}$, i.e., for the sake of simplicity and without losing generality, only `task` elements may have impacts greater than 0 for some components.

Let $T = \{v \in V : L(v) = \texttt{task}\}$. We borrow the definition of *conformant execution* from [45], that is, a conformant execution of G is any word $t_1 \ldots t_h$ in T^+ that may be executed in G with fitness equal to 1. We define the $\mathscr{I}$ impact of $w = t_1 \ldots t_h$ as

$$\mathscr{I}(w) = \sum_{1 \leq i \leq h} \mathscr{I}(t_i).$$

Given a bound $b \in \mathbb{N}^k$, we say that w is *impact-compliant* with respect to $(G, \mathscr{I}, b)$ if and only if

$$\mathscr{I}(w) \leq b.$$

For the sake of brevity, in the following, we will assume that G, b, and $\mathscr{I}$ are clear from the context, and we will simply state that w is impact-compliant.

The core notion of our strategy is to evaluate the impact of the business process life cycle. For clarity, consider the grape production process, which starts with site selection and finishes with harvesting. To illustrate this, we choose the agricultural sector, specifically grape cultivation.

Problem 1. *Given a `BPMN` G, an impact function $\mathscr{I} : V \to \mathbb{N}^k$, and an impact bound b, determine whether or not there exists an execution $w \in T^+$ that is below the value for each component of the bound.*

It is not possible to define a meaningful *exact* dual problem of Problem 1. That would require that *every* execution of a given `BPMN` results below the value of each component of the input bound. If this is the case, the problem becomes trivial to answer. If the `BPMN` contains one loop, and the tasks within the loop sum to a non-null vector, we shall say that the `BPMN` does not satisfy such a formulation of the dual problem. Conversely, when no loop exists or when all the existing loops have internal tasks summing up to the null vector, the problem could be solved.

Therefore, we need to formulate the problem by letting it depend on two aspects:

- The number of executions for each loop;
- The choices that we consider to be fixed for the XOR-splits.

We can consider the loop case as removed for this specific configuration of the computational problem, as we can transform the process by devising the exact number of repetitions of each loop, and eliminate, therefore, the loop itself. The result of this rewriting will be a business process with no loop cases and with part of the alternatives in XOR-splits eliminated.

Problem 2. *Given a loop-free* `BPMN` *G, an impact function $\mathscr{I} : V \to \mathbb{N}^k$, and an impact bound b, determine whether or not all the executions in T^+ are below the value for each component of the bound.*

Before delving into the discussion on computational complexity, it is relevant to differentiate between planned and mined process models. A planned process model entails the deliberate incorporation of compliance principles from the outset, essentially embodying the concept of *process by design*. This, rather naturally, concerns regulatory compliance as well as impact compliance. On the other hand, mined process models are derived from existing traces using process mining techniques, often necessitating subsequent adjustments (in particular regarding the deviations) to ensure compliance with given constraints. In the context of this research, which revolves around finding traces with specific structures below a constraint and impact measures, we are analyzing the case of design elements as well as mined ones. There is a significant difference in terms of *quality* of the structure, with the mined ones being markedly more random than the designed. Therefore, as we point out in Section 3.5, the performance of the algorithms depends on several factors, analyzed in Section **??**, which can be much higher in mined than in designed processes.

3.4 Solving Impact-Aware Compliance Problems

In order to devise a formal method to solve Problem 1, we need to introduce the notion of *dominance*. Given two vectors $\overrightarrow{n}, \vec{n}' \in \mathbb{N}^k$, we say that $\vec{n}'$ *dominates* $\vec{n}$, written $\vec{n} \leq \vec{n}'$, if and only if for every $1 \leq i \leq k$ we have $\vec{n}[i] \leq \vec{n}'[i]$. If neither $\vec{n} \leq \vec{n}'$ nor $\vec{n}' \leq \vec{n}$, we say that $\vec{n}$ and $\vec{n}'$ are incomparable, written $\vec{n} \not\leq\geq \vec{n}'$.

Before going into the formal proofs of correctness and identifying the computational complexity of the problem, let us present informally the method underlying Algorithm 1. Algorithm 1 solves Problem 1 in the following way. First, it generates the region RT of the input structured `BPMN` G. This is achieved by performing a Depth First Visit of G and, since each node in G has a degree at most 3 we have that this task is performed in linear time w.r.t. the number of nodes in G. Then, at line 2 Algorithm 1 computes the Breadth First Visit of RT starting from its root, but the result, i.e., the list of regions/nodes of RT encountered during the visit, is returned in reverse order. Since the number of nodes in RT is less or equal than $|V|$ we have that also this operation has time complexity $\mathcal{O}(|V|)$. The matrix $M_I \in \mathbb{N}^{k \times m}$ that is initialized to 0 in each cell, at line 3 of Algorithm 1, at the end will keep the impact of the regions considered by Algorithm 1. By the property of the Breadth First Visit we have that for each $1 \leq i \leq m$ the index j of any proper sub-region $R_j \subset R_i$ satisfies $j < i$. This property is crucial because in the main loop of Algorithm 1 for determining the impact $M_I[:, i] \in \mathbb{N}^k$ of a region R_i it suffices to use just the impact of the region in $submax(R_i)$ assuming that the impacts for all the regions $submax(R_i)$ have been already defined. Then the algorithm proceeds from $i = 1$ to m computing a *minimal* impact for R_i according to its type defined in Section 3.2.1:

1. if R_i is a sequence case, lines 5-8, the impact is given by simply adding up the

impacts of its maximal subregions and the impact for the tasks which are newly introduced in R_i;

2. if R_i is a parallel case, lines 9-11, we simply add the impacts of its two maximal subregions;
3. if R_i is an exclusive case, let R_{i_1} and R_{i_2} its two maximal proper sub-regions, three possibilities may arise depending if $M_I[:, i_1] \leq M_I[:, i_2]$, $M_I[:, i_2] \leq M_I[:, i_1]$, and $M_I[:, i_1] \not\lesseqgtr M_I[:, i_2]$. The first two of such possibilities are symmetric, lines 17-20, and deal with the fact in which $M_I[:, i_1]$ and $M_I[:, i_2]$ are comparable, and the algorithm simply chooses to assign to $M_I[:, i]$ the values for the the less impactful of the two. The third case, lines 21-23, deals with the fact in which $M_I[:, i_1]$ and $M_I[:, i_2]$ are incomparable, that is, there exist two distinct indexes $1 \leq j, j' \leq k$ such that $M_I[j, i_1] < M_I[j, i_2]$ and $M_I[j', i_1] > M_I[j', i_2]$. In this and only this case, the algorithm non-deterministically guesses one between i_1 and i_2 and assign it to $M_I[:, i]$. More in detail, in Algorithm 1, we introduce the term standard non-deterministic operator GUESS to address scenarios involving XOR-splits. The concept revolves around selecting a region with a lower impact measure when one region's impact dominates the other. However, if the impacts are incomparable, the algorithm employs an *oracle* to make a non-deterministic selection. The main point of the XOR-split treatment lies in identifying a dominant region, where all components of the impact vector overwhelm those of the other region. In such cases, we opt for the dominant region. This approach serves to verify if a solution exists, as the failure of the dominated region to meet the bound automatically indicates the absence of a solution. Conversely, when no dominating alternative emerges from the split, we must explore all possibilities. The oracle at line 22 recursively selects alternatives for each split. Consequently, the number of generated alternatives depends solely on the nested splits within the `BPMN`, thereby ensuring scalability as we see later. When only one component exists, determinism prevails, as the domination relation forms a total order, eliminating the need for guesswork. This underscores the deterministic nature of the algorithm in scenarios devoid of competing alternatives.

At the end of the for loop of lines $4-24$, the algorithm returns $\perp$ (i.e., *fail*) if there exists a component $1 \leq h \leq k$ for which $M[h,m] > b[h]$. On the other hand, if the algorithm reaches the line 26 it succeeds since $M[:,m] \leq b$. Let us notice that lines $22-23$ may be executed only if $k > 1$, since all the elements of $\mathbb{N}^1$ are always pairwise comparable. Then, we can prove the following results.

Theorem 3.4.1. *Algorithm 1 correctly decides Problem 1 for all instances of input, and terminates in a finite number of steps.*

Proof: We subdivide the proof into three parts. At first (A) we show that any input will provoke the end of computation in a finite number of steps. We then (B) show that the algorithm is correct, namely that it decides the exact output when given an input, and finally (C) we shall consequently prove that the algorithm is correct, complete, and terminates in a finite number of steps.

(A) Assume by contradiction that one input could provoke infinite looping. This is due to the steps in the algorithm can be produced by the computation of Step 1, Step 2, or by cycles 4-23. In turn, if it is caused by cycles 4-23, it could be intrinsic to the cycle, provoked by inner cycles, that do not exist, or generated by changes in cycle control variables within the cycle itself. Now, Step 1 is a Depth-first visit, and therefore terminates for any input, Step 2 is the reverse of a Breadth-first visit, and again it cannot be an infinite loop. Again on cycles 4-23, the cycle depends on variables i and m that are never modified within the cycle itself. Therefore point (A) is proven.

(B) Let us consider a triple that constitutes the input of the algorithm, formed by a structured `BPMN` $G = (V,E,L)$, an impact function $\mathscr{I} : V \to \mathbb{N}^k$, a bound $b \in \mathbb{N}^k$. We may have two possible outcomes determined by the traversing of the region tree while respecting the input constraint vector: either there is a way to traverse the tree without incurring a value of the impact that does not generate a total impact greater than the impact of the constraint vector, or this is not the case.

The method is based on the transformation of the `BPMN` given in input into its Region

Tree, where, however, we revert the order of the result of a Breadth-First visit of the tree itself. Now, the input to the cycle 4-23 is formed by m regions in inverse order $1 \ldots m$ with respect to the start and end node in the `BPMN`. We set a matrix M_I (as many rows as the impact dimensions, and as many columns as the number of regions) to the null matrix. We then enter the cycle and non-deterministically visit the region tree, in different cases depending on the single cases.

Summarising the cases not involving a split:

- When we have a sequence in the region, then we recursively compute the set of subregions that are maximal, namely those subregions that are connected directly to the region itself in the graph we obtained. We then compute the set of tasks that belong to the sequence and sum up the matrix computed at that stage to the sum of the task impacts for each component, at the column corresponding to the visited region. This computation requires us to visit backward the `BPMN` towards the current position in the specific region until we reach the regions in the level before.
- When we have parallel subregions we sum up the two subregions pointing to the visited one in the same way of the sequence.
- When we have a loop in the region, we sum up the forward region matrix to the output matrix.

If a split is actually present, then we have a non-deterministic evolution of the algorithm. The essence of the split treatment consists in determining whether there is a *dominant* region pointing to the split, namely a region in which all components of the impact vector are greater than the components in the other region split region. If this is the case, we choose the *dominated* region. If there is no solution, the fact that the dominated region is not making the bound respected proves it automatically, for the other split alternative would be worse.

When, however, the split does not generate one dominating alternative, then we have to explore all the alternatives. We, therefore, ***guess*** (at line 22) which region to identify that belongs to the solution or is anyhow best choice locally. The oracle at line 22 is a basic nondeterministic one, since it chooses recursively for any split, and therefore generates a number of alternatives that only depend on the number of splits nested into each other, in turn, bound by the number of splits in the `BPMN`.

Clearly, when one component only exists, we shall have no alternatives to choose from, for the domination relation becomes a total order, and therefore we have no case in which we need to guess, and the algorithm will be deterministic.

Considering the possible inputs (one case in which the specific `BPMN` offers one alternative path that satisfies the bound, and one case in which no alternatives like this exist) we can conclude, by the above reasoning that, in the case in which we have an impact vector with at least two components:

1. Whenever there is no alternative satisfying the bound the oracle will explore non-deterministically each alternative;
2. Whenever there is an alternative satisfying the bound the oracle will guess one of these alternatives;
3. Whenever there is a unique way of obtaining the final state in the `BPMN` the cycle will determine it;
4. Whenever there is not a unique way, but there is no split, then the cycle simply explores the whole region tree.

When, instead the component is only one, we do not need the oracle at all, so points 2 and 1 are in no way different from the other pathways through the `BPMN`. The above reasoning proves point (B).

(C) Since all cases are covered by the reasoning on point (B) and we know that the algorithm terminates, as proven in point (A), we can thus conclude the claim of the theorem is proven.

It is rather easy to accomplish the following operation. When Algorithm 1 non-deterministically decides that it is possible to satisfy the input bound, it does so by determining one *minimal* trace[3]. We may be tempted to reduce, in a trivial way, the computation tree, while avoiding to compute any further solution when one is found. This needs to change the algorithm structure. It is moreover clear that this will never provide an actual advantage to the computation in the worst case, and in general will never change the computational cost in the average case as well, as this problem is assimilated to any graph exploration one, where the above limits to heuristics employed to reduce the cost have been deeply investigated back in the Seventies of the past century. Therefore, if we employ the current approach, we complete the computation having the possibility of exhibiting a minimal solution. We then formulate a computation problem for the minimal trace in the definition of Problem 3.

Problem 3. *Given a structured* BPMN *G, an impact function $\mathscr{I} : V \to \mathbb{N}^k$, compute a trace of G that minimizes the impact.*

The solution of Problem 3 lies in the search for a trace that results minimal on each component of the impact vector. Although this may appear different from solving Problem 1 we can prove that it only consists in searching based on Algorithm 1. More specifically, algorithm 2 solves Problem 3 using Dichotomic Search.

Let max_i be the value for a loop-free path in $\mathscr{G}$ that maximizes impact component i, and let max be the maximum of such values over all components. It is easy to prove that Algorithm 2 returns a minimal impact vector after $\mathscr{O}(|V|) + (k-1)\log_2(max)\text{NPTIME}(|V|)$ steps. Let us assume $k = \mathscr{O}(|V|)$ and $max = \mathscr{O}(2^{|V|})$, which are reasonable assumptions since components are always of the order of the nodes (if not of a smaller one), and due to the binary encoding, the values in such components may be exponential in the number of nodes. Finally, under the previous assumptions, we can say that Problem 3 can be solved in NPTIME$(|V|)$ which has the same complexity of Problem 1.

Theorem 3.4.2. *Algorithm 2 correctly solves Problem 3 for all instances of input in a finite number of steps.*

---

[3] A trace is minimal when it is not dominated by any other trace.

The proof of the above theorem is straightforward and thus omitted for the sake of conciseness.

Thanks to Theorem 3.4.1 we have proof of correctness. The complexity upper bound for Problem 1 is claimed in Theorem 3.4.3.

Theorem 3.4.3. *Problem 1 with* $k = 1$ *may be solved in linear time* $\mathscr{O}(|V|)$*, otherwise, if* $k > 1$ *the same problem belongs to the complexity class NP.*

Proof: In case of one component impact, Algorithm 1 actually determines the best alternative among the possible ways of traversing the `BPMN`. When this is not the case, it is possible that all the alternatives that satisfy the bound, if any exist, are sub-optimal, when at least one split condition is guided by choice that is by no means with a dominant vector.

In other terms, independently of the existence of one or more than one component in the impact vector, Algorithm 1 *decides* whether all alternatives needed to be tested respect the bound.

It is evident that the decision on the bound is maximized by the number of regions and the number of components, and therefore its cost is $O(|V|)$ in the number of vertices in the `BPMN`. Conclusively, the algorithm oracle guesses one admissible traversing alternative and then concludes for it satisfying the bound in polynomial time. This proves the claim.

Differently from Problem 1, Problem 2 *can be solved polynomially*. Assume that we modify the behavior of Algorithm 1 in a way that, similarly to the solution of Problem 3, determines a *maximal* trace. Obviously, if this change is performed, we have a method that, when applied to a scalar version of Problem 3 determines one *maximum* trace. Now, in order to solve Problem 2 we need to prove that all traces remain below a given limit. We can solve the problem by considering each component of the bound vector separately, computing the maximum trace for that component, and if one of these components does not satisfy the bound, we conclude that the process is not *always* below the bound. Algorithm 3 implements the above-defined strategy.

Theorem 3.4.4. *Problem 2 may be solved in time* $\mathcal{O}(k|V|)$.

Proof: Contrary to *shortest path*, the longest path belongs to the class of $NP-hard$ problems, for generic graphs. However, it is polynomially solvable on deterministic machines, being linear in the number of vertices, for acyclic direct graphs. Now, since a loop-free `BPMN` is an acyclic direct graph, the result follows.

Based on the schema of Algorithm 1 we can compute the solution in parallel for all components of the impact function and bound. This approach is implemented in Algorithm 3.

We can look for a proof of actual reducibility, in order to provide a lower bound to the complexity of Problem 1, and consequently, of Problem 3. In Definition 3.4.5 we introduce the notion of *impact of a trace*, employed in the solution by Algorithm 3.

Definition 3.4.5. *Given a structured* `BPMN` *process* $G = (V, E, L)$ *and an impact function* $\mathscr{I} : V \to \mathbb{N}^k$, *the impact of a trace* w *in the set* T^+ *is defined as the sum of the impacts of the tasks in the trace. Let* $w = (v_1, v_2, \ldots, v_n)$ *be a trace in* T^+, *where* v_i *represents a task in the* `BPMN` *process. The impact of the trace* w *is given by:*

$$Impact(w) = \sum_{i=1}^{n} \mathscr{I}(v_i)$$

Here, $\mathscr{I}(v_i)$ *represents the impact of task* v_i *according to the impact function* $\mathscr{I}$. *The impact of the trace is the sum of the impacts of all tasks in the trace.*

We are now ready to discuss about the reducibility of Problem 1. In order to prove that Problem 1 is NP-complete, as stated in Theorem 3.4.9 we need to provide a polynomial reduction from a known NP-complete problem towards Problem 1. To do so, we introduce the definition of the well-known problem of *distinct partition*, in Definition 4.

Problem 4. (Distinct Partition) *Given a set of natural numbers* $S = \{n_1, \ldots, n_m\}$ *decide whether or not there exists a partition* (S_1, S_2) *of* S *such that* $\sum_{n \in S_1} n = \sum_{n \in S_2} n$.

As formulated by Korf in [28], Problem 4 is actually NP-complete. We recall this in Theorem 3.4.6.

Theorem 3.4.6. *Distinct Partition (Problem 4) is NP-Complete [28].*

There exists a simple LOG-SPACE reduction from Distinct Partition to Problem 1 for $k \geq 2$. We show this in Lemma 3.4.7.

The Distinct Partition problem is a well-known problem in computer science. It involves dividing a set of positive integers into two subsets such that the sums of integers in the two subsets are equal, and each integer appears in exactly one subset.

Lemma 3.4.7. *Problem 1 with impact vector dimension at least 2 is NP-hard.*

Proof: Let $N = \{n_1, \dots n_m\}$ be our instance of Distinct Partition, we build a structured `BPMN` $G = (V, E, L)$ and its impact function $\mathscr{I} : V \to \mathbb{N}^2$ as follows:

1. $V = \{v_0, \dots, v_{4m+1}\}$;
2. $E = \{(v_0, v_1), (v_{4m}, v_{4m+1})\} \cup \{(v_{4i+1}, v_{4i+2}), (v_{4i+1}, v_{4i+3}), (v_{4i+2}, v_{4i+4}), (v_{4i+3}, v_{4i+4}) : 0 \leq i < m\}$;
3. $L(v_0) = start$, $L(v_{4m+1}) = end$, and for each $0 \leq i < m$ we have $L(v_{4i+2}) = L(v_{4i+3}) = task$, $L(v_{4i+1}) = \mathit{Exclusive\ Split}$, and $L(v_{4i+4}) = \mathit{Exclusive\ Join}$;
4. $\mathscr{I}(v_0) = \mathscr{I}(v_{4m+1}) = \overrightarrow{0}$, and for each $0 \leq i < m$ we have $\mathscr{I}(v_{4i+2}) = \begin{bmatrix} n_{i+1} \\ 0 \end{bmatrix}$, $\mathscr{I}(v_{4i+3}) = \begin{bmatrix} 0 \\ n_{i+1} \end{bmatrix}$.

Finally, let $B = \frac{\sum_{i=1}^{m} n_i}{2}$, we put $\overrightarrow{b} = \begin{bmatrix} B \\ B \end{bmatrix}$.

We should now argue that we have a computational problem, Problem 3, that corresponds (in terms of abstract polynomial reduction of decision problems to computation problems) to Problem 1. Therefore, the reduction employed in the proof of Lemma 3.4.7 is not necessarily applicable to the computation case.We can observe that the Distinct Partition Problem involves finding a combination of values from the input multiset, which is formed by two sub-multisets that each sum to half of the total sum of the entire multiset. If we determine a non-minimal solution in polynomial time on non-deterministic machines, we can do the same computation for minimal solutions, due to the combinatorial nature of the problem itself. This is therefore the intrinsic correspondence determined by the log-space reduction of Lemma 3.4.7. Straightforwardly, the problem of determining one minimal solution of the Distinct Partition Problem is NP-hard, and therefore we can prove Lemma 3.4.8, whose proof is again a trivial consequence of the above reasoning and it is omitted

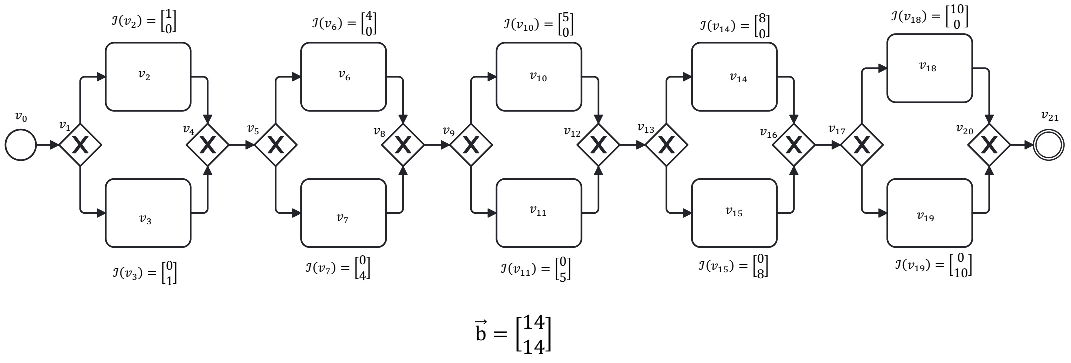


Figure 3.6: An instance of the reduction used in Theorem 3.4.9 on the instance $S = \{1,4,5,8,10\}$ of the Distinct Partition problem.

for the sake of conciseness.

Lemma 3.4.8. *Problem 3 with impact vector dimension at least 2 is NP-hard.*

As you can see in Figure 3.6, $S = \{1,4,5,8,10\}$ specifies one given case of the Distinct Partition problem that the reduction is applied. If we wish to map S, we need to consider the input sum, that is 28, an even number, which means that in principle it would be possible to solve the problem (with odd numbers it is excluded a priori). We assign a bound vector $\overrightarrow{b} = \begin{bmatrix} 14 \\ 14 \end{bmatrix}$ and construct a `BPMN` (as in the mentioned Figure) as provided in Lemma 3.4.7.

Based on Lemma 3.4.7 and Theorems 3.4.1 and 3.4.3 we can derive Theorem 3.4.9.

Theorem 3.4.9. *Problem 1 with impact vector dimension at least 2 is NP-Complete.*

Quite naturally, when we have a single-valued impact measure, we can solve the problem polynomially, as stated in Theorem 3.4.3. This is due to the total ordering of the single-valued impact vectors. On the other hand, let us assume that we convert Algorithm 1 into a deterministic version. The oracle behavior can be simulated by a depth-first visit to the tree formed by the *independent* XOR-splits appearing in the business process. This means, in turn, that when the independent XOR-splits are limited, we may have a polynomial treatment of the problem on deterministic machines. This concept is expressed in Theorem 3.4.13. Let us now formally account the notion of independent, and nested splits.

Definition 3.4.10. *Two XOR regions $R1$ and $R2$ in T are independent if and only if their least common intersection in T is not in an XOR region.*

When two regions are not independent with each other we say that they are *nested*. Algorithm 4 determines the maximum number of independent XOR-splits in a `BPMN`.
The way in which the aforementioned methods work are claimed in Theorems 3.4.11 and 3.4.12, whose proofs are straightforward consequences of Definition 20, and therefore omitted.

Theorem 3.4.11. *Given a structured `BPMN` $G = \langle V, E, L \rangle$ Algorithm 4 correctly computes the number of independent XOR-splits in the input graph in $O(|V|)$.*

Theorem 3.4.12. *Given a structured `BPMN` $G = \langle V, E, L \rangle$ Algorithm 5 correctly computes the number of independent XOR-splits in the input graph in $O(|V|)$.*

We can now formulate a general result on the actual deterministic complexity of Problem 1, that is solved by linearizing non-deterministic Algorithm 1, in Theorem 3.4.13.

Theorem 3.4.13. *Problem 1 with impact vector dimension at least 2, can be solved in $O(|V| \cdot 2^{k \cdot h})$ on deterministic machines, where k is the number of maximum number of independent XOR and h is the maximum number of nested XOR.*

Consequently to Theorem 3.4.13 we can formulate a method to heuristically solve Problem 1 in polynomial time.

To do so, we need to assume that we have some preliminary measure of the computational effort we intend to consider acceptable. This will be measured by a *polynomial payout*, namely an integer number p whose meaning is that we shall consider acceptable to solve a problem in $O(n^p)$ where n is, in this case, the number of vertices in the graph (or equivalently, the number of regions in the region tree).

We first compute the maximum number k of independent XOR-splits and the maximum number h of nested XOR-splits contained in a `BPMN`. Then, we compare the obtained numbers with an input p which is the maximum polynomial payout we accept. When we have that:

$$p \geq \frac{k \cdot h}{\log |V|}$$

If the above holds, we have, as a consequence of Theorem 3.4.13, that the complexity of deterministic solution of Problem 1 would be $O(|V|^{p+1})$.

$$p \geq \frac{k \cdot h}{\log |V|} \quad \rightarrow \quad p \cdot \log n \geq k \cdot h \quad \rightarrow \quad \log n^p \geq k \cdot h \quad \rightarrow$$

$$2^{\log n^p} \geq 2^{k \cdot h} \quad \rightarrow \quad n^p \geq 2^{k \cdot h}$$

Based on the above reasoning we can claim the following result.

Theorem 3.4.14. *Given a* `BPMN` *with impact constraints* $G = \langle V, E, L \rangle$*, the deterministic solution of Problem 1 runs in* $O(|V|^{p+1})$ *time when* $p \geq \frac{k \cdot h}{\log |V|}$ *holds, where* k *is the maximum number of independent XOR-splits and* h *is the maximum number of nested XOR-splits in* G*.*

3.5 Experimental Evaluation and Analysis

In this section, we analyze the the behavior of the algorithms summarized in Table 3.3. In particular, we will focus on Algorithm 3 for the polynomial side of the theoretical complexity, and on Algorithm 1 and Algorithm 2 as $\mathscr{NP}$-hard representatives, in the context of a synthetic dataset of randomly generated business processes and impact vectors.

We conducted a focused, small-scale experiment to study how both the structure of processes and the properties of the impact vectors associated with their tasks affect the computational time for the aforementioned algorithms. The experiment was performed on a system equipped with an *Intel Core i9-10980HK CPU clocked at 2.40GHz, 32GB of RAM*, and running *Ubuntu 22.04.3 LTS* as the operating system. We generated random business processes with varying control flow complexities, such as different numbers of nested XORs (MNXN) and independent (MIX) XORs.

Processes are randomly generated as follows. Initially, we start with a seed string _, representing a single task. Iteratively, we replace underscores with one of three possible structures:

* *XOR split*: *(_ ^_)*
* *Parallel split*: *(_ ||_)*
* *Sequential*: *(_ , _)*

Then, we replace the remaining underscores with task labels (T1, T2, etc.). The replacement process continues until the desired complexity, in terms of MNXN and MIX, is achieved. Our objective is to generate a specified number of `BPMN` processes with constraints on these XOR counts. We generated 10 `BPMN` processes for each combination of MNXN and MIX, ranging from 1 to 10. This resulted in *a total of 1,000 unique processes (10 × 10 × 10)*. The generation process employs *weighted random choices* to determine which structure to use for each replacement, allowing for controlled variability in the generated processes. The generation process follows these general steps:

a) *Probabilities for generation*: Defines the probabilities for generating different types of structures in the `BPMN` processes. Alternatively, *'None'* can be used for equal probabilities.

b) *Process replacement*: Specifies how many times replacements are allowed in the process generation.

c) *Target number of processes*: The desired number of unique `BPMN` processes to generate.

d) *Number of trials*: The maximum number of attempts to generate one of the required processes.

e) *XOR constraints*: the MNXN and MIX parameters that each generated process must meet.

For instance, the `BPMN` diagram provided in Figure 3.7, generated randomly, comprises *66 tasks*, with *MNXN equal to 9* and *MIX equal to 2*. Though in practice, it is unlikely that a business process has more than forty tasks, we can show that the number of tasks is not so influential towards the computational complexity, that actually depends on the maximum number of independent (MIX) and nested (MNXN) XORs in the process model itself.

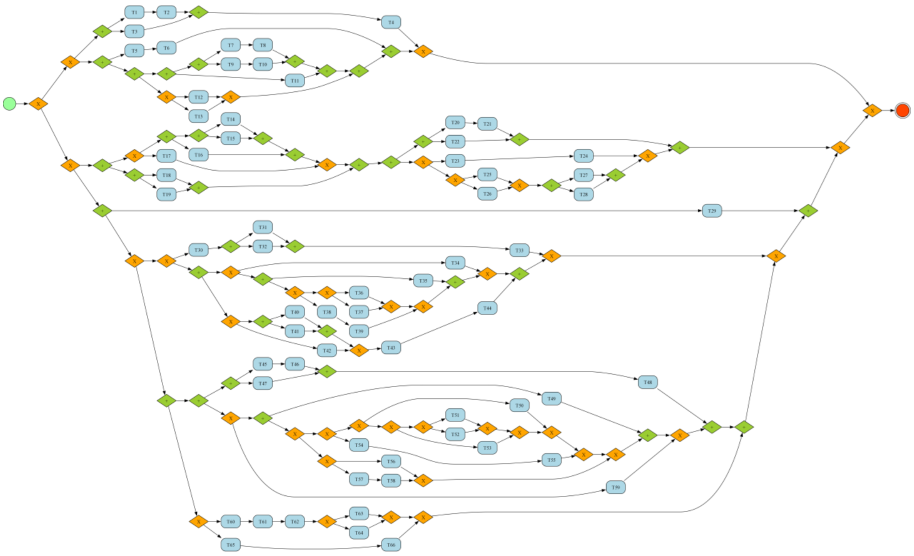


Figure 3.7: Synthetic BPMN Model with 66 Tasks, MNXN=9, and MIX=2.

Structuring the generation in this manner ensures that the synthetic `BPMN` processes meet specified constraints and provide feedback on the generation process. We generate processes for combinations of MNXN and MIX, with each ranging from 1 to 10. For each combination of MNXN and MIX, we generate 10 different business processes. For example, we generate 10 processes for combinations such as (MNXN=1, MIX=1), (MNXN=1, MIX=2), ..., up to (MNXN=10, MIX=10).Thus, we have 100 possible combinations (10 MNXN values × 10 MIX values), and for each combination, we generate 10 processes. This experiment involved 100 × 10 different business processes, resulting in 1,000 unique random business processes, all fulfilling the SESE structure.

For the generation of impacts associated to the tasks, we created random vectors with specified dimensions and optional modifications. By default, the vectors are generated randomly with values between 0 and 1 for each dimension, but various *modes* can be specified to alter the vectors. We used *six different modes* to generate these vectors, each mode using values between 0 and 1 for each dimension, and each providing a unique distribution of values. These *modes* make use of the bagging technique [**?**] which, in its essence, involves creating multiple subsets of the original dataset through random sampling with replacement. In our case, after a round of bagging over the components of a vector, we have partitioned our such components in two subsets the one extracted, with some of them possibly repeated, and a set of excluded components. For instance, consider a vector of dimension 5, with components ranging from 0 to 4. After a round of bagging, it might occur that components $\{1,2,4\}$ are extracted, with components 1 and 4 repeated 2 times each, and component 2 repeated once. Consequently, $\{0,3\}$ becomes the set of excluded components. In the following, we describe how a single vector is generated according to each mode:

- *random*: Generate real values between 0 and 1 uniformly at random.
- *bagging_divide*: After random generation and a round of bagging, divide each component by a power of 10 based on its frequency in the bagging round.
- *bagging_remove*: After random generation and a round of bagging, set the components excluded by the bagging round to zero.
- *bagging_remove_divide*: Similar to *bagging_remove*, but also divide each selected component by a power of 10 based on its frequency in the bagging round.
- *bagging_remove_reverse*: Similar to *bagging_remove*, but the components selected by the bagging round are set to zero (i.e., the reverse of *bagging_remove*).
- *bagging_remove_reverse_divide*: Similar to *bagging_remove_reverse*, but perform an additional round of bagging on the non-zero components. Scale these components by a power of 10 based on their frequency in this additional bagging round.

These impact vectors represent the *multi-dimensional effects* or outcomes of the tasks within each process. The *dimension* of each vector corresponds to the number of *impact factors* being considered (ranging from 1 to 10 in our experiments). Before showing the experimental results, we may argue about the vector generation modes employed in our study. The modes can be broadly categorized based on the density of the vectors they produce. In particular, *random*, *bagging_divide*, *bagging_remove*, and *bagging_remove_divide* generally generate dense vectors, meaning many components are different from 0. In contrast, *bagging_remove_reverse* and *bagging_remove_reverse_divide* typically produced sparse vectors.

In our opinion, the sparse vectors represent a more realistic situation. This is because tasks usually affect the impacts of only a few components, rather than influencing all components in a similar way. The sparsity of these vectors aligns more closely with real-world task characteristics.

Moreover, three of our strategies – *bagging_divide*, *bagging_remove_divide*, and *bagging_remove_reverse_divide* – incorporated an additional feature: scaling some of the vector components. This scaling mechanism represents another step towards realism in our simulations. From the perspective of a single component, different tasks typically affect it with varying magnitudes. By incorporating this scaling, we aimed to more accurately model the diverse effects that different tasks can have on individual components.

We employ the *cosine distance*, defined as $cos\_dist(\vec{n},\vec{n}') = 1 - cos\_sim(\vec{n},\vec{n}') = 1 - \frac{\vec{n}\cdot\vec{n}'}{|\vec{n}||\vec{n}'|}$, where $\vec{n}$ and $\vec{n}'$ are impact vectors, to compare sets of vectors generated using the same mode, allowing us to quantify the dissimilarity between vectors within each generation round. Since the dot product of two non-negative vectors is always non-negative, and the magnitudes of non-negative vectors are always positive, we have that for our vectors, the cosine distance is always between 0 and 1 inclusive.

We argue, and then experimentally verify, that the more distant (with respect to cosine distance) the vectors in a set we employ for labeling each task are, the more computationally challenging it will be for our algorithms to converge to a solution (if any exists).

For this purpose, we conducted further analysis of each *dimension* and *mode*. In our experiment, we considered *dimensions* ranging from 1 to 10 and utilized all *six modes*. To analyze these vectors, we compute the *cosine distance* between all possible pairs of vectors within each generated set of 100 vectors. For each set of vectors (*4950 pairs per set*), we will compute the *mean* and the *standard deviation* of the cosine distances.

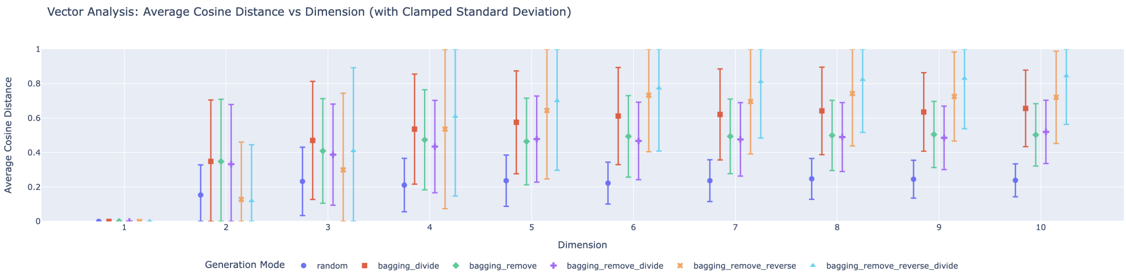


Figure 3.8: Variation of mean and standard deviation of cosine distance across dimensions and modes.

In Figure 3.8, we show the results of our analysis. We observe that *bagging_divide* produces vectors with high average distances across all dimensions. In contrast, *bagging_remove_reverse* and *bagging_remove_reverse_divide* (i.e., the sparse generating modes) produce vectors with very high distances, particularly in higher dimensions.

Before delving into the algorithms analyzed in our experiments, it is essential to introduce our concept of the Pareto frontier, which plays a crucial role in understanding the performance and outcomes of one of those algorithms.

Definition 3.5.1 (Pareto Frontier)**.** *A set of vectors* $P \subseteq \mathbb{R}^k$ *is a Pareto frontier if and only if:* $\forall \vec{n}, \vec{m} \in F$ *with* $\vec{n} \neq \vec{m}$ *we have* $\vec{n} \not\leq \vec{m} \wedge \vec{m} \not\leq \vec{n}$*.*

We associate a Pareto frontier $P(R)$ to each node R in a region tree, where $P(R)$ represents the set of minimal impact vectors achievable from that region. Formally:

$$P(R) = \{\vec{n} \in \mathbb{R}^k \mid \vec{n} \text{ is a minimal impact vector achievable from region } R\}$$

where an impact vector $\vec{n}$ is considered minimal if there exists no other achievable impact vector $\vec{m}$ from region R such that $\vec{m} < \vec{n}$. It is easy to see that the Pareto frontier of the root of region tree represents the set of all and only minimal bounds for which the associated process and impacts admits a solution for Problem 1.

At this point, we have developed a method for generating `BPMN` processes with specified MNXN and MIX parameters, and for assigning impact vectors to the generated tasks based on given dimension and mode. Using this foundation, we analyze the performance of two algorithms related to the problems summarized in Table 3.3.
The first algorithm, which we call `ComputeMax` for brevity, is a polynomial-time variant of Algorithm 3. `ComputeMax` calculates the maximum obtainable impacts for each component in every trace of the input process. This means that if we require all traces to be below a given bound (as checked by Algorithm 3), such a bound must dominate the output of `ComputeMax`. `ComputeMax` has the same theoretical complexity as Algorithm 3, but eliminates the need for an additional bound parameter. Moreover, `ComputeMax` represents the worst-case computation of Algorithm 3 over all possible bounds, as it does not benefit from early failure detection optimizations present in Algorithm 3. In our experiments, we denote the computational time of `ComputeMax` on our synthetic dataset as Max_time.

The second algorithm, `ComputePareto`, is designed to represent the worst-case scenario for Algorithms 1 and 2 (which are NP-hard for $k > 1$), without requiring the specification of an input bound. `ComputePareto` operates similarly to Algorithm 1, but instead of guessing, for each node in the region tree it constructs the entire Pareto frontier. For XOR gateways, the cardinality of the Pareto frontier is at most the sum of the frontiers of its children, while for parallel gateways, it is at most the product. The actual frontier may be smaller than this theoretical maximum (which is computed and denoted as Max_theoretical_pareto in our experiments) due to the removal of dominated vectors during merging operations. In our experiments, we denote the size of the maximum real Pareto frontier encountered while processing the region tree as Max_pareto. It can be shown that the theoretical worst-case scenario produces a frontier that is exponential in the size of the input process, and consequently, `ComputePareto` has exponential complexity in the input process size. In our experimental results, we denote the computational time of `ComputePareto` as Pareto_time. The code for both algorithms is available in the repository provided for reproducibility (file `algorithms.py`).

In our experiment, we investigated how different modes affect the size of the Pareto frontier. More broadly, this analysis sheds light on how the distribution of the impact vectors, as measured by the cosine distance, affects the size of the Pareto frontier.

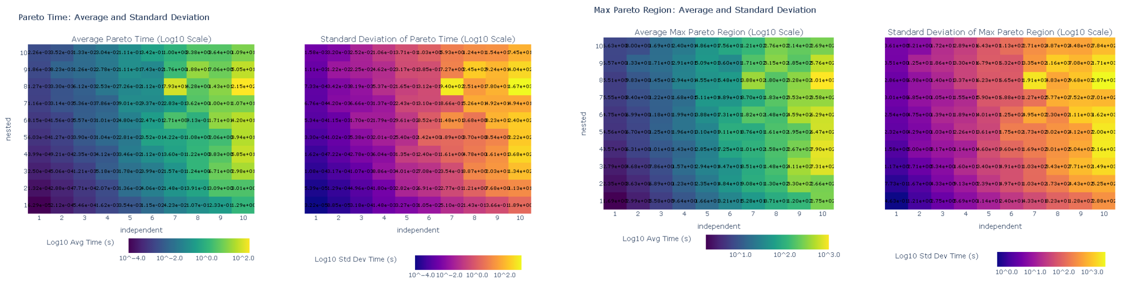
Our possible configurations span several parameters. We consider dimensions ranging from *1 to 10*, MNXN values from *1 to 10 nested XORs*, and a fixed MIX of *10 independent XORs*. Additionally, we employ *6 different generation modes*. For each combination of MNXN and MIX, we generate 10 distinct processes that meet the requirements of the combination. By combining these parameters, we achieved a total of *60,000 unique configurations*.

We performed a detailed analysis of the relationships between the *process structure* (characterized by MNXN and MIX parameters), the *characteristics of the impact vectors* (determined by dimension and mode), and the *performance of the algorithms*.

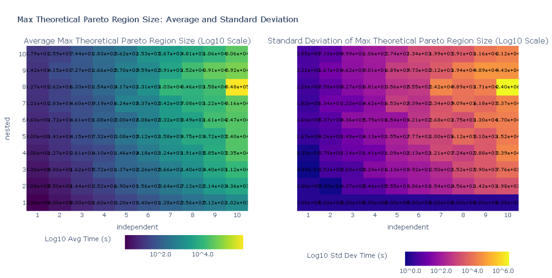
The code and experimental results are available in the GitHub repository. This repository includes comprehensive *data* and *analyses of the business processes*, offering valuable insights and tools for further research and development. The detailed documentation and code can be accessed directly at the following URL:

`https://github.com/PietroSala/process-impact-benchmarks`.

Using this dataset, we conducted various analyses, as detailed below. Specifically, in Figure 3.9, heatmaps are created for the experimental metrics: Pareto_time, Max_pareto, Max_theoretical_pareto, and Max_time. These heatmaps are plotted on a logarithmic scale to capture the exponential nature of the metrics, except for Max_time, which is plotted on a linear scale due to the fact that represents the execution time of an algorithm working in polynomial time. Each cell in the heatmap groups processes based on their *independent* and *nested* parameters, representing the maximum number of *independent XORs (MIX)* and *nested XORs (MNXN)*, respectively. For each metric, two heatmaps are produced: one depicting the *average value* and another showing the *standard deviation*. This analysis leads to two key findings:



(a) Pareto_time (b) Max_pareto



(c) Max_theoretical_pareto (d) Max_time

Figure 3.9: Heatmaps of various metrics. Each row represents a different metric: Pareto_time, Max_pareto, Max_theoretical_pareto, and Max_time, respectively. The left column shows the *average values* (with a log scale applied to the first three metrics), and the right column shows the *standard deviations* (with a log scale applied to the first three metrics).

- *Combined effects of parameters*: When both the *independent and nested* parameters are increased together, the *averages* and *standard deviations* for metrics such as Pareto_time are significantly higher than when either parameter is increased alone. In other words, both the *average* and the *standard deviation* are affected more when we increase the *independent and nested* parameters together than when we increase only one of the two parameters.

- *High standard deviations* indicate considerable variability within each group, which is likely due to unaccounted factors such as *modes* and *dimensions*. That is all processes with the same combination of *independent* and *nested XORs* are categorized in the same range, regardless of their *modes* or *dimension*.

These observations highlight the complex interactions between the *independent* and *nested* parameters and suggest indicating that further refinement is needed to incorporate additional variables, such as *modes* and *dimensions*.

To evaluate the performance of different *modes* across various *dimensions*, we conducted an experiment analyzing three modes: *random*, *bagging_divide'*, and *bagging_remove_reverse_divide*, for dimensions 2, 6, and 10. The three modes - *random*, *bagging_divide*, and *bagging_remove_reverse_divide* - were strategically chosen to represent a spectrum of dissimilarity according to the results presented in Figure 3.8: from similar impact vectors, through an average case of dissimilarity, to strong dissimilarity, respectively.

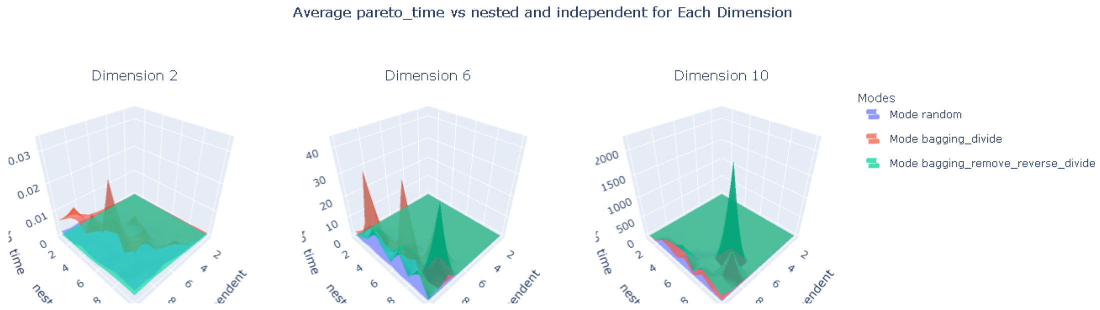
The results are visualized using rotated 3D plots in both logarithmic and non-logarithmic scales (Figure 3.10). Key observations resulting from this analysis include:

- *Performance Trends*: Across all *modes* and *dimensions*, we consistently observed that the *average computation time* increases as both the *nested* (MNXN) and *independent* (MIX) parameters grow. This trend highlights the escalating *computational complexity* associated with the simultaneous growth of the MIX and MNXN parameters.

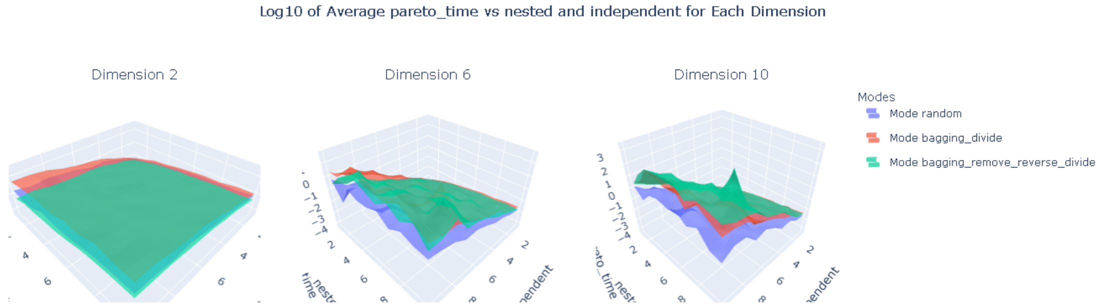
- *Mode Comparisons*: The analysis reveals similar trends across all modes. However, a notable pattern emerges as the dimension increases.
  The *bagging_remove_reverse_divide* strategy consistently produces impact vectors that are more computationally challenging for the same process. This aligns with our expectations, given the strong dissimilarity between the impact vectors generated by this mode.

The analysis depicted in Figure 3.11 demonstrates that as the average similarity among vectors decreases and the impact vector dimension increases, both the Pareto_time (Figure 3.11a) and the Max_pareto_frontier (Figure 3.11b) increase. This situation is closer to real-life scenarios where the impact of specific tasks may vary significantly. When comparing different modes,*bagging_remove_reverse_divide* consistently exhibits a greater Pareto_time on larger dimensions. Moreover, the correlation between dimensions and Max_pareto suggests that *bagging_remove_reverse_divide* yields a higher Max_pareto compared to other modes. This observation implies that even with growing dimensions if the vectors are not similar, the size of the Pareto frontier (Max_pareto) tends to grow. Then, as computational time is significantly increased, we have that *bagging_remove_reverse_divide* is an effective *mode* for simulating challenging impact vectors in high dimensions. This approach allows us to push computational resources to their limits.

The analysis of Figure 3.12 provides relevant information about how the complexity of `BPMN` processes relates to *computation time*. In these visualizations, the x-axis represents the overall

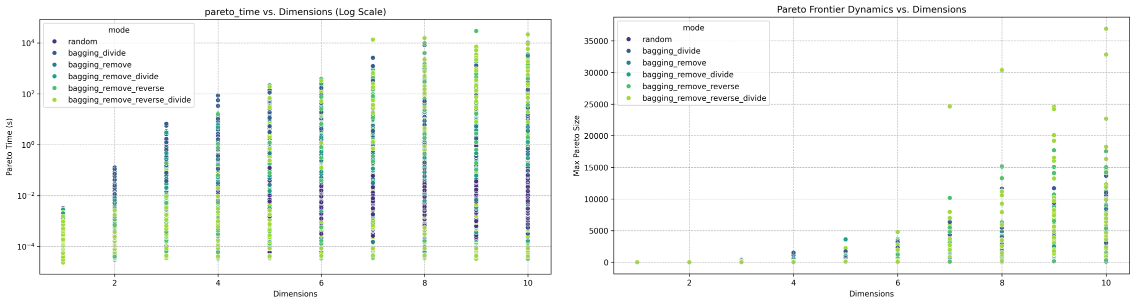


(a) Average computation time across dimensions.



(b) Logarithmic scale for better visualization of the data.

Figure 3.10: The influence of modes *random*, *bagging_divide*, and *bagging_remove_reverse_divide* on the average Pareto_time for dimensions 2, 6, and 10. The average Pareto_time is shown in both normal (a) using logarithmic scale (b).



(a) Pareto_time vs. Dimensions (Log Scale). (b) Max_pareto vs. Dimensions.

Figure 3.11: Comparison of different modes: (a) Impact on computational time and complexity for varying dimensions, and (b) Relationship between dimensions and the size of the Pareto frontier.

complexity of the BPMN *process*, defined as the product of nested and independent XOR levels. The y-axis shows the mean Pareto_time on a logarithmic scale, indicating the average processing duration for each level of complexity. The analysis of these plots reveals a clear trend across all

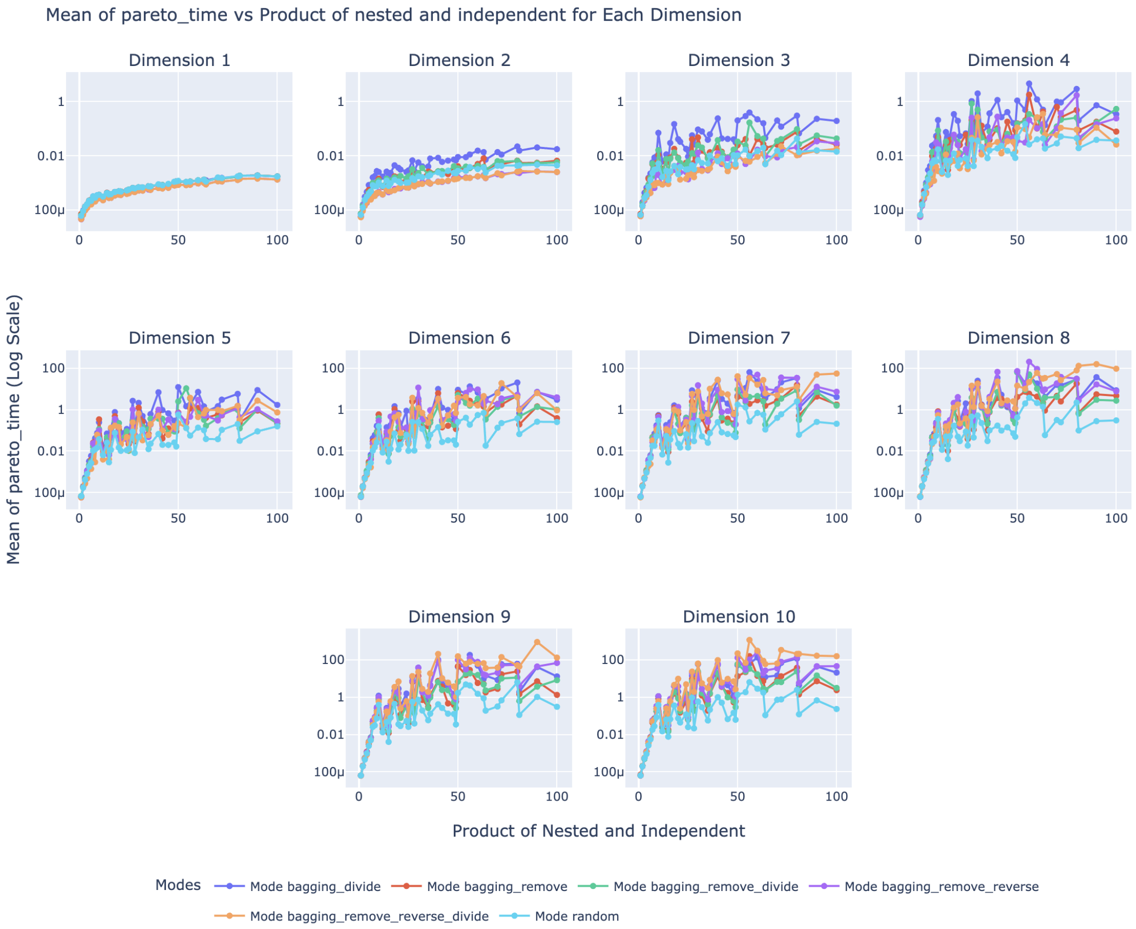


Figure 3.12: Relationship between BPMN **process complexity** and **computation time** across different vector generation strategies.

six vector generation strategies. As the product of *nested* (MNXN) and *independent XOR* (MIX) levels increase, we consistently notice a rise in Pareto_time. The positive correlation indicates that processing time increases with more complex `BPMN` structures with more *nested* and *independent XOR gateways*. Importantly, this trend holds regardless of the vector generation method used. The consistency across different modes highlights the relationship between process complexity and computational demands. As `BPMN` processes become more complex, with more decision points and parallel paths, the time needed to analyze and compute relevant metrics also increases.

This section provides a first analysis highlighting how the interplay between MNXN and MIX parameters is crucial for the *computational costs* linked to complex process structures. Organizations and process designers need to be aware that adding more nested and independent XOR gateways to `BPMN` models will likely cause the computation time to increase if the analysis requires solving problems like Problem 1 or Problem 3. Organizations can effectively manage the balance between *complexity* and *performance* by considering detailed process models and computational resource limitations.

3.6 Summary of Findings

In this dissertation, we concentrate on the novel concept of *impact compliance*, which we have shown in Section 3.1 finds several real-life applications. The general results are negative, for we cannot solve the problem of compliance in polynomial time on deterministic machines (if $P \neq NP$), but we can also determine both subcases in which the problem is so, and a method to establish the actual computational needs for a single instance in order to plan the execution relatively to the computational resources, providing a polynomial heuristics.

In order to give the reader a map of the main theoretical results before entering the details of the research, we devise a summary of these in Table 3.3.

Further on, we analise the results by means of a small-scale experiment that illustrates the practical behavior of the algorithms, while confirming the relevance of the heuristics we found.

Algorithm 1: An algorithm for solving Problem 1.

```
Input  : A structured BPMN $G = (V, E, L)$, an impact function $\mathscr{I} : V \to \mathbb{N}^k$, a
         bound $b \in \mathbb{N}^k$
Output: A boolean value which is true if and only if there exists an
         impact-conformant word $w \in T^+$
let $RT = (\mathscr{R}, \mathscr{E})$ be the region tree of $G = (V, E, L)$
$\langle R_1, \dots, R_m \rangle \leftarrow reverse$(Breadth First Visit of $RT$)
$M_I \leftarrow 0^{k \times m}$
for $i \leftarrow 1$ to $m$ do
    if $R_i$ is sequence-case then
        $submax(R_i) \leftarrow \{R_{i_1}, \dots, R_{i_{m_i}}\}$
        $task_i \leftarrow \{v \in V : L(v) = task, v \in R_{i_j} \setminus \bigcup_{1 \le j \le m_i} \overline{R}_{i_j}\}$
        $M_I[:, i] \leftarrow \sum_{j=1}^{m_i} M_I[:, i_j] + \sum_{v \in task_i} \mathscr{I}(v)$
    if $R_i$ is parallel-case then
        $submax(R_i) \leftarrow \{R_{i_1}, R_{i_2}\}$
        $M_I[:, i] \leftarrow M_I[:, i_1] + M_I[:, i_2]$
    if $R_i$ is loop-case then
        let $R_{i'}$ be the forward region of $R_i$
        $M_I[:, i] \leftarrow M_I[:, i']$
    if $R_i$ is exclusive-case then
        $submax(R_i) \leftarrow \{R_{i_1}, R_{i_2}\}$
        if $M_I[:, i_1] \le M_I[:, i_2]$ then
            $M_I[:, i] \leftarrow M_I[:, i_1]$
        else if $M_I[:, i_2] \le M_I[:, i_1]$ then
            $M_I[:, i] \leftarrow M_I[:, i_2]$
        else
            GUESS $j \in \{1, 2\}$
            $M_I[:, i] \leftarrow M_I[:, i_j]$
if $\exists h$ $(1 \le h \le k$ and $M_I[h, m] > b[h])$ then
    return $\bot$
return $\top$
```

Algorithm 2: Dichotomic Search Algorithm for 3

```
Data: Graph $\mathscr{G}$, Parameters $k$, $|V|$, Impact vector $b \in \mathbb{N}^k$
Result: Minimal impact vector
let $b$ an impact vector that minimizes component 0[4]
$i \leftarrow 1$
$up \leftarrow b[i]$
$low \leftarrow -1$
while $up \neq low + 1$ do
    if $i < k - 1$ then
        $i \leftarrow i + 1$
    else
        return $b$
    $current \leftarrow \lfloor \frac{b[i]}{2} \rfloor$
    let $b'$ such that $b'[j] = \begin{cases} current & \text{if } j = i \\ b[j] & \text{otherwise} \end{cases}$ for all $0 \leq j < k$
    if Algorithm 1 on $\mathscr{G}$, $\mathscr{I}$, $b'$ returns $\top$ then
        $up \leftarrow$ current
    else
        $low \leftarrow$ current
return $b'$
```

Algorithm 3: Algorithm to determine when impact of Maximum Path is below a bound

```
Input  : A structured BPMN $G = (V, E, L)$, an impact function $\mathscr{I} : V \to \mathbb{N}^k$ a
         bound $b \in \mathbb{N}^k$
Output: $\top$ if and only if $\mathscr{I}$ for all the traces $w \in T^+$, $\mathscr{I}(w) \leq b$
let $RT = (\mathscr{R}, \mathscr{E})$ be the region tree of $G = (V, E, L)$
$\langle R_1, \dots, R_m \rangle \leftarrow reverse($Breadth First Visit of $RT$ starting from $root(RT))$ )
$M \leftarrow 0^{k \times m}$
for $i \leftarrow 1$ to $m$ do
    if $R_i$ is task then
        $M_I[:, i] \leftarrow \mathscr{I}(i)$
for $i \leftarrow 1$ to $m$ do
    if $R_i$ is sequence-case then
        let $(R_i) = \{R_{i_1}, \dots, R_{i_{m_i}}\}$
        $M[:, i] \leftarrow \sum_{j=1}^{m_i} M[:, i_j]$
    if $R_i$ is parallel-case then
        let $(R_i) = \{R_{i_1}, R_{i_2}\}$
        $M[:, i] \leftarrow M[:, i_1] + M[:, i_2]$
    if $R_i$ is exclusive-case then
        let $(R_i) = \{R_{i_1}, R_{i_2}\}$
        for $j \leftarrow 1$ to $k$ do
            $M[j, i] \leftarrow \max(M[j, i_1], M[j, i_2])$
    if there exists $1 \leq j \leq k$ s.t. $M[j, i] > b[j]$ then
        return $\perp$
return $\top$
```

Algorithm 4: Max Independent Number-XOR (MIX).

```
input  : A structured BPMN $G = (V, E, L)$
output: A number representing the maximum independent XOR-splits in $G$
let $RT = (\mathscr{R}, \mathscr{E})$ be the region tree of $G = (V, E, L)$
$\langle R_1, \ldots, R_m \rangle \leftarrow$ reverse $\left( \begin{array}{c} \text{Breadth First Visit of } RT \\ \text{starting from } root(RT) \end{array} \right)$
for $i \leftarrow 1$ to $m$ do
    if $R_i$ is Task then
        return 0                // Base case: leaf node representing a task
    if $R_i$ is sequence-case then
        return $\text{MIX}(R1) + \text{MIX}(R2)$
    if $R_i$ is parallel-case then
        return $\text{MIX}(R1) + \text{MIX}(R2)$
    if $R_i$ is exclusive-case then
        return $\max(\text{MIX}(R1), \text{MIX}(R2))$
```

Algorithm 5: Max Nested XOR Number (MNXN).

```
input  : A structured BPMN $G = (V, E, L)$
output: A number representing the maximum nested XOR-splits in $G$
let $RT = (\mathscr{R}, \mathscr{E})$ be the region tree of $G = (V, E, L)$
$\langle R_1, \ldots, R_m \rangle \leftarrow$ reverse $\begin{pmatrix} \text{Breadth First Visit of } RT \\ \text{starting from } root(RT) \end{pmatrix}$
for $i \leftarrow 1$ to $m$ do
    if $R_i$ is Task then
        return 0                              // Base case: leaf node representing a task
    if $R_i$ is sequence-case then
        return max(MNXN($R1$) + MNXN($R2$))
    if $R_i$ is parallel-case then
        return MNXN($R1$) + MNXN($R2$)
    if $R_i$ is exclusive-case then
        return $\max($MNXN($R1$), MNXN($R2$)$) + 1$
```

Problem	**Algorithm**	**Complexity**	**Reference**
Problem 1: show the existence of one trace below a bound of size $k > 1$.	Alg. 1 p. 65	$\mathscr{NP}$-complete	Th. 3.4.1 p. 44 and Th. 3.4.9, p. 51.
Problem 1: exhibit one trace below an input bound of size $k = 1$.	Alg. 1 p. 65	$\mathscr{O}(\lvert V\rvert)$	Th. 3.4.1 p. 44 and Th. 3.4.3.
Problem 2: determine whether traces are below one input bound of size k.	Alg. 3 p. 67	$\mathscr{O}(k\lvert V\rvert)$	Th. 3.4.4 p. 49.
Problem 3: exhibit a minimal trace with $k > 1$ impacts.	Alg. 2 p. 66	$\mathscr{NP}$-hard	Th. 3.4.2 and Th. 3.4.3 p. 47.
Problem 3: exhibit a minimal trace with $k = 1$ impacts.	Alg. 2 p. 66	$\mathscr{O}(\lvert V\rvert)$	Th. 3.4.2 p. 47 and Th. 3.4.9.
Heuristics for Problem 1: computing the no. of independent and nested XORs	Alg. 4 and 5 p. s 68 and 69	$\mathscr{O}(\lvert V\rvert)$	Th. 3.4.11 p. 52 and Th. 3.4.12 p. 52.

`Table 3.3: Summary of the most important results of the paper.`

Chapter Four

An Approach for Computing Similarity Between Business Processes Based on Resource Consumption Impacts

Business Process Management (BPM) plays a critical role in the effective operation of organizations, where business processes (BPs) refer to the series of tasks executed to achieve specific objectives. As organizations expand, understanding the *similarities* and *differences* between these processes becomes essential for enhancing efficiency, identifying best practices, and ensuring compliance with established protocols.

Existing methods for measuring business process similarity predominantly focus on the *structural* and *behavioral* dimensions, such as the sequence of activities, control flows, or execution traces. However, these approaches often overlook the influence of *resource consumption*, a factor crucial to optimizing overall process performance. To address this limitation, I propose a novel approach for computing business process similarity that incorporates the impact of *resource consumption*.

Several methods have been proposed to assess *business process similarity*, with an emphasis on *structural* and *behavioral* aspects. Early work, such as that by Dijkman et al. (2009), introduced *graph-based* techniques where business processes are represented as directed graphs, and similarity is computed using *graph-edit distances* [15]. These methods focus on quantifying the structural differences between process models by analyzing their control flow.

Trace-based approaches, such as the *trace clustering* technique proposed by Greco et al. (2006), group similar processes by examining their *execution traces* [48]. This approach compares process behavior by focusing on the sequence of tasks as they are executed.

More recently, performance metrics such as *time* and *cost* have been integrated into similarity measures. For instance, Polyvyanyy et al. (2010) extended traditional approaches by incorporating *process compliance* and performance metrics to enhance similarity calculations [17]. Despite these advancements, these studies fail to adequately address the critical impact of resource consumption, which is vital in practical applications.

This work fills that gap by introducing similarity measures that explicitly account for *resource consumption impacts*. By incorporating resource utilization, the proposed approach offers a more comprehensive perspective on business process similarity, extending beyond purely structural or behavioral factors.

To compute resource impacts, I represent resource consumption using a *matrix* and introduce two similarity measures: a modified version of *cosine similarity* and *Euclidean-based similarity*. Additionally, I present an *algorithm* for calculating the resource impacts of processes modeled in *Business Process Model and Notation (BPMN)*.

4.1 Background and Motivation

The *business process similarity* problem is traditionally approached by examining the *structural* or *behavioral similarities* between two processes. Behavioral similarity measures compare *execution traces*, while structural similarity focuses on *process control flows*. Although these methods are effective, they fail to account for *resource impacts*, critical in real-world applications, particularly in industries that rely heavily on *resource optimization*.

Resource consumption in business processes can refer to various factors, such as *time*, *cost*, or *energy* expended during task execution. Understanding these dimensions allows organizations to identify processes that are not only *functionally similar* but also *efficient* in terms of resource utilization.

This dissertation extends existing similarity measures by incorporating *resource consumption* and providing a comprehensive method to calculate *business process similarities*.

4.2 Proposed Methodology

4.2.1 Terms and concepts

The Business Process Model and Notation (BPMN) provides various methods for tracing process flows, including sequences, parallel flows, splits, and task flows. These elements are described as follows:

- *Sequence Flow*: A graphical representation illustrating the sequence of activities within a business process. It provides a clear depiction of the flow of work or information.
- *Parallel Flow*: Represents the simultaneous execution of multiple activities within a process. This contributes to greater efficiency and flexibility in handling different aspects of the overall workflow.
- *Split Flow*: Represents the division of a process into several parallel paths, allowing different activities to be executed simultaneously. This concept is essential for modeling situations where tasks can occur concurrently, contributing to a more streamlined and efficient workflow.
- *Task*: Represents a single unit of work that must be carried out within the process.

Suppose that an $m \times n$ matrix A represents resource consumption during the execution of each task in a process. Matrix A can be expressed as

$$A = (a_{ij})_{m \times n}$$

where i represents the task index, and j represents the trace index. In order to calculate the average consumption of each task, we sum the consumption of each task across all traces and divide it by the number of traces. This can be mathematically represented as:

$$\text{avg}_i = \frac{1}{N} \sum_j a_{ij}$$

where N is the number of traces.

Then, we can represent the impact of each task as vector B, where the ith element of the vector represents the average consumption of the ith task. This can be denoted as:

$$B = (\text{avg}_1, \text{avg}_2, \ldots, \text{avg}_n)$$

where n denotes the number of tasks in a process. Thus, B is the average consumption of a trace.

We consider a small subset of BPMN, namely a subset containing just *pools, swimlanes, tasks, starting/ending events, and, exclusive/parallel gateways*. However, for the purposes of this work, both pools and swimlanes will be used just as syntactic annotations, so we omit them from the formal notation.

A BPMN is a labelled directed graph $G = (V, E, L)$ where

$$L : V \to \{\text{task, start, end, } \textit{Exclusive Split}, \textit{Exclusive Join}, \textit{Parallel Split}, \textit{Parallel Join}\}$$

such that for every $v \in V$:

(i) if $L(v) = \text{start}$, then for every v' with $L(v') = \text{start}$, we have $v = v'$ (*unique start event*). Moreover, $|\{(v, v') \in E\}| = 1$ (*exactly one outgoing edge from the start event*) and $|\{(v', v) \in E\}| = 0$ (*no incoming edges to the start event*).

(ii) if $L(v) = \text{end}$, then $|\{(v', v) \in E\}| = 1$ (*exactly one incoming edge to an end event*) and $|\{(v, v') \in E\}| = 0$ (*no outgoing edges from an end event*).

(iii) if $L(v) = \text{task}$, then $|\{(v', v) \in E\}| = |\{(v, v') \in E\}| = 1$ (*exactly one incoming and outgoing edge for a task*).

(iv) if $L(v) = \textit{Exclusive Split}$ (resp., $L(v) = \textit{Parallel Split}$), then $|\{(v', v) \in E\}| = 1$ (*exactly one incoming edge to a split gateway*) and $|\{(v, v') \in E\}| = 2$ (*exactly two outgoing edges from a split gateway*).

(v) if $L(v) = \textit{Exclusive Join}$ (resp., $L(v) = \textit{Parallel Join}$), then $|\{(v', v) \in E\}| = 2$ (*exactly two incoming edges to a join gateway*) and $|\{(v, v') \in E\}| = 1$ (*exactly one outgoing edge from a join gateway*).

Let us observe that we may have multiple end events.

In the following, we assume our BPMN to be structured as *Single Entry Single Exit (SESE) Regions* [16]. Given a BPMN $G = (V, E, L)$, we may create a *set cover*

$$\mathscr{R} = \{R_1, \ldots, R_m\}$$

of V, i.e., a set of subsets of V whose union is equal to V itself, such that for every $1 \leq i \leq m$:

1 - *disjointness or inclusion* - for every $1 \leq j \leq m$, either $R_i \cap R_j = \emptyset$, $R_i \subseteq R_j$, or $R_i \supseteq R_j$.

2 - *single entry* - there exists at most one $v \in V \setminus R_i$ such that $(v, v') \in E$ with $v' \in R_i$. We call v' (if any) the *entry-point* of R_i, denoted by $entry(R_i)$.

3 - *single exit* - there exists at most one $v \in V \setminus R_i$ such that $(v', v) \in E$ with $v' \in R_i$. We call v' (if any) the *exit-point* of R_i, denoted by $exit(R_i)$.

4 - *region maximality* - let $submax(R_i)$ denote the set of all and only maximal proper sub-regions of R_i. Various cases may arise:

  a - *branching case* - $submax(R_i) = \{R_j, R_{j'}\}$, R_i contains a split and join.

  b - *loop case* - $submax(R_i) = \{R_j, R_{j'}\}$, R_i contains a loop with forward and backward regions.

  c - *sequence case* - $submax(R_i) = \{R_1^i, R_{m_i}^i\}$, where R_i consists of sequential tasks.

From now on, we call a BPMN $G = (V, E, L)$ structured in SESE regions a *structured BPMN*. The unique set cover $\mathscr{R}$ can be organized as a rooted tree, called the *region tree* of G. Below we present a diagram denoted as G_1. Diagram G_1 is structured in Single Entry Single Exit (SESE) form.We assume that the input process is initially provided in Single Entry Single Exit form (SESE) as described in [16]. Business Process Model and Notation (BPMN) diagram in SESE form adheres to a specific structure. For every gateway-split node, there exists a unique corresponding gateway-join node. This relationship ensures that the region formed by such a pair of gateway nodes has precisely one incoming edge and one exit edge. Conversely, a BPMN diagram without SESE form may have a situation where one incoming edge has more than one exit edge. The main advantage of SESE representation is : (i) a trace-equivalent, [1] It can always be obtained from a generic BPMN diagram.

4.2.2 Similarity Measures

In the context of business processes, similarity measures refer to methods or techniques used to determine the degree of resemblance between two processes. It is often desirable to compare processes that achieve the same goals but have different resource impacts, allowing for the selection of the process with the best overall impact. The similarity measures discussed in this context aim to quantify the degree of similarity between two processes based on their *resource impacts*.

---

[1] Two BPMN diagrams are trace equivalent if they produce the same traces, see [44]

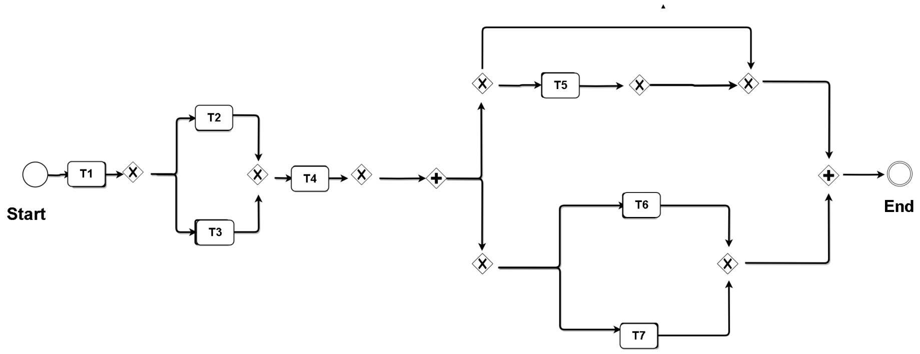


Figure 4.1: G_1 BPMN structure with Single Entry Single Exit (SESE) form

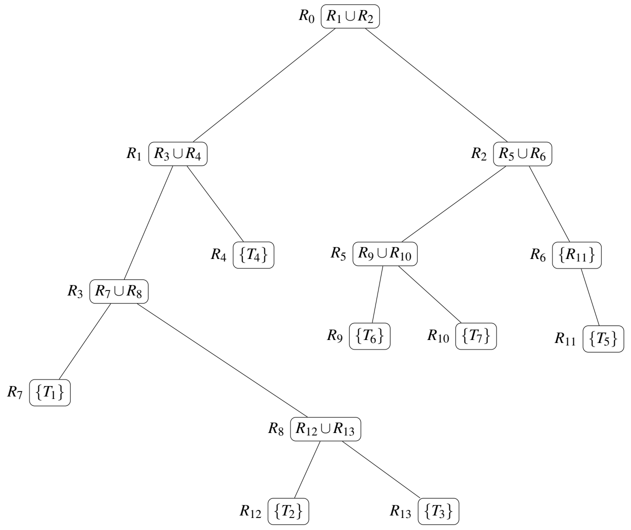


Figure 4.2: The region tree of the BPMN depicted in Figure4.1

The study of business process similarity involves a variety of methodologies designed to quantify the similarity between different business processes. These methodologies span across several domains, including *structural-based*, *behavioral-based*, *graph-based*, *semantic-based*, and *metric-based* approaches.

Structural-based similarity analysis focuses on the structural aspects of business processes, examining the arrangement of activities, tasks, and their relationships within process models. This approach often compares the topology, ordering, and flow of activities to assess similarity.

In contrast to *behavioral-based* methods, which emphasize the dynamic execution of processes, structural-based approaches prioritize the static representation of process models. By examining structural characteristics such as the sequence of activities and their dependencies, structural-based similarity analysis provides valuable insights into the overall structure and organization of business processes.

Structural-based similarity metrics may include measures such as *edit distance*, *graph similarity*, or *feature-based comparison*. These metrics quantify the similarity between process models based on their structural properties, allowing for objective comparisons and evaluations. Overall, structural-based similarity analysis offers a foundational framework for assessing the likeness of business processes, enabling organizations to identify commonalities, differences, and potential areas for optimization or standardization.

Average Consumption: *Average consumption* refers to the amount of resources consumed by tasks across all lanes/traces of a business process.

Minimum Impact: *Minimum impact* represents a lower bound on the minimum amount of resources consumed by tasks in all lanes/traces of a business process.

Maximum Impact: *Maximum impact* represents a lower bound on the maximum amount of resources consumed by tasks in all lanes/traces of a business process.

Both the maximum and minimum impact bounds are used in place of exact minimal/maximal impacts for several reasons:

1. From a computational perspective, calculating the exact minimal/maximal impact of a BPMN diagram is an NP-complete problem when dealing with multiple impacts.
2. Minimal/maximal impacts form a *Pareto-like frontier*, meaning there may be more than one minimal/maximal impact, complicating the normalization of impact vectors, as we will propose later.

3. By considering the *minimum* (or maximum) *impact bound*, defined as the largest (or smallest) impact vector dominated by (or dominating) all the vectors in the minimal (or maximal) Pareto frontier, we can establish a unique and strict minimal (or maximal) bound for the impact vectors.
4. As we will show in section 4.4, the maximal/minimal error bounds are easy to compute and can be calculated alongside the average impact vector of a BPMN process.

4.3 Computing Similarity Between Business Processes

4.3.1 Resource Consumption as a Matrix

We represent the *resource consumption* or *impact* of tasks in a business process as a matrix A. Each row of the matrix corresponds to a task, while each column represents a specific process trace. The matrix is defined as a Boolean matrix:

$$a_{ij} = \begin{cases} 1 & \text{if task } i \text{ was executed in trace } j, \\ 0 & \text{otherwise.} \end{cases}$$

Here, i is the task index, and j is the trace index.

To compute the *average consumption* of each task across multiple traces, we sum the task's consumption over all traces and divide by the total number of traces. This can be expressed as:

$$\text{avg}_h = \frac{1}{N} \sum_j a_{ij} \cdot \mathscr{I}(i)[h],$$

where N is the number of traces, a_{ij} indicates the execution of task i in trace j, and $\mathscr{I}(i)[h]$ is an indicator function that equals 1 if task i is present in trace h, and 0 otherwise.

The *impact* of each task is then represented as a vector B, where the i-th element corresponds to the average consumption of task i. This vector is denoted as:

$$B = [\text{avg}_1, \text{avg}_2, \ldots, \text{avg}_k],$$

where k is the total number of tasks in the process.

4.4 An Algorithm for Computing Impacts of BPMN processes

In this section, we provide an algorithm for computing the average, minimum, and maximum impact of a BPMN process to compute the two similarity measures proposed above. We assume that the input process is given in Single Entry Single Exit form (SESE) [17]. Informally speaking, a BPMN in SESE form is a diagram where for each gateway-split node, there is a unique corresponding gateway-join node such that the region formed by such a pair of gateway nodes has exactly one incoming edge and one exit edge. The main advantages of this representation are the following: (i) a trace-equivalent [2] it is always available from a generic BPMN diagram; (ii) the construction is effective even if its corresponding decision problem is NP-complete, related to the unfolding of Petri nets [24]; (iii) the fact that BPMN diagrams are special cases (1-place bounded) of Petri nets and there are a plethora of optimized tools for the unfolding problem that make the equivalent SESE BPMN achievable in practice [18]. A notebook containing a language for expressing BPMN diagrams in SESE form, a recursive version of the algorithm below, and a couple of running examples based on the ones proposed in Section 4.5is available at

`https://github.com/PietroSala/process-impacts` (the main notebook is `impact_based_similarity.ipynb`).

In the following, given a SESE structured BPMN $G = (V, E, L)$ we denote with $V_\times \subseteq V$ the set of or-splits gateways in V and with $V_\circlearrowleft \subseteq V$ the set of or-splits in V that close a loop-region (i.e., $V_\circlearrowleft \subseteq V_\times$).

In Algorithm 6 we assume that the impact vectors $M_I[:,m], M_\downarrow[:,m], M_\uparrow[:,m]$ contains in the first k components the impacts for cumulative resources and in the last $k+1$ to $k+h$ components the impacts for non-cumulative resources. Moreover, given two vectors $\overline{v_1}, \overline{v_2}$ belonging to the same space $\mathbb{N}^n$ we use the notation $\min(\overline{v_1}, \overline{v_2})$ (resp., $\max(\overline{v_1}, \overline{v_2})$) to denote the component-wise minimum (resp., maximum) of the two impact vectors. Finally, let us observe that the algorithm has the same complexity as a breadth-first visit of the BPMN diagram (which is linear in the size of the diagram) provided that the BPMN is given in SESE form.

---

[2] Two BPMN diagrams are trace equivalent if they produce exactly the same traces, see [44] for example.

Algorithm 6: Average impact of BPMN

```
Input  : BPMN $G = (V, E, L)$, cumulative impact $\mathscr{I} : V \rightarrow \mathbb{N}^k$, non-cumulative
         impact $\mathscr{C} : V \rightarrow \mathbb{N}^h$, split probability $\mathscr{P} : V_\times \rightarrow [0,1]$, loop threshold
         $\mathscr{P}_\circlearrowleft : V_\circlearrowleft \rightarrow [0,1]$
Output: Average impact vector of $G$ and $H_\downarrow, H_\uparrow \in \mathbb{N}^k$
$RT = (\mathscr{R}, \mathscr{E}) \leftarrow$ region tree of $G$
$\langle R_1, \ldots, R_m \rangle \leftarrow$ reverse BFS of $RT$
$M_I \leftarrow M_\downarrow \leftarrow M_\uparrow \leftarrow 0^{(k+h)\times m}$
for $i \leftarrow 1$ to $m$ do
    if $R_i$ is task then
        $M_*[:, i] \leftarrow \begin{bmatrix} \mathscr{I}(i) \\ \mathscr{C}(i) \end{bmatrix}$
for $i \leftarrow 1$ to $m$ do
    if $R_i$ is sequence then
        $M_*[:, i] \leftarrow \sum_j M_*[:, i_j]$
    else if $R_i$ is parallel then
        $M_*[:, i] \leftarrow \begin{bmatrix} M_*[1:k, i_1] + M_*[1:k, i_2] \\ \max(M_*[k+1:k+h, i_1], M_*[k+1:k+h, i_2]) \end{bmatrix}$
    else if $R_i$ is exclusive then
        $M_I[:, i] \leftarrow \mathscr{P}(v) M_I[:, i_1] + (1 - \mathscr{P}(v)) M_I[:, i_2]$
        $\substack{M_{\substack{\downarrow \\ \uparrow}}[j,i] \leftarrow \min \\ \max(M_{\substack{\downarrow \\ \uparrow}}[j,i_1], M_{\substack{\downarrow \\ \uparrow}}[j,i_2])}$
    else if $R_i$ is loop then
        $M_*[:, i] \leftarrow M_I[:, j]$
        while $q > \mathscr{P}_\circlearrowleft(v)$ do
            $M_I[:, i] \leftarrow M_I[:, i] + q M_I[:, j]$
            $M_\uparrow[:, i] \leftarrow M_\uparrow[:, i] + M_\uparrow[:, j]$
            $q \leftarrow q \mathscr{P}(v)$
return $M_I[:, m], M_\downarrow[:, m], M_\uparrow[:, m]$
```

This algorithm efficiently computes the average, minimum, and maximum resource consumption for a BPMN process using a breadth-first traversal of the process's region tree.

4.4.1 Modified Cosine Similarity

Cosine similarity is a well-known measure for comparing the direction of vectors, typically ranging between -1 and 1. In this work, we modify the standard cosine similarity formula to account for differences in magnitudes between the vectors:

$$\text{Similarity} = \cos(B_1, B_2) \cdot \frac{\min(||B_1||, ||B_2||)}{\max(||B_1||, ||B_2||)}, \tag{4.4.1}$$

where B_1 and B_2 are the impact vectors of two processes.

This modification normalizes the similarity measure, ensuring a fair comparison even when the magnitudes of the vectors differ.

4.4.2 Euclidean-based Similarity

Another widely-used metric in machine learning and data analysis is *Euclidean distance*, which measures the distance between two points in space. We define *Euclidean-based similarity* as follows:

$$\text{Similarity} = 1 - \frac{d(B_1, B_2)}{d(H_{\min}, H_{\max})}, \tag{4.4.2}$$

where $d(B_1, B_2)$ is the Euclidean distance between B_1 and B_2, and $H_{\min}$, $H_{\max}$ are the minimum and maximum impact vectors.

This yields a *normalized similarity score* between 0 and 1, providing a useful measure for comparing business processes.

4.4.3 Algorithm for Resource Impact Computation

An algorithm is developed to compute the average, minimum, and maximum impacts for business processes modeled using BPMN diagrams. The process assumes a Single Entry Single Exit (SESE) format for the BPMN representation.

4.5 Impact-Based Similarity in Business Processes: Illustrated with a Running Example

Impact-based similarity is crucial in various domains, enabling data-driven analysis and decision-making in real-world scenarios. In today's data-centric landscape, understanding the relationship between different inputs and their corresponding impacts is essential for optimizing system performance and enhancing decision-making processes. When analyzing business processes, selecting the optimal option based on impact becomes a natural outcome of data analysis. With this in mind, I have selected the impact of a business process as one of the key criteria for quantifying similarity between two processes. In this dissertation, I propose a method for comparing business processes based on their impacts. I introduce measures such as *average consumption*, *minimum impact*, and *maximum impact* to evaluate the similarity between processes. I focus on loop-free BPMN models to demonstrate the approach, specifically using Single-Entry Single-Exit (SESE) diagrams. These diagrams serve as the basis for illustrating the proposed method and demonstrating how it can be applied through a concrete example using the algorithm stated above.

As an example, I used two business processes labeled B_1 and B_2, each consisting of the same tasks and having the same goal. Every task in these processes is associated with specific impacts. These impact values represent different resources consumed by the tasks. For our illustration, I utilized three types of impact measures: money/cost (in euros), electricity (in kWh), and work hours (per hour).To see how I can represent this, here is a sample figure showing the representation with four impact measures:

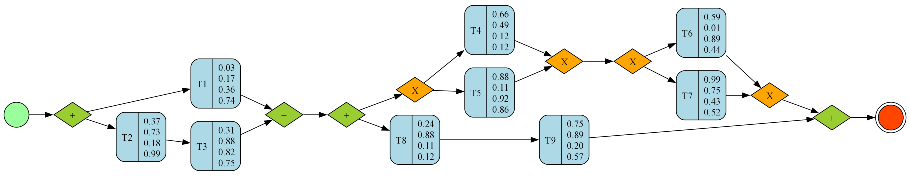


Figure 4.3: Sample figure showing the representation with four impact measures.

By utilizing a simple and expressive language, users can either directly create their **SESE (Single-Entry Single-Exit)** diagram as a region tree or convert an existing SESE diagram, expressed in the standard **XML format for BPMN (Business Process Model and Notation)** [16], into an equivalent region tree. These diagrams can be enriched with additional attributes, such as:

- **Impacts**: Impacts refer to resource consumption in simple terms. There are two types of impacts: cumulative and non-cumulative. Non-cumulative impacts are specific to individual tasks and do not accumulate over time or across other tasks. For example, the time required for each task is a common non-cumulative impact. In contrast, cumulative impacts accumulate as the process progresses, such as total cost or total energy consumed over multiple tasks.

- **Split Probability**: This explains how to handle probabilities associated with decision splits in the diagram, which influence decision-making at branches.

- **Loop Threshold**: This describes how to manage loop thresholds, determining how long tasks within loops are repeated under certain conditions.

The following example demonstrates the process impacts:

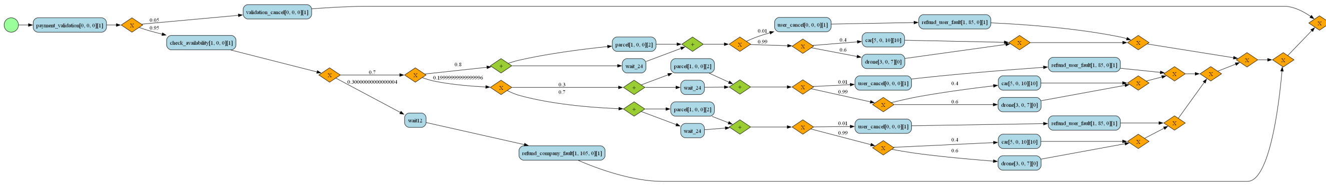


Figure 4.4: Process example for B_1

Node/Process	**Money/Cost (EUR)**	**Electricity (kWh)**	**Work Hours (h)**
`payment_validation`	0	0	0
`check_availability`	1	0	0
`parcel`	1	0	0
`validation_cancel`	0	0	0
`user_cancel`	0	0	0
`refund_user_fault`	1	85	0
`car`	5	0	10
`drone`	3	0	7
`wait_12`	0	0	0
`wait_24`	0	0	0
`refund_company_fault`	1	105	0

Table 4.1: Impact vectors for tasks in B_1 (Figure 4.4)

Similarly, the second business process, B_2, is depicted as follows:

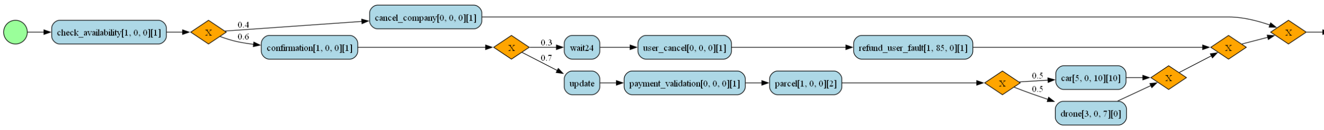


Figure 4.5: Process example for B_2

For this illustration, the two business processes and code used can be found in our GitHub repository. The code demonstrates how we implement the algorithm with an example.

4.6 Conclusion

This chapter introduced a novel approach for computing business process similarity based on resource consumption impacts. By integrating resource dimensions into similarity measures, the methodology provides a comprehensive tool for process analysis and optimization. Future work will explore real-time applications and multidimensional resource impacts.

Node/Process	**Money/Cost (EUR)**	**Electricity (kWh)**	**Work Hours (h)**
`check_availability`	1	0	0
`confirmation`	1	0	0
`cancel_company`	0	0	0
`user_cancel`	0	0	0
`payment_validation`	0	0	0
`car`	5	0	10
`drone`	3	0	7
`parcel`	1	0	0
`wait_24`	0	0	0
`update`	10	0	0

Table 4.2: Impact vectors for tasks in B_2 (Figure 4.5)

Chapter Five

Conclusion and Recommendation

5.1 Implications for Business Process Management

In this dissertation, I investigated the problem of how to determine compliance of business processes concerning a set of constraints, superimposed a priori onto the execution of the process itself, related to the *impact* of the execution. I show that when a constraint superimposition is limiting one single type of impact, then the problem can be solved polynomially on deterministic machines, but when the number of constraints is two or more, the problem is NP-complete. I also provided heuristics to pre-evaluate a process, to establish whether it is decidable in a shorter time, determined by its structure. I tested these results against a dataset of randomly generated business processes and have proven that the behavior is, in the experiments, fits the theoretical expectations.

I am interested in a wide range of developments, especially in agriculture, industrial production, and finance, where the aforementioned notion of impact can be fruitfully applied.

A case that I need to consider of specific theoretical interest arises when we consider non-additive constraints (for instance those that act on an entire trace). Consider, for instance, the cold chain constraint: when transporting certain goods it is mandatory to keep them below 20 degrees below zero. The constraint cannot be expressed in the formalism we have shown. It could be the case that a mixed formulation of these kinds of constraints results easier, computationally speaking, than a pure additive one.

On the opposite side, I am interested in studying two extensions, that relate constraints to the form of the business process. The first extension regards the interaction between the impact constraints and the execution of tasks. I may have, for instance, an effect of the impact that depends in a non-linear way on the task itself. Certain consumption curves exhibit a worse impact when starting, while a long-lasting process can be cheaper than a repetition of short leaps. The second extension aims at introducing the notion of *resource*, not intended only as a source of the production process, but also as a result of the process itself. Processes that involve both resource consumption and emissions are more complicated to deal with than those with only impact and the algorithmic methods investigated in this paper may be insufficient to compute the correct solutions. This aims at applying the notions I developed here in *circular economy*applications where waste minimization and recycling are key.

In the specific interest I devised above, we also have a special focus upon the topic of the relationship with other formalisms, especially when intended to give account to the correspondence between a model of the functioning of a complex organization, as intended for business processes, and the actual implementation of this as a means to realize the digital twin of the business process itself. There is an evident similarity, that does not match directly on the structure of the problem as we provided here, with the Next Release Problem in Software Engineering [3]. The problem is formulated as the task of identifying a set of customers whose requirements can be met in a way that ensures that the associated cost, either in terms of money or workload, remains below a specified bound. I can see this as a generalization of the problem of bounding a business process with single values, that in many senses could be seen as based on a single user. However, there are aspects, in particular related to the complexity of representing the structure of the business process that cannot be mapped in the Next Release Problem, and therefore this may need an extension of the two formalisms to capture both meanings.

An even more advanced version of the Next Release Problem, the multi-objective Next Release Problem, is provided in [10]. In that case, it is not easy to map the two problems one to each other, but I envision a correspondence in a common evolution of the formalisms used to represent the evolution of software and the Business Process, something that could further be used to devise the evolution of Business Processes, a problem that is certainly related to issues in impact, such as reducing the environmental impact, the workload, or the overall cost of a business process.

I have also explored the significance of business process similarity and its impact on organizational efficiency, best practices, and compliance. Traditional approaches focus on structural and behavioral aspects, such as control flow and execution traces, to measure similarity. However, these methods often overlook the critical role of resource consumption in process optimization.

To address this gap, I introduced a novel approach that integrates resource consumption into business process similarity calculations. By leveraging a matrix representation of resource utilization, I proposed two refined similarity measures: a modified cosine similarity and an Euclidean-based similarity metric. Additionally, I developed an algorithm that enables the computation of resource impacts in business process models represented using BPMN.

This approach provides a more comprehensive perspective on business process similarity, allowing organizations to compare processes not only in terms of their structure and behavior but also in relation to their efficiency in resource utilization. This contribution enhances process analysis by offering insights that facilitate resource optimization, cost reduction, and improved decision-making.

5.2 Limitations and Future Research Directions

Future research can extend this work by integrating additional factors such as dynamic process changes, real-time resource monitoring, and adaptive similarity measures that evolve with operational demands. Furthermore, applying these methods to real-world case studies will provide further validation and practical insights into their applicability in various industries.

By incorporating resource consumption into business process similarity measures, this work advances the field of Business Process Management, offering a more holistic framework for process analysis and optimization.

Bibliography

[1] E. C. Akrida, J. Gzyzowicz, L. Gasieniec, L. Kuszner, and P. G. Spirakis. Temporal flows in temporal networks. *Journal of Computer and System Sciences*, 103:46–60, 2019.

[2] D. Ardagna, C. Cappiello, M. Lovera, B. Pernici, and M. Tanelli. Active energy-aware management of business-process based applications. *LNCS (including subseries LNAI and Lecture Notes in Bioinformatics)*, 5377 LNCS:183–195, 2008.

[3] A.J. Bagnall, V.J. Rayward-Smith, and I.M. Whittley. The next release problem. *Information and Software Technology*, 43(14):883 – 890, 2001.

[4] M. Baumann, M.H. Baumann, S. Schönig, and S. Jablonski. Resource-aware process model similarity matching. *LNCS (including subseries LNAI and Lecture Notes in Bioinformatics)*, 8954:96–107, 2015.

[5] S. Bisogno, A. Calabrese, M. Gastaldi, and N. Levialdi Ghiron. Combining modelling and simulation approaches: How to measure performance of business processes. *Business Process Management Journal*, 22(1):56–74, 2016.

[6] Marcus Bolsinger, Andreas Elsässer, Christoph Helm, and Maximilian Röglinger. Process improvement through economically driven routing of instances. *Business Process Management Journal*, 21(2):353–378, 2015.

[7] C. Cabanillas. Enhancing the management of resource-aware business processes. *AI Communications*, 29(1):237–238, 2016.

[8] C. Cabanillas, M. Resinas, and A. Ruiz-Cortés. Defining and analysing resource assignments in business processes with ral. *LNCS (including subseries LNAI and Lecture Notes in Bioinformatics)*, 7084 LNCS:477–486, 2011.

[9] C. Cappiello, M.G. Fugini, G.R. Gangadharan, A.M. Ferreira, B. Pernici, and P. Plebani. First-step toward energy-aware adaptive business processes. *LNCS (including subseries LNAI and Lecture Notes in Bioinformatics)*, 6428 LNCS:6–7, 2010.

[10] José M. Chaves-González and Miguel A. Pérez-Toledano. Differential evolution with pareto tournament for the multi-objective next release problem. *Applied Mathematics and Computation*, 252:1 – 13, 2015.

[11] Haifang Cheng, Shimiao Jiang, and Danfeng Zou. Risk identification of public infrastructure projects based on vfpe. In *2010 The 2nd International Conference on Computer and Automation Engineering, ICCAE 2010*, volume 4, page 173 – 176, 2010.

[12] M. De Leoni, W.M.P. Van Der Aalst, and B.F. Van Dongen. Data- and resource-aware conformance checking of business processes. In *Lecture Notes in Business Information Processing*, volume 117 LNBIP, pages 48–59. Springer Verlag, 2012.

[13] A. Del-Río-Ortega, M. Resinas, C. Cabanillas, and A. Ruiz-Cortés. Defining and analysing resource-aware process performance indicators. In *CEUR Workshop Proceedings*, volume 998, pages 57–64, 2013.

[14] Peter Delfmann, Daniel Breuker, Martin Matzner, and Jörg Becker. Supporting information systems analysis through conceptual model query - the diagramed model query language (dmql). *Communications of the Association for Information Systems*, 37(1):24, 2015.

[15] R. Dijkman, M. Dumas, B. van Dongen, R. Käärik, and J. Mendling. Similarity of business process models: Metrics and evaluation. *Information Systems*, 36(2):498–516, 4 2011.

[16] Marlon Dumas, Luciano García-Bañuelos, and Artem Polyvyanyy. Unraveling unstructured process models. In Jan Mendling, Matthias Weidlich, and Mathias Weske, editors, *Business Process Modeling Notation - Second International Workshop, BPMN 2010, Potsdam, Germany, October 13-14, 2010. Proceedings*, volume 67 of *Lecture Notes in Business Information Processing*, pages 1–7. Springer, 2010.

[17] Marlon Dumas, Luciano García-Bañuelos, and Artem Polyvyanyy. Unraveling unstructured process models. In *Business Process Modeling Notation: Second International Workshop, BPMN 2010, Potsdam, Germany, October 13-14, 2010. Proceedings 2*, pages 1–7. Springer, 2010.

[18] Javier Esparza, Stefan Römer, and Walter Vogler. An improvement of mcmillan's unfolding algorithm. *Formal Methods in System Design*, 20:285–310, 2002.

[19] Markus Fellmann and Andreas Zasada. State-of-the-art of business process compliance approaches - a survey. *EMISA Forum*, 36(2):45–48, 2016.

[20] S.R. Gohar and M. Indulska. Environmental sustainability through green business process management. *Australasian Journal of Information Systems*, 24, 11 2020.

[21] G. Governatori, F. Olivieri, S. Scannapieco, and M. Cristani. Designing for compliance: Norms and goals. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7018 LNCS:282–297, 2011.

[22] G. Governatori, F. Olivieri, S. Scannapieco, A. Rotolo, and M. Cristani. The rationale behind the concept of goal. *Theory and Practice of Logic Programming*, 16(3):296–324, 2016.

[23] M. Hashmi, Guido Governatori, H. P. Lam, and M. T. Wynn. Are we done with business process compliance: state of the art and challenges ahead. *Knowledge and Information Systems*, 57(1):79–133, 2018.

[24] Keijo Heljanko, Victor Khomenko, and Maciej Koutny. Parallelisation of the petri net unfolding algorithm. In Joost-Pieter Katoen and Perdita Stevens, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 371–385, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[25] K. Hoesch-Klohe, A. Ghose, and L.S. Le. Towards green business process management. In *Proceedings - 2010 IEEE 7th International Conference on Services Computing, SCC 2010*, pages 386–393, 2010.

[26] K. Kammerer, R. Pryss, and M. Reichert. Context-aware querying and injection of process fragments in process-aware information systems. In *Proceedings - 2020 IEEE 24th International Enterprise Distributed Object Computing Conference, EDOC 2020*, pages 107–114, 2020.

[27] F. Klessascheck. Towards a framework for business process sustainability analysis. In *CEUR Workshop Proceedings*, volume 3113, pages 6–10, 2022.

[28] Richard E. Korf. A complete anytime algorithm for number partitioning. *Artificial Intelligence*, 106(2):181–203, 1998.

[29] B. López, A. Ghose, T. Savarimuthu, M. Nowostawski, M. Winikoff, and S. Cranefield. Towards energy-aware optimisation of business processes. In *SMARTGREENS 2014 - Proceedings of the 3rd International Conference on Smart Grids and Green IT Systems*, pages 68–75, 2014.

[30] P. Lübbecke, P. Fettke, and P. Loos. Towards guidelines of modeling for ecology-aware process design. *Lecture Notes in Business Information Processing*, 308:510–519, 2018.

[31] Dina Neiger, Leonid Churilov, Michael Zur Muehlen, and Michael Rosemann. Iintegrating risks in business process models with value focused process engineering. In *Proceedings of the 14th European Conference on Information Systems, ECIS 2006*, pages 1–10, 2006.

[32] A. Nowak, T. Binz, C. Fehling, O. Kopp, F. Leymann, and S. Wagner. Pattern-driven green adaptation of process-based applications and their runtime infrastructure. *Computing*, 94(6):463–487, 2012.

[33] A. Nowak and F. Leymann. Green business process patterns - part ii. In *Proceedings - IEEE 6th International Conference on Service-Oriented Computing and Applications, SOCA 2013*, pages 168–173, 2013.

[34] F. Olivieri, M. Cristani, and G. Governatori. Compliant business processes with exclusive choices from agent specification. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9387:603–612, 2015.

[35] F. Olivieri, G. Governatori, S. Scannapieco, and M. Cristani. Compliant business process design by declarative specifications. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8291 LNAI:213–228, 2013.

[36] Artem Polyvyanyy. *Process querying: methods, techniques, and applications.* Springer, Cham, 2021.

[37] Artem Polyvyanyy, Chun Ouyang, Alistair Barros, and Wil van der Aalst. Process querying: Enabling business intelligence through query-based process analytics. *Decision Support Systems*, 100:41–56, 2017.

[38] Michael Rosemann. Explorative process design patterns. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 12168 LNCS, page 349 – 367, 2020.

[39] Kristian Rotaru, Carla Wilkin, and Leonid Churilov. Proactive risk identification in cross-organizational business processes. In *Proceedings of the 25th Australasian Conference on Information Systems, ACIS 2014*, pages 1–10, 2014.

[40] Kristian Rotaru, Carla Wilkin, Leonid Churilov, Dina Neiger, and Andrzej Ceglowski. Formalizing process-based risk with value-focused process engineering. *Information Systems and e-Business Management*, 9(4):447 – 474, 2011. Cited by: 16.

[41] T. Schoormann, K. Kutzner, S. Pape, and R. Knackstedt. Elevating social sustainability in business processes: A pattern-based approach. In *40th International Conference on Information Systems, ICIS 2019*, pages 1–30, 2019.

[42] E.R. Taghiabadi, V. Gromov, D. Fahland, and W.M.P. Van Der Aalst. Compliance checking of data-aware and resource-aware compliance requirements. *LNCS (including subseries LNAI and Lecture Notes in Bioinformatics)*, 8841:237–257, 2014.

[43] Arava Tsoury, Pnina Soffer, and Iris Reinhartz-berger. How well did it recover? impact-aware conformance checking. *Computing*, 103:1–25, 01 2021.

[44] W. M. P. van der Aalst, A. K. Alves de Medeiros, and A. J. M. M. Weijters. Process equivalence: Comparing two process models based on observed behavior. In Schahram Dustdar, José Luiz Fiadeiro, and Amit P. Sheth, editors, *Business Process Management*, pages 129–144, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[45] Wil M. P. van der Aalst. *Process Mining - Data Science in Action, Second Edition*. Springer, 2016.

[46] W.M.P. van der Aalst. The application of petri nets to workflow management. *Journal of Circuits, Systems, and Computers*, 8(1):21–66, 1999.

[47] M.T. Wynn, J.D. Weerdt, A. ter Hofstede, W. Van der Aalst, H.A. Reijers, M. Adams, C. Ouyang, M. Rosemann, and W.Z. Low. Cost-aware business process management: A research agenda. In *Proceedings of the 24th Australasian Conference on Information Systems*, pages 3–13, 2013.

[48] Z. Yan, R. Dijkman, and P. Grefen. Fast business process similarity search. *Distributed and Parallel Databases*, 30(2):105–144, feb 24 2012.

[49] A. Zasada. A box of bricks for modelling domain-specific compliance pattern. In *Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOCW*, volume 2018-October, pages 37–42, 2018.

[50] H. Zhang and J. Han. Mathematical models for information classification and recognition of multi-target optical remote sensing images. *Open Physics*, 18(1):951–960, 2021.

List of Publications

1. Tewabe Chekole Workneh, Pietro Sala, Romeo Rizzi, Matteo Cristani, *Business Process Compliance with Impact Constraints*, **Information Systems**, Elsevier, 2024. [*Published*]
2. Tewabe Chekole Workneh, Matteo Cristani, Claudio Tomazzoli, *Assessing the Impact of Climate Change on Mineral-Associated Organic Carbon (MAOC) Using Machine Learning Models*, **AI4CC-IPS-RCRA-SPIRIT 2024:** *International Workshop on Artificial Intelligence for Climate Change, Italian Workshop on Planning and Scheduling, RCRA Workshop on Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion, and SPIRIT Workshop on Strategies, Prediction, Interaction, and Reasoning in Italy*, November 25-28th, 2024, Bolzano, Italy. [*Accepted*]
3. Matteo Cristani, Mattia Zorzan, Tewabe Chekole Workneh, Claudio Tomazzoli, *Data Augmentation for Business Process Alignment: Proof of Concept and Experimental Design*, **KES AMSTA 2024:** *International Conference on Agent and Multi-Agent Systems: Technologies and Applications*. [*Accepted*]
4. Hailemicael Lulseged Yimer, Matteo Cristani, Tewabe Chekole Workneh, Claudio Tomazzoli, *AI-Driven Nitrogen Stress Management in Cereal Crops via Drone Technology*, **KES AMSTA 2024:** *International Conference on Agent and Multi-Agent Systems: Technologies and Applications*. [*Accepted*]