UNIVERSITY OF MACEDONIA
DEPARTMENT OF APPLIED INFORMATICS
GRADUATE PROGRAM

Investigation of Business Process Optimization Using Design of Experiments

M.Sc. THESIS

of

Dimitrios Paganias

Thessaloniki, October 2019

Investigation of Business Process Optimization Using Design of Experiments

Dimitrios Paganias

B.Sc. in Physics, Physics Department
University of Ioannina, 2016

M.Sc. Thesis

submitted as a partial fulfillment of the requirements for

THE DEGREE OF MASTER OF SCIENCE IN APPLIED INFORMATICS

Supervisor: Dr Kostas Vergidis

Approved by examining board on 29 October 2019

Dr Kostas Vergidis	Prof. Dimitrios Hristu-Varsakelis	Dr Sophia Petridou
______________________	______________________	______________________

Abstract

Business process optimization can be considered as the problem of constructing business process design with optimum attributes. Vergidis (2008) proposed an approach for the Evolutionary Multi-objective business process optimization, the business process optimization framework (bpo^F). The bpo^F utilizes as a main component the proposed business process representation and EMOAs in order to generate alternative optimized designs. The business process representation is described by mathematical parameters and the composition of a business process design is based on an algorithm that is named Process Composition Algorithm (PCA). This thesis provides a complete and extended investigation of the business process optimization framework (bpo^F). Employing a series of scalable tests the main elements of bpo^F, are examined in-depth. In addition, this thesis applies the statistical approach of Design of Experiments (DoE) in the business process optimization problem. This approach provides the necessary tools needed to analyze and interpret the results.

Keywords: business process, business process optimization, business process optimization framework (bpo^F), Design of Experiments (DoE)

Acknowledgments

While writing these lines, an intense and significant cycle of my life come to its end.

I would like to thank my supervisor Dr Kostas Vergidis for our great collaboration. I feel truly glad that I met and cooperated with him. Moreover, I really appreciate the interest he has shown me.

I would like to thank my family not only for the support but also for the opportunity they give me to make my dreams come true.

I would like to thank my loyal friends who are always by my side all these years and most significantly my friend Alexandros and his family. I will always remember your help.

Last but not least, I would like to thank my partner, Despoina. Your love gives me the motive to move on.

Table of Contents

List of Figures

List of tables

Chapter 1 – Introduction

This chapter introduces in brief the concept of the business processes, states the aim and the objectives of this research and concludes with the structure of the thesis.

1.1 Introduction to business processes

This section provides a first introduction to the main concept behind this research, the business processes. The first definitions of business processes appeared in literature in the 1990's. Havey (2005) proposed the business processes to be defined as the 'step-by-step rules specific to the resolution of a business problem. Gunasekaran and Gobu (2002) suggested defining the business processes as a group of related tasks that were combined, in order to create value for a customer.

In general, there is a variety of definitions and in their majority can be considered similar in terms of concepts that were used in order to describe business processes. Despite this fact, a lot of criticism has been raised for not adequately highlighting the business component in the definitions and not sufficiently distinguishing them from manufacturing or production processes. Volkner and Werners (2000) pointed out that there is not a generally accepted definition for the business processes as a result of the different disciplines that have approached them. Two issues that Vergidis (2008) noticed for the business process definitions are the following:

- They are simplistic and basic thus too generic to provide any tangible contribution
- They are confined to a very specific application area

In order to answer these two issues Vergidis (2008) proposed the business processes to be perceived as a collective set of tasks that when properly connected and sequenced perform a business operation and their aim is to perform a business operation.

In contrast to the wide variety of definitions in the literature, the main elements of the business processes have a common way to be perceived. Vergidis (2008) presented a business process schema in order to involve the most common elements found in literature (Figure 1.1). Particularly, he depicted with solid arrows the main elements and the optional ones with dashed arrows. On the top of the schema he placed the generic processes and below the business processes. In this way he wanted to show that business processes inherit all the main properties such as resources, from the generic processes. Parallel to business processes he placed the workflow in order to show that they are linked and many times interchangeable. Under these two he placed the actors, the activities and the resources because these three concepts were included in the most process definitions. The main remarks that he added for these three concepts are the following:

- The Actors are sometimes involved in a business process definition (Lindsay et al., 2003) or sometimes perceived as external entities that enact or execute the process.
- The Activities are widely accepted as the central elements that execute the basic business process steps utilizing the process inputs in order to produce the desired outputs.
- The Resources are frequently classified as inputs or output resources that are required for the execution of the activities.

Finally, below these activities he placed the tasks in order to show that most of the authors consider them as synonyms for activities.

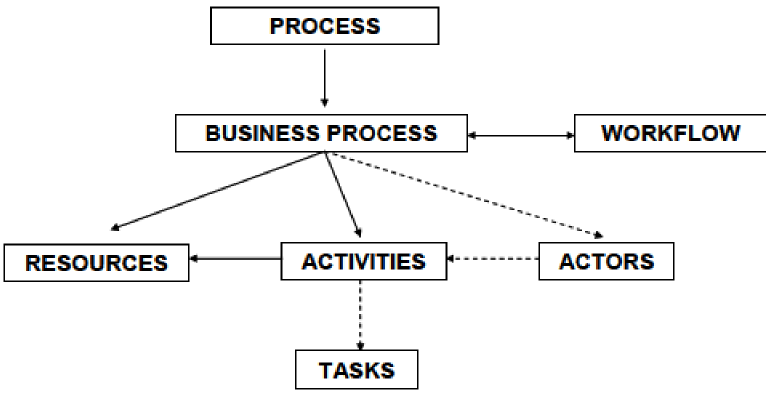


Figure 1.1. Schematic relationship of the main business process elements by Vergidis (2008)

1.2 Aim and objectives

This section states and discusses the aim and the objectives of this research.

Research aim

The aim of this thesis is the systematic investigation of the business process optimization problem parameters as they were introduced at the business process optimization framework (bpo^F) (Vergidis,2008). For this reason, a series of scalable business process test problems are employed. In addition, a notable novelty in this research is the application of a statistical approach named Design of Experiments (DOE) in the Business Process Optimization (BPO) problem. This approach provides not only tools needed to analyze and interpret the results but also strategies to design an experiment and collect data.

Research objectives

The main objectives of this thesis can be summarized as follows:

1. Studying and understanding Business Process Optimization
2. Reviewing the results of BPO^F reported by Vergidis (2008)
3. Studying and understanding Design of Experiments (DoE)
4. Determination of the problem parameter limits that generate reliable results
5. Characterization of the problem parameters significance & their influence on the results

1.3 Thesis structure

The rest of the thesis is organized in six chapters. The second chapter presents a review of the literature about the main concepts around business processes. In the beginning it provides a summary of the various definitions that are met in the literature for the business processes. Next, it introduces the two major groups of business process modeling and it closes with the evolutionary business process optimization approaches that are found in the literature. The third chapter details the business process optimization framework (bpo^F) that was proposed by Vergidis (2008). In addition, the results that were reported by Vergidis (2008) are reviewed. The fourth chapter introduces the Design of Experiments (DoE), a statistical method which determines the relationship between factors affecting a process and the output of that process. Particularly, the statistical approaches for identifying the significance of the process factors and the way in which these factors influence the output of a process are analyzed. The fifth chapter presents a series of scalable tests in order to investigate the parameters of the business process optimization problem that was proposed by Vergidis (2008). The limits of the parameters in which the bpo^F generates reliable results are determined. The sixth chapter specifies the way in which the parameters of the business process optimization problem influence the results that the bpo^F generates. In order to do this, the Design of Experiments (DoE) is employed. Finally, the seventh chapter summarizes this research, discusses the contribution and the limitations of this research and provides suggestions for future work.

Chapter 2 - Literature review

This chapter discusses the main concepts around business processes. The literature review focuses on the aspects of definition, modeling and evolutionary optimization of business processes. Thus, in this chapter an overview of the most common business process definitions, of the most significant business process modeling techniques as well as the business process optimization approaches existing in literature are provided in order to highlight their strengths and weaknesses.

2.1 Business process definitions

This section introduces the various business process definitions existing in literature. The reason behind such diversity is that every author describes a business process model highlighting only specific aspects based on the field of study he comes from. It is worth mentioning that there is not such a definition globally accepted and none of the existing definitions prevails over the others. As Shen et al. (2004) stating, each business process definition attempt has its own advantages and disadvantages but what remains the same is that each method is used to represent a certain view of enterprise. The aim of this section is to provide an insight towards the main concepts around business processes and clarify how these are perceived by the authors.

The first definitions of business processes appeared in literature in the 1990's and almost any of them seems to be an improved version of the business process definitions provided by Hammer and Champy (1993) and Davenport (1993). Melao and Pidd (2000) and Tinnila (1995) have gathered such definitions which are provided in Table 2.1. This table depicts the diversity of the existing business process definitions in literature.

Author(s)	Business process definitions
Agerfalk (1999)	A *business process* consists of activities ordered in a structured way with the purpose of providing valuable results to the customer.
Castellanos *et al.* (2004)	The term *business process* is used to denote a set of activities that collectively achieve a certain business goal. Examples of these processes are the hiring of a new employee or the processing of an order.

Davenport and Short (1990)	*Business process* is a set of logically related tasks performed to achieve a defined business outcome.
Davenport (1993)	*Business process* is defined as the chain of activities whose final aim is the production of a specific output for a particular customer or market
Fan (2001) Shen *et al.* (2004)	*Business process* is a set of one or more linked procedures or activities that collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships.
Gunasekaran and Kobu (2002)	A group of related tasks that together create value for a customer is called a *business process.*
Hammer and Champy (1993)	A *business process* is a collection of activities that takes one or more kinds of inputs and creates an output that is of value to the customer. A business process has a goal and is affected by events occurring in the external world or in other processes.
Irani *et al.* (2002)	A *business process* is a dynamic ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs.
Johanson *et al.* (1993)	A business process is a set of linked activities that takes an input and it transforms it to create an output. It should add value to the input and create an output that is more useful and effective to the recipient.
Pall (1987)	Business process is the logical organization of people, materials, energy, equipment and procedures into work activities designed to produce a specified end result.
Soliman (1998)	*Business process* may be considered as a complex network of activities connected together.
Stock and Lambert (2001)	A *business process* can be viewed as a structure of activities designed for action with focus on the end customer and the dynamic management of flows involving products, information, cash, knowledge and ideas.
Stohr and Zhao (2001)	A *business process* consists of a sequence of activities. It has distinct inputs and outputs and serves a meaningful purpose within an organization or between organizations.
Volkner and Werners (2000)	*Business process* is defined as a sequence of states, which result from the execution of activities in organizations to reach a certain objective.
Wang and Wang (2005)	*Business process* is defined as a set of business rules that control tasks through explicit representation of process knowledge.

Vergidis (2008)	*Business process* is a collective set of tasks that when properly connected and sequenced perform a business operation. The aim of a business process is to perform a business operation, i.e. any service-related operation that produces value to the organization.

Table 2.1. Business process definitions existing in literature

As it seems from table 2.1 and stated before, most definitions are somewhat related to those by Davenport (1993) and Hammer and Champy (1993). The differences found among them, rely on the emphasis than the authors give to specific aspects of business processes and all of them except Vergidis (2008), have received criticisms for not sufficiently identifying the business component and not clearly distinguishing them to manufacturing or production processes. Agerfalk (1999), Davenport (1993), Hammer and Champy (1993), Stock and Lambert (2001) and Gunasekaran and Kobu (2002), provide more customer-oriented definitions. Castellanos et al. (2004), Fan (2001) and Shen et al. (2004) emphasize on the goal orientation of a business process. Agerfalk (1999) sees business processes as an ordered structure of activities. Pall (1987) who has provided one of the earliest definitions, also involves the human factor in the context of business processes along with the material resources and sees business processes as a structure of all of them logically connected. The term of logical connection is also referred in the definition provided by Davenport and Short (1990) and the term of proper connection and sequence, which is similar, by Vergidis (2008). On the other hand, Soliman (1998) identifies the complexity that a business process may have through his definition. Stock and Lambert (2001) and Irani et al. (2002) point out the necessity of clearly identified inputs and outputs for a business process. Hammer and Champy (1993) highlight the fact that a business process may be affected by the external world or the execution of other processes. Additionally, Stock and Lambert (2001) imply through their definition, that the management of activities and resources participating in a business process may alter dynamically. Furthermore, there are also two definitions worth mentioning, coming from Volkner and Werners (2000) and Wang and Wang (2005) respectively. The first one, emphasizes on states as the main structural elements of a business process. This attempt provides a different insight into business processes as evolving series of states that modify the result of the execution of the participating activities. The second one, introduces business processes as a set of rules that control tasks; unfortunately, without mentioning who is in charge for executing these tasks and if they have an ordered structure.

2.2 Business process modeling

Business process modeling (BPM) in business process management and systems engineering is the activity of representing processes of an enterprise, so that the current process may be analyzed, improved and automated. The context of business process modeling indicates and facilitates the level of perception and understanding of business processes within a company. As human beings, we can process and understand things better if we can see them. Therefore, the elements and the capabilities of a business process model play a significant role in the business world. According to Luttighuis et al. (2001) a main objective of business process intelligence is to provide an insight in the structure of business processes and the relation among them. This insight can be easily obtained by creating business process models that clearly and precisely illustrate the essence of the business organization. These models should contain organizational level details, capabilities for easily identifying bottlenecks and quick assessment of the consequences of a potential change to the customers and the organization itself. According to van der Aalst et al. (2003), business process modeling is used to characterize the identification and specification of business processes. Business process modeling includes modeling of activities and their causal and temporal relationships as well as specific business rules that process activities must comply with. Lindsay et al. (2003) describe business process modeling as a snapshot of what is perceived at a point of time regarding the actual business process. The objective of business process modeling, as provided by Sadiq and Orlowska (2000), is the high-level specification of processes, while Biazzo (2002) says that it is the representation of relationships between the activities, people, data and objects involved in the production of a specified output. Volkner and Werners (2000) and Aguilar-Saven (2004) claim that business process modeling is essential for the analysis, evaluation and improvement of business processes as it is used to structure the process, such that the existing and alternative sequence of tasks can be analyzed systematically and comprehensively. As Guha et al. (1993) and Abate et al. (2002) state, business process modeling is a useful tool to capture, structure and formalize the knowledge about business processes. Aguilar-Saven (2004) suggest that business process models are mainly used to learn about the process, to make decisions on the process, or to develop business process software. Shen et al. (2004) supports that business process modeling is an essential part of developing an enterprise information system. According to Vergidis (2008), the business process design is the representation of a business process depicting the participating tasks and their connectivity patterns that determine the flow of the process. The aim of such a design, is to capture, visualize and communicate a business process.

In this section, we provide an overview of the most significant business process modelling techniques existing in literature. The necessity behind an overview like this, is because business process models are mainly used either to learn about the business process itself, as stated before, or to make decisions on the process or to develop business process software. As it is evident, such purposes involve an extension over some model characteristics. Considering these characteristics, the main modeling concepts can be classified into two major groups. The first classification can be formed by the modeling techniques using a visual diagram, called as diagrammatic models. On the other hand, the second classification corresponds to models consisting of elements that have a mathematical or a formal basis. Both classifications are presented below along with the most representative examples of each of them.

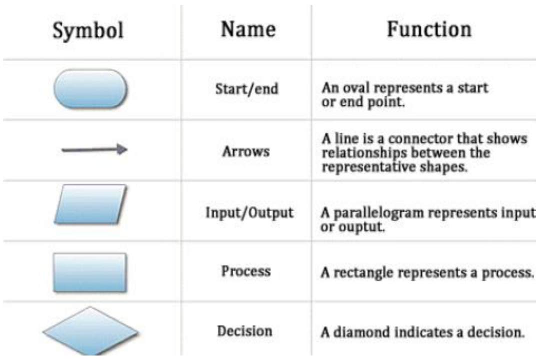
Diagrammatic models

The first and most straightforward business process modeling techniques were plain graphical representations and were initially developed for software specification ((Knuth, 1963), (Chapin N., 1971)). The main characteristic of such techniques is the common approach to depict a business process by using a diagram with defined notation e.g. shapes, lines, arrows etc. These diagrammatic techniques have the prominent advantage of illustrating the business process, hence making it easy to follow and understand without the need of any technical expertise. However, if there is no universal standard notation and methodology used, this can lead to misunderstandings about a business process model among people (Havey, 2005). For this reason, BPMN which stands for Business Process Model and Notation, has been developed and is mainly used nowadays among businesses. BPMN will be further discussed later in this section. The main business process modeling techniques are as follows:

1. Flowchart technique

   The Flowchart model is probably the first and most popular process notation since it has frequently been used over many years to represent algorithms, workflows and processes. Flowchart is defined by Lakin et al. (1996) as a formalized graphic representation of a program logic sequence, work or manufacturing process, organization chart or similar formalized structure. Flowcharts consist of special symbols representing different types of actions or steps in a process, along with lines and arrows indicating the sequence of steps, and the relationships among them. The basic symbols of a Flowchart are represented in Figure 2.1.

   Figure 2.2 represents an example of the flowchart technique using the symbols presented in Figure 2.1 and demonstrates the simplicity of this technique. The main advantages of this method are the very easy follow-up of the described process, the quick and easy drawing, the flexibility it provides, as a process can be described in various ways, and the communication ability provided by the standard notation. For example, the process described in Figure 2.2 can be easily understood as a medical service process, despite the label, and it seems that no special effort made to draw it.



Symbol	Name	Function
	Start/end	An oval represents a start or end point.
	Arrows	A line is a connector that shows relationships between the representative shapes.
	Input/Output	A parallelogram represents input or ouptut.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.

Figure 2.1. Basic elements of a flowchart

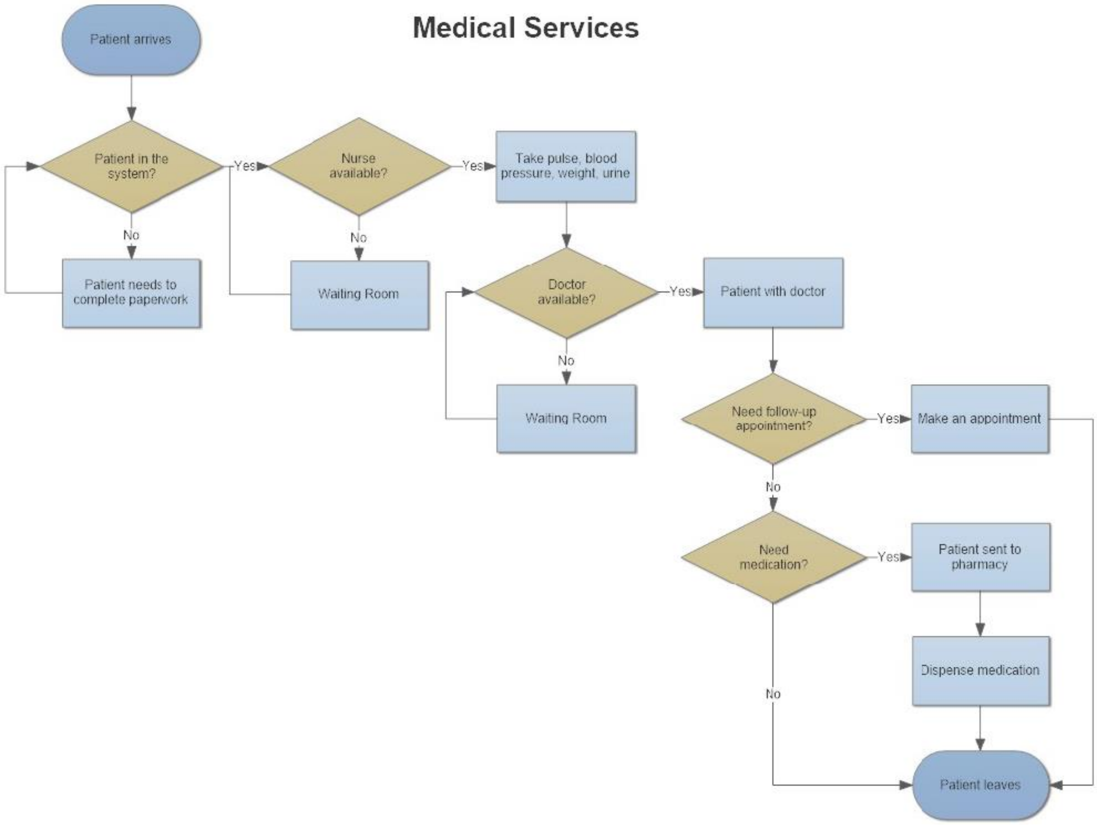


Figure 2.2. Example of process using a flowchart

On the contrary, flowcharts may become too large in effort to capture more and more information within it, hence more difficult to be read. In addition, most of the times, the flexibility comes with no standard methodology and the boundaries of a business process may become unclear. For example, someone could draw the models of all sub-processes of Figure 2.2 within the same model, making it very large and difficult to read. Someone could also draw another model for the process of Figure 2.2 by setting another step between nurse availability and doctor availability to check for doctor availability of other specialty to take pulse, blood pressure, weight and urine.

To sum up, the best use of the flowchart model technique is for the high-level understanding of a business process and if someone needs to provide much information and many details about a business process, he must choose another modeling technique.

2. Integrated Definition for Function Modeling (IDEF)

The lack of a standard methodology and necessary semantics to support more complex and standardized structures in the flowchart technique, led to the development of standard methodologies such as IDEF and Unified Modeling Language (UML) for process modeling and/or software development. We are going to discuss UML later in this section.

IDEF is a family of modeling languages in the field of systems and software engineering, capable of graphically representing a wide range of business, manufacturing and other types of enterprise operations to any level of detail. According to Kim et al. (2003), IDEF provides a suite of graphical modeling techniques designed to specify and communicate important aspects of business processes. IDEF was initially developed by US Air Force Materials Laboratory in the mid-1970s as a part of the Integrated Computer-Aided Manufacturing (ICAM). The ICAM program office deemed it valuable to create a “neutral” way of describing the data content of large-scale systems and proceeded with developing methods for processing data independently of the way it was physically stored. The IDEF methods are classified according to the applications they are used. Table 2.2 shows the scope of each method of IDEF family. In the context of this dissertation we are going to discuss the IDEF0 and IDEF3 methods since these are related to process modeling.

Method	Scope
IDEF0	Function modelling
IDEF1	Information modelling
IDEF1X	Data modelling
IDEF2	Simulation model design
IDEF3	Process description capture
IDEF4	Object-oriented design
IDEF5	Ontology description capture
IDEF6	Design rationale capture
IDEF7	Information system auditing
IDEF8	User interface modelling
IDEF9	Business constraint discovery
IDEF10	Implementation architecture modelling
IDEF11	Information artefact modelling
IDEF12	Organization modelling
IDEF13	Three schema mapping design

IDEF14	Network design

Table 2.2. IDEF methods

IDEF0 is a functional modeling method designed to model the decisions, actions and activities within an organization or system. It is used for analyzing, communicating and understanding the functional perspective of a system and the relationships within it. For example, where a flowchart model is used to show the functional flow of a process, IDEF0 is used to show data flow, system control, and the functional flow of lifecycle processes. IDEF0 models consist of a hierarchical series of diagrams, text and glossary cross-referenced to each other. The two primary modeling components are the functions, represented by boxes, and the data and objects that inter-connect those functions, represented by arrows. The basic syntax for an IDEF0 model is shown below in the Figure 2.3.

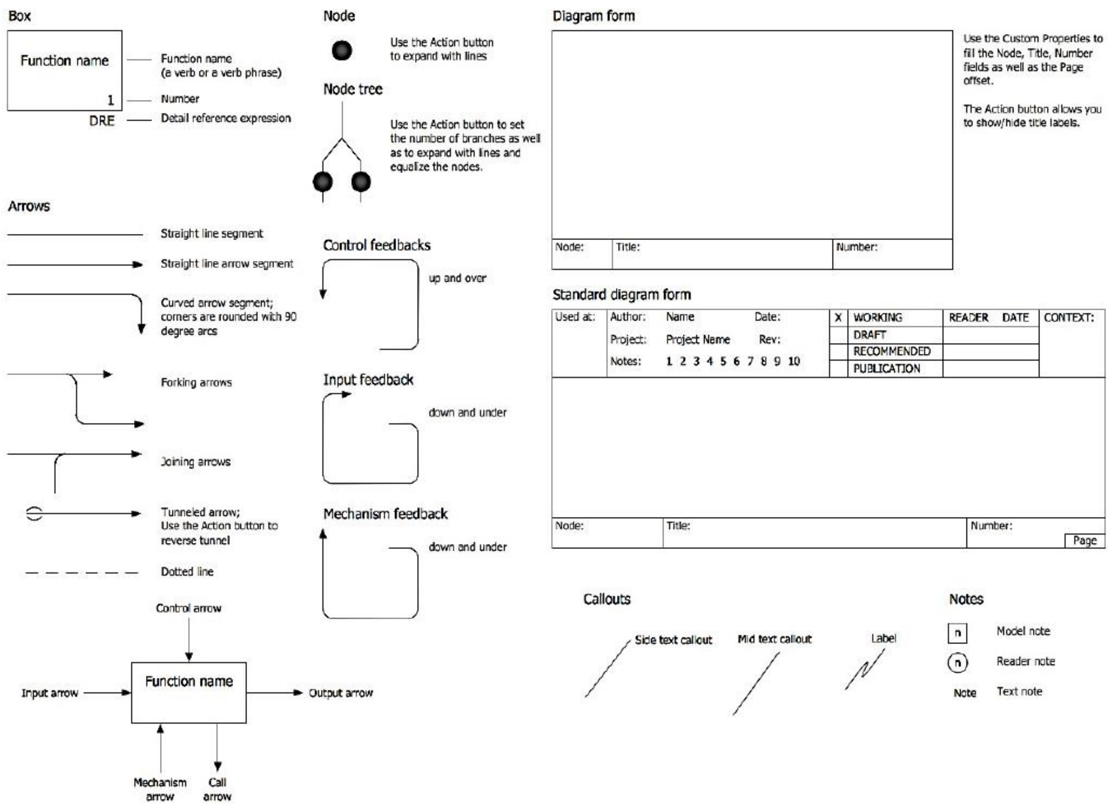


Figure 2.3. IDEF0 basic syntax

An IDEF0 process starts with the identification of the prime function to be decomposed. This function is identified on a "Top Level Context Diagram" that defines the scope of a particular IDEF0 analysis. An example of a "Top Level Context Diagram" is illustrated in Figure 2.4.

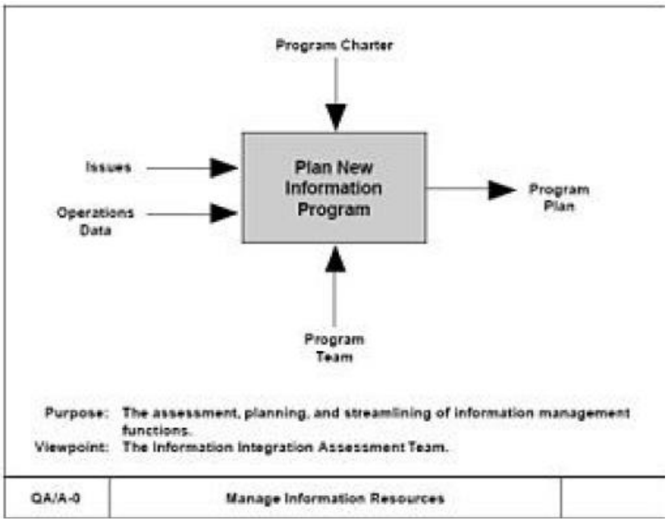


Figure 2.4. Example of Top Level Context Diagram

Then, the prime function can be logically decomposed into its component functions. This process can be continued recursively to the desired level of detail. An example of the IDEF0 process decomposition is presented in Figure 2.5 below.

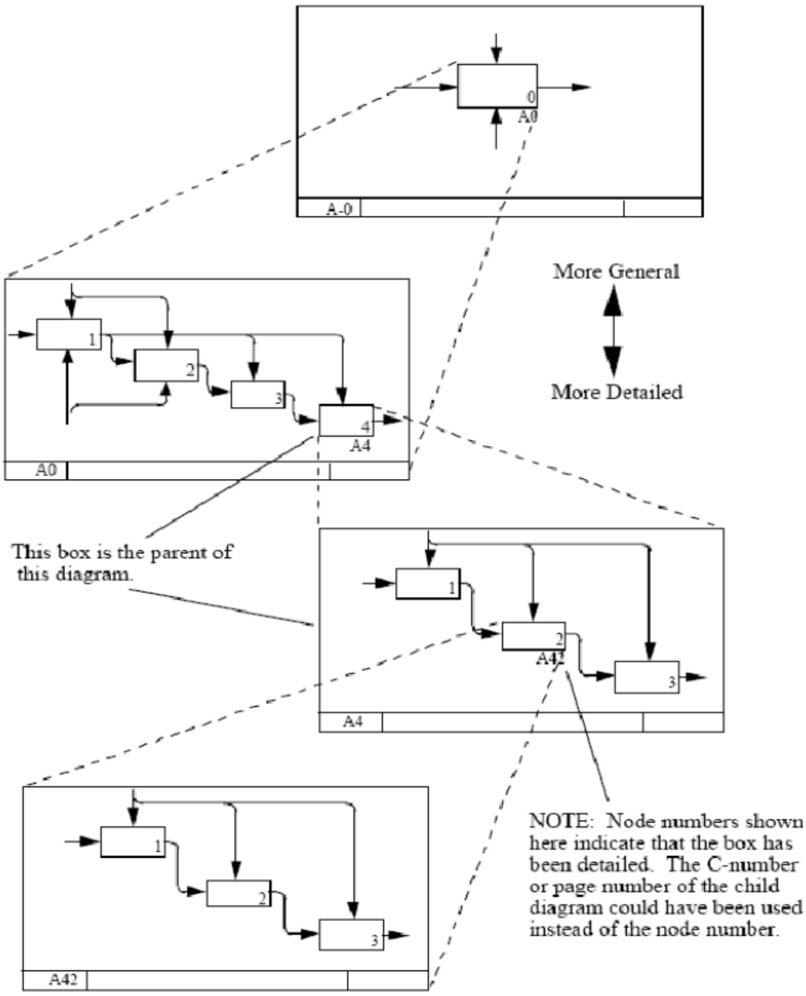


Figure 2.5. Example of the IDEF0 process decomposition

One of the strengths of IDEF0 modeling technique is the extended level of detail that can be provided, making the model as descriptive as necessary for the decision-making task to be at hand. Additionally, another strength of IDEF0 emerges from its hierarchical nature by facilitating the development of (AS-IS) models that have a top-down representation and interpretation, but which are based on a bottom-up analysis process.

One the other hand, one potential disadvantage comes from the level of detail described in an IDEF0 model and if it is very concise, it may be understandable only from readers be domain experts or have been participated in the model development. In addition, another weakness is the tendency of IDEF0 models to be interpreted as representing a sequence of activities even though IDEF0 is not intended to be used for modeling activity sequences. The activities may be placed in a left to right sequence within a decomposition and connected with the flows. It is natural to order the activities left to right because, if one activity outputs a concept that is used as input by another activity, drawing the activity boxes and concept connections is clearer. The solution to this weakness has been given by IDEF3 which is described below.

IDEF3 is a process description capture method to capture descriptions of sequences of activities, which is considered the common mechanism to describe a situation or process. It is a business process modeling method complementary to IDEF0. The difference between IDEF0 and IDEF3 is that the former shows what is done within an organization or system while the latter shows how things work with it. IDEF3 provides a mechanism for collecting and documenting processes. It captures the precedence and causality relations between situations and events in a form natural to domain experts by providing a structured method for expressing knowledge about how a system, process, or organization works. The basic organizing structure for IDEF3 process descriptions is the notion of scenario. A scenario can be thought as a recurring situation, or a set of situations that describe a typical class of problems addressed by an organization or system, or the setting within which a process occurs. Scenarios establish the focus and boundary conditions of a description and humans must describe what they know in terms of an ordered sequence of activities within the context of the given scenario or situation.

IDEF3 provides two description modes: The Process Flow Description which captures the knowledge of "how things work" in an organization or system and the Object State Transition Network Description which summarizes the allowable transitions of an object throughout a particular process. Both the Process Flow Description and Object State Transition Network Description contain units of information that make up the system description. These model entities, as they are called, form the basic units of an IDEF3 description. The resulting diagrams and text comprise what is termed a "description" as opposed to the focus of what is produced by the other IDEF methods whose product is a "model." The basic syntax for an IDEF3 process description is shown below in the Figure 2.6.a and Figure 2.6.b.

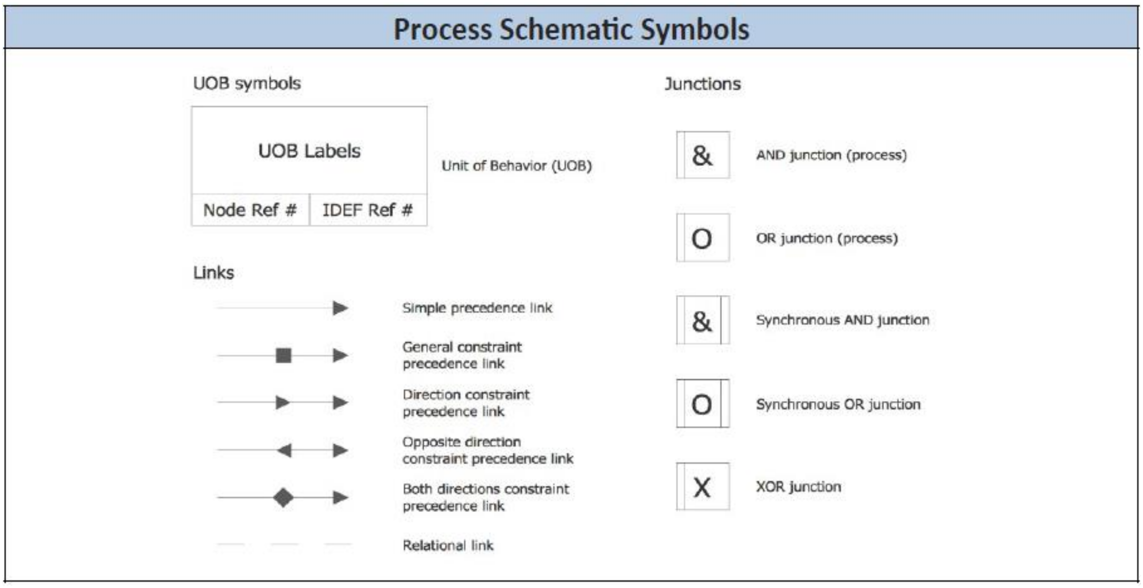


Figure 2.6.a. Basic syntax for an IDEF3 process

An example of IDEF3 description of a process using the process flow description and the object state transition description is shown in Figure 2.7 and Figure 2.8 respectively.

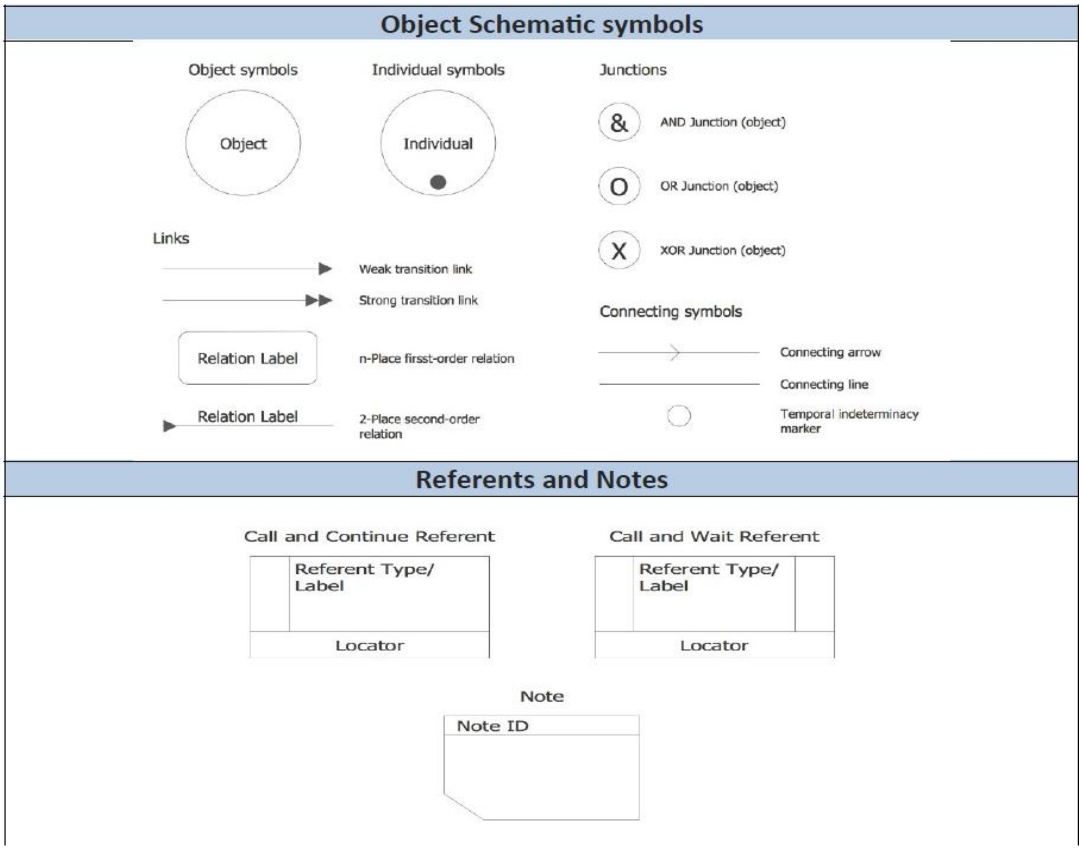


Figure 2.6.b. Basic syntax for an IDEF3 process

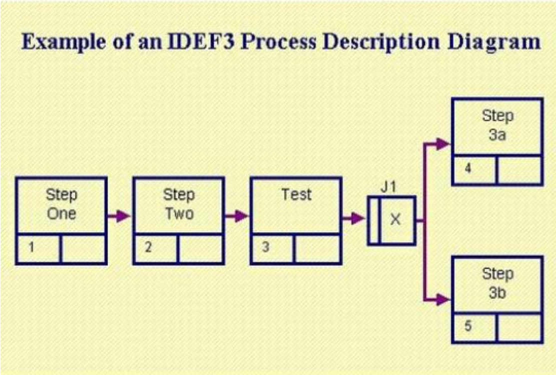


Figure 2.7. Example of process flow description

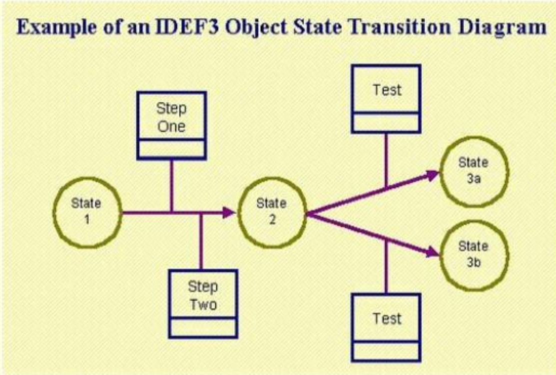


Figure 2.8. Example of Object state Transition

3. Unified Modeling Language (UML)

   Unified Modeling Language (UML) is described as a general-purpose, developmental, modeling language in the field of software engineering, that is intended to provide a standard way to visualize the design of a system. UML was originally motivated by the desire to standardize the disparate notational systems and approaches to software design and has its roots in the object-oriented programming methods.

   The main benefit of UML is that is not assumed any specific methodology for analyzing and designing when UML is used to express the results. In addition, a UML model can be transferred from one tool into a repository, or into another tool for refinement or the next step in your chosen development process. UML can be used for business modeling and modeling of other non-software systems. Business process modeling with UML can be considered as an extension of the UML- based modeling discipline, related to system modeling using the same notation. The types of the diagrams supported by UML are divided into three categories as follows:

   a) Structure diagrams

      Structure diagrams emphasize on the things that must be present in the system being modeled. Since structure diagrams represent the structure, they are used extensively in documenting the software architecture of software systems. For example, the component diagram describes how a software system is split up into components and shows the dependencies among these components.

   b) Behavior diagrams

      Behavior diagrams emphasize on what must happen in the system being modeled. Since behavior diagrams illustrate the behavior of a system, they are used extensively to describe the functionality of software systems. As an example, the activity diagram describes the business and operational step-by-step activities of the components in a system.

c) Interaction diagrams
   Interaction diagrams is a subset of behavior diagrams which emphasize on the flow of control and data among the things in the system being modeled. For example, the sequence diagram shows how objects communicate with each other regarding a sequence of messages.

The two mainstream diagrams in business process modeling are two behavior diagrams, the use-case and the activity diagram. The first one extends the software system use case concept to model the business system while the second one is focused on business processes. The use-case diagram is used to define the behavior of a system or other semantic entity without revealing the entity's internal structure. Figure 2.9 shows an example of a use-case diagram.

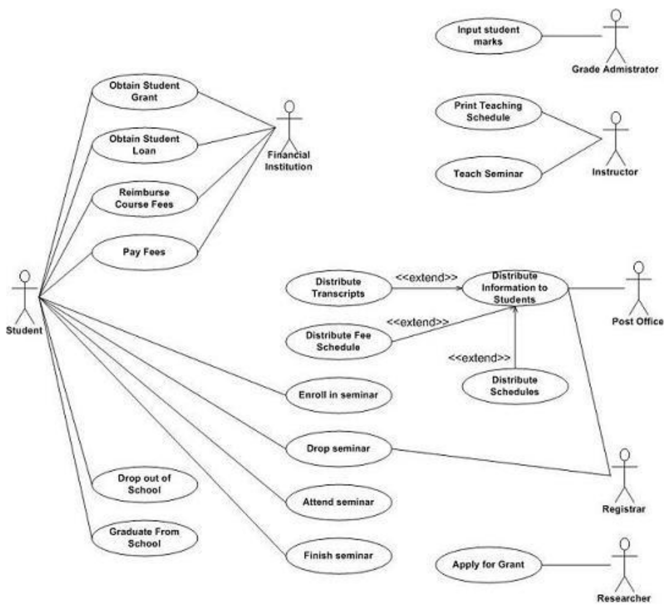


Figure 2.9. Example of a use-case diagram

An activity diagram is the graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. It is intended to model both computational and organizational processes along with the data flows intersecting with the related activities. In addition, it is typically used for business process modeling to visualize the logic captured by a single use-case or a usage scenario, or the detailed logic of a business rule. Consequently, it is considered as the object-oriented equivalent of flowcharts. Figure 2.10 shows an example of an activity diagram.

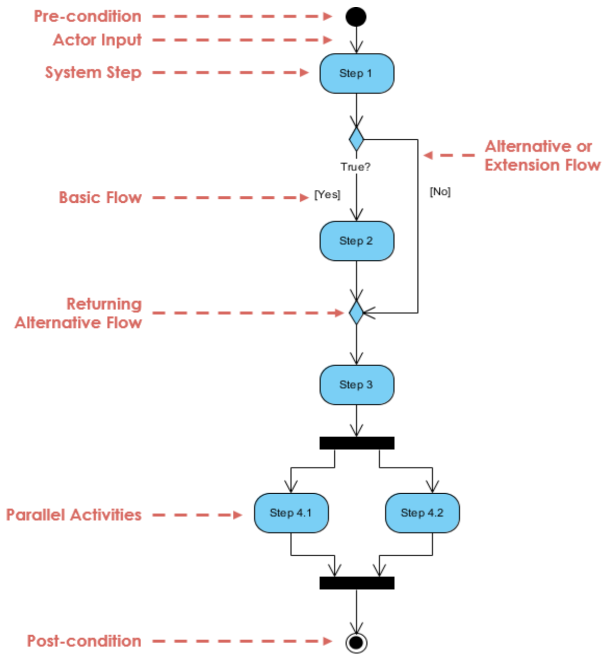


Figure 2.10. example of an activity diagram

4. Business Process Model and Notation (BPMN)

Business Process Model and Notation (BPMN) is a standard for business process modeling that provides a graphical notation for specifying business processes in a Business Process Diagram (BPD), based on a flowcharting technique very similar to activity diagrams from Unified Modeling Language (UML). The difference between BPMN and UML is that UML is object-oriented where BPMN takes a process-oriented approach which is more suitable within a business process domain. BPDs are commonly used to represent, analyze and implement the current (AS-IS) and improved (TO-BE) processes. The objective of BPMN is to support business process management, for both technical users and business users, by providing a notation that is intuitive to business users, yet able to represent complex process semantics. Its purpose is to model ways to improve efficiency, account for new circumstances or gain competitive advantage. The BPMN specification also provides a mapping between the graphics of the notation and the underlying constructs of execution languages, particularly Business Process Execution Language (BPEL).

BPMN supports modeling concepts only applicable to business processes. Other types of modeling for non-process purposes such as organizational structures, functional breakdowns or data models, are out of scope for BPMN. In addition, BPMN is not a data flow diagram, even though shows the flow of data (messages), and the association of data artifacts to activities. BPMN defines a set of graphical objects, and rules indicating the available connections between these objects. The four basic element categories of BPMN are presented in Figure 2.11.

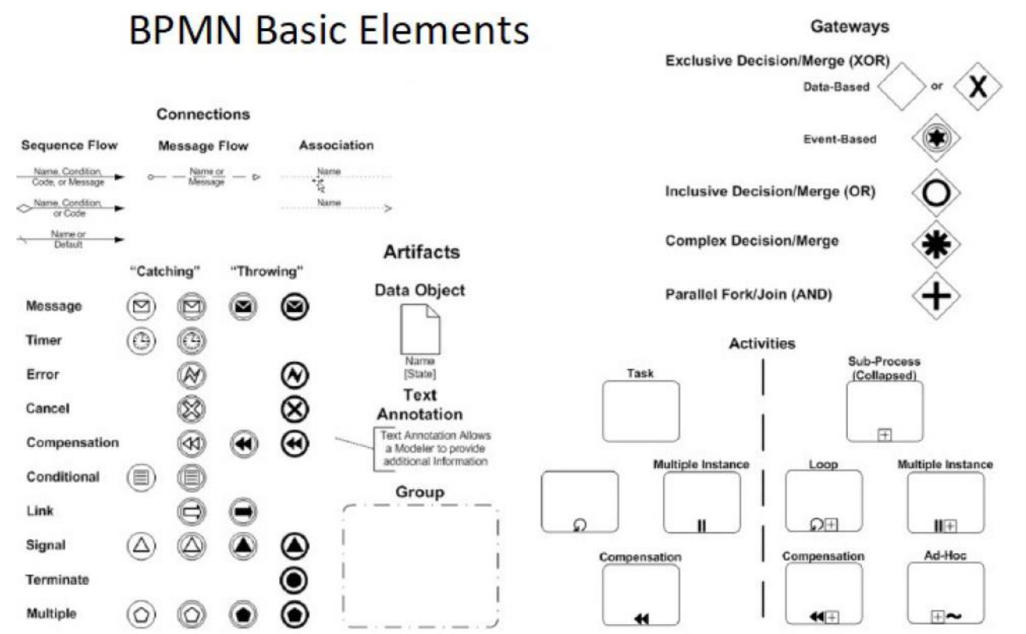


Figure 2.11. BPMN basic elements

A Business Process Diagram (BPD) depicts a detailed sequence of business activities and information flows needed to complete this process. It consists of the start events, the processes to be performed within the process to be modeled and the outcomes of the process. Decisions and branching of flows are modeled by gateways. A gateway is like a decision symbol in the flowchart technique. A process can also contain sub-processes which can be modeled in another BPD connected via a hyperlink to a process. If a process cannot be decomposed, it is considered a task, the lowest-level process. A "+" mark in a process denotes its capability for decomposing. An example of a BPD is shown in Figure 2.12.

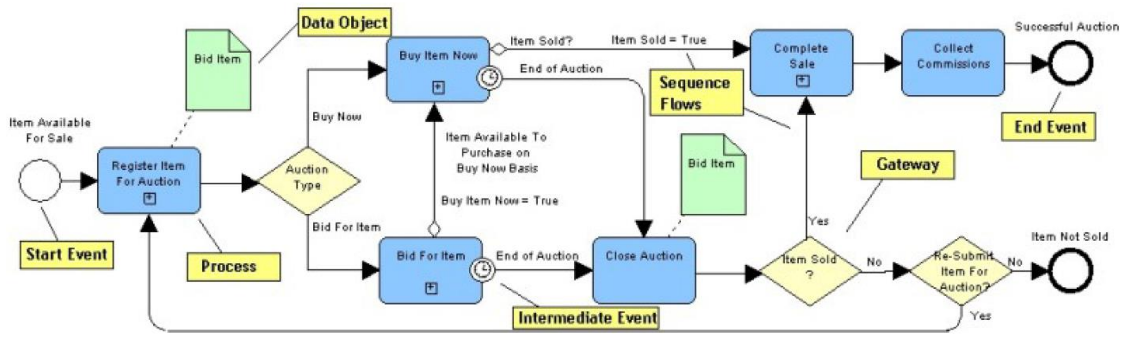


Figure 2.12. Example of BPD

Finally, you can drive further into business analysis by specifying 'who does what' by placing the events and processes into shaded areas called pools that denote who is performing a process. You can further partition a pool into lanes. A pool typically represents an organization and a lane typically represents a department within that organization (although you can make them represent other things such as functions, applications, and systems). An example of BPD containing pools is presented in figure 2.13.

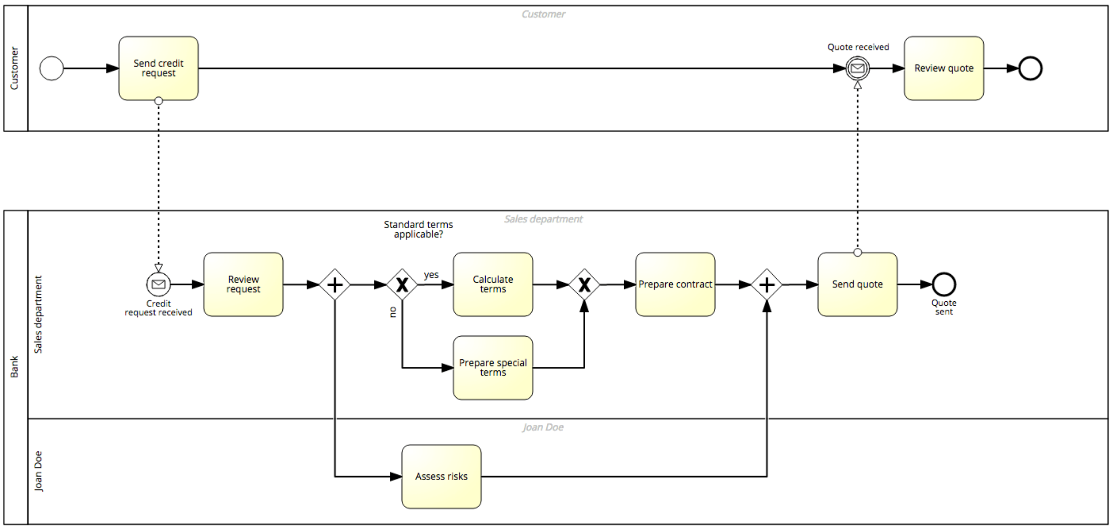


Figure 2.13. Example with pools

Mathematical/Formal Models

The main drawback of the business process modeling techniques described in the previous section is than none of them provides quantitative information to be used for analysis purposes. Zakarian (2001) points out that the process modeling techniques will be more attractive if formal techniques for analysis of process models are also provided. Formal models can define the process concepts rigorously and precisely so that mathematics can be used to analyze, extract knowledge from and reason about them. Koubarakis and Plexoudakis (2002) highlight the capability of formal models to be verified mathematically, as of high importance because this means that they can be proved as being self-consistent and have or lack certain properties. Van der Aalst et al. (2003) suggest that a formal foundation should be an integral part of business process models since it does not leave any room for ambiguity and the potential for analysis increases. Business process modeling lacks formal methods to support the business process model (BPM), according to Hofacker and Vetschera (2001). The reason is the qualitative nature of the business process elements and constraints; hence it is hard to parameterize them in a mathematical way, suitable for analytical methods, Tiwari (2001). There are few approaches that use mathematical models but there is not a common model to follow. Hence, every author found in literature, formulates the mathematical model of a business process according to the scope of his research. Furthermore, Hofacker and Vetschera (2001) in an effort to provide analytical support for business process optimization (BPO), note that the description-oriented models such as the diagrammatic modeling techniques discussed in the previous section, assume that the sequence of the activities involved in a business process is taken for granted while formal techniques have the structure of a process model to be determined by the problem specification. This constraint according to the authors, is weaker than the precedence constraint usually considered in scheduling problems, since the same resources can be generated by different activities. Next, two business process modeling approaches are presented using a formal model. These are the most representative and comprehensive approaches found in literature so far.

1. Modeling approach by Hofacker and Vetschera

   The first step towards analytical methods for business process modeling is owed to Hofacker and Vetschera (2001). They developed a general framework to represent administrative processes by setting mathematical constraints and a set of objective functions. These mathematical constraints define the feasibility boundaries of a business process. As objective functions, they use an additive function to be minimized and the maximization of the minimum value found in all activities, but any other objective function can also be used. The additive function simulates a cost function which sums the costs across the activities in a business process. Activities along with resources, are the main elements in a process model. Resources are divided into physical and information objects that flow through the system while activities demonstrate the transformation steps which use input resources and produce new ones as output resources. Each activity is represented by a node and uses one or more input resources and generates one or more output resources. Both input and output resources are represented by arcs connecting an activity to other ones. A business process has its own input and output resources called as global inputs and global outputs respectively. Additionally, a set of attributes is assigned to each activity for evaluation purposes for the entire process by aggregating the evaluations of the activities contained in the process, e.g. cost, duration or quality aspects. The sequence of activities is to be determined and the potential sequences are constrained by the requirement that resources must be produced by some activity before they can be used by other activities. Hence, there may be different sets of activities that when properly connected lead to different process models for the same process. The only difference is the sequence that these activities are executed with. For this reason, they consider a set of potential activities with different characteristics, e.g. inputs, outputs, attributes, which the process must be constructed from and try to find the subsets of these potential activities that comply with the problem specification.

   The problem specification for the relationship between the activities and resources is based on the following three assumptions/constraints:

   a) Each activity consumes exactly one unit of its input resources and generates one unit of its output resources
   b) All input resources of an activity must be available before this can be executed
   c) All output resources of an activity are generated when this is executed

For the first constraint mentioned above, Hofacker and Vetschera (2001) point out that this can be extended to allow for arbitrary input and output coefficients. It is up to the designer's perception and the examined process itself. In addition, they characterize the second assumption as non-critical because alternative input resources of an activity can be modeled in the same framework by defining additional potential activities and provide the following example:

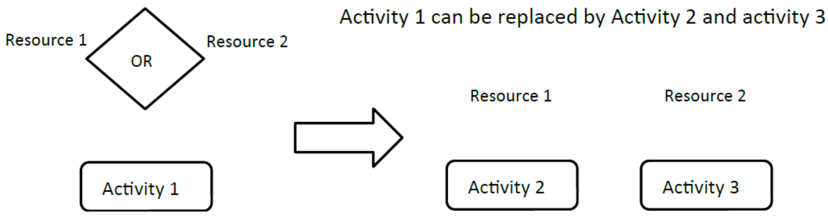


Figure 2.14. Example of a potential activity

The set of global inputs is available at the beginning of the process and the process must produce the global outputs. The assumptions taken according to the feasibility of a process design, are presented below:

a) For all activities contained in the process design, all their input resources are either contained in global inputs or are generated by other preceding activities. For physical activities, no other activity must consume the same unit of the resource.
b) All global outputs are generated by some activity contained in the process design and again, physical resources must not be consumed by other activities.

An example of a feasible business process design is shown below in figure 2.15.

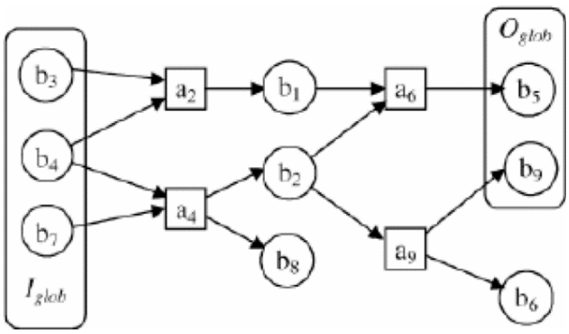


Figure 2.15. Example of feasible business process design

This attempt established the connection between a process design and a mathematical model and enabled the business process optimization (BPO). However, it is too generic to capture the aspects of the real-world processes. The outputs of real-world activities highly depend on their execution, e.g. a manufacturing activity can produce good or bad parts. This XOR junction must be incorporated somehow in the mathematical model. For real-world resources, there is no assumption that a resource is generated but not used in the subsequent process because of its cost for the company. E.g. a bad part in manufacture will not be thrown away but will take the way of rebuilding. By dividing resources into disposable and non-disposable ones, the authors managed to overcome this issue. If there are non-disposable resources not consumed in a process model, the whole model is infeasible. Finally, they recognize that real-world objectives may depend on the joint presence of activities in a model, hence you cannot simply evaluate them individually. Their proposal for this issue, has to do with assigning a presence indicator value to each activity for synergy identification purposes. Then, the corresponding coefficients to those synergies can be considered in the process evaluation.

2. Modeling Approach by Vergidis
   An innovative formal specification and representation technique was proposed by Vergidis (2008) to enable the application of state-of-the-art evolutionary multi-objective optimization algorithms to business process optimization (BPO). This technique aimed to support a visual diagrammatic representation of processes and have a formal/mathematical underpinning so that quantitative measures can be extracted. The new in his research, was the development of the Process Composition Algorithm (PCA) to compose algorithmically business processes based on specific requirements and fill the gap between the visual and the quantitative perspective of business processes.
   Due to the fact that the business process representation is a core element for the business process optimization as it was proposed by Vergidis (2008), it consists a main subject of this dissertation and it will be discussed in-depth in the next chapter.

2.3 Business process optimization (BPO)

Business process optimization (BPO) is considered as the problem of constructing feasible business process designs with optimum attribute values such as duration and cost Georgoulakos et al. (2017). The business process modeling techniques described in the previous section, are strongly motivated by the need for business process improvement. According to Smith (2003), large organizations need to map their processes for two main reasons: One is to have a clear picture of the current situation and the flow of activities within the organization and second is to improve those processes efficiently to meet the organizational goals. Similarly, Grigori et al. (2004) acknowledge that organizations need to provide their processes with a high, consistent and predictable quality. They also identify as prerequisites for BPO that business processes should be correctly designed, their execution should be supported by a system that can meet the workload requirements and the process resources, e.g. human, material and non-material, should be able to perform their work items in a timely fashion. Therefore, an approach for BPO should clearly define and specify how optimization is perceived and which aspect of the process is going to be optimized. In this section, we are going to present some of the optimization approaches found in literature.

1. Optimization Approach by Hofacker and Vetschera

   Hofacker and Vetschera (2001) attempt to optimize the design of (mainly administrative) business processes. They introduce formal models for the business process design problem which can be used to analytically determine optimal designs with respect to various objective functions subject to several constraints. It is perceived to be the most comprehensive work towards BPO because three different optimization techniques have been examined along with the process formal model: mathematical programming, a branch and bound method and genetic algorithms.

   Mathematical Programming Formulation

   Their first attempt consists of the formulation of the process design to a mathematical problem. They use an additive function and several constraints to describe the problem and cover all its aspects. The objective function is minimized or maximized according to the optimization goal and the constraints describe and ensure the feasibility of the process in a mathematical formal way. As mentioned in the previous section, the main elements used in the process design are the activities and the resources. The mathematical constraints can be grouped into two major categories:

   1. constraints related to input and output resources of each activity and
   2. constraints regarding the time sequence of resources and activities.

   Every process has a set of process input resources available and must produce the set of process output resources. The participating activities must be sequenced in such way that they use some resources as inputs and then produce resources that can be used as inputs by other activities until the set of process output resources is generated. The constraints of the first group ensure that input resources are available by activities to use and the set of process output resources is eventually produced.

   In order to set formally the constraints, they introduce several variables and arrays that bind together the activities and the resources. That increases the complexity of the process model but also ensures its strict mathematical formality. In addition, it makes the model more flexible as a constraint can be eliminated to simplify a particular aspect of the model or extra constraints can be added to shape the model further. According to the experiments performed, the mathematical approach produced satisfying results but poor execution times.

Genetic Algorithms

Genetic algorithms have been successfully applied to complex problems in a variety of areas. Their advantage is that they maintain a population of possible solutions to reach feasibility and this makes them powerful. Another significant advantage is their extendibility to optimize a problem under more than one criterion. Multi-objectivity makes genetic algorithms a flexible methodology that can be applied to any optimization problem.

A genetic algorithm imitates the process of natural evolution to find an optimal solution. It works on many solutions in parallel, where each solution corresponds to an individual in the population. Each solution is represented by an appropriately coded string, its genome. A mutation operation changes the values of randomly chosen positions of that string. The resulting mutated individuals are then selected for mating. A crossover operation exchanges information between two individuals. Finally, the selection operation selects randomly the superior solutions to form the new generation. The selection probability depends on the objective function value, and the process continues until some pre-defined termination criteria are fulfilled.

The business process design described before, must be solved with respect to several constraints. The authors chose between two approaches to deal with the constraints within a GA framework. In the first approach, a penalty term for constraint violation is added to the original objective function. The second approach modifies the genetic operators to limit the search space to feasible solutions. This approach is appropriate if feasible changes can be easily determined. They decided to follow the first approach as the second would require extensive computational effort.

The initial tests showed weak performance for genetic algorithms. The main issue was that the genetic algorithms could not maintain the feasibility of a design alternative in a tightly constrained problem as the business process optimization problem. The design of a process requires activities to be ordered so that all inputs of an activity are generated by preceding activities. The feasibility cannot be maintained by the operations of the genetic algorithms and therefore it is incorporated via the penalty terms in the fitness function. For this reason, the authors suggested that in a highly constrained problem an algorithm which maintains feasibility must search a much smaller space, leading to better performance.

2. Optimization Approach by Vergidis

Vergidis (2008) proposed a business process optimization framework (BPO^F) to capture, visualize and express a business process design in a quantitative way that allows Evolutionary Multi-Objective Optimization Algorithms (EMOAs) to generate a series of alternative optimized designs. The next chapter is dedicated to the business process optimization framework (BPO^F) as it is the issue that this dissertation is mainly concentrated on.

2.4 Summary

This chapter examined the basic aspects regarding business process definition, modeling and optimization. Moreover, it provided an overview of the most common definitions for business processes, the main techniques for business process modeling and the most representative approaches for evolutionary multi-objective business process optimization. The next chapter presents in-depth the Business Process Optimization framework (BPO^F) that was proposed by Vergidis (2008) and is considered to be the core of this research

Chapter 3 – Business Process Optimization Framework

This chapter introduces the business process optimization framework (bpo^F) that was proposed by Vergidis (2008). The bpo^F is considered to be the core of this research. The chapter starts by presenting in detail the proposed business process representation which is a main component of the framework. Then an analytical description of the framework follows in which the main operation and the main steps of the optimization are presented. Moreover, the framework implementation is provided. The chapter concludes with the review of some experiments that were done by Vergidis (2008) in order to evaluate the performance of the proposed framework and the validation of these results.

3.1Business Process Representation

This section introduces the representation technique that was proposed by Vergidis (2008) on Business Process Optimization Framework (bpo^F). Vergidis (2008) focused on business processes found in the service industry, hence a business process itself is considered as a service and its outcomes are non-material equivalents of goods based on the service definition. The proposed representation included all the value-adding business processes operations performed within an organization. It captured business processes regarding the functionalities that are involved instead of the steps that should be executed and emphasized accordingly on the flow and connectivity of the participating functionalities rather than on execution details. In addition, it allows the hierarchical structuring of business processes like diagrammatic modeling techniques do. This comes from the perspective of identifying the main functionalities within a business process since it implies that a strategic process and an operational process can be similarly perceived. Therefore, a functionality identified in a higher level can itself be a business process at a lower level.

Visual representation of a business process design

The visual representation proposed by Vergidis (2008) is made via a simple flowchart and the main elements of a process are as follows:

- The participating tasks
- The resources of a task / business process
- The attributes of a task / business process
- The connectivity patterns

The tasks represent specific functionalities intended to perform core operations. The difference among tasks is found in the core operation they perform. Being properly connected and sequenced, the tasks perform the business operation of a higher-level business process. Resources are related to the input and output products of the tasks and the business process. They are transformed while they are flowing through the tasks of a process to produce the process output resources. They also control the way tasks are connected within a business process and help in shaping the connectivity patterns occurring in the process design. Every task has also attributes which represent their measurable characteristics to be used for the evaluation of the business process design. The connectivity patterns are essential in order to express recurring paths in a process and responsible for shaping the process design. The proposed notation for a flowchart depicting a business process design is:

- Two rounded boxes marked as 'START' and 'END' appear in every design and denote the beginning and the end of the process.
- The participating tasks are sketched as boxes.
- The resources are the connecting arrows that link the tasks
- The patterns are depicted as follows:
    - Sequence is sketched as the connecting arrow between two tasks
    - Parallel flow (AND) is sketched as box
    - Multi-choice (OR) is sketched as rhombus
    - Arbitrary loops (GOTO) are sketched as arrows pointing backwards

Figure 3.1 shows an example flowchart for a generic business process design based on the proposed notation.

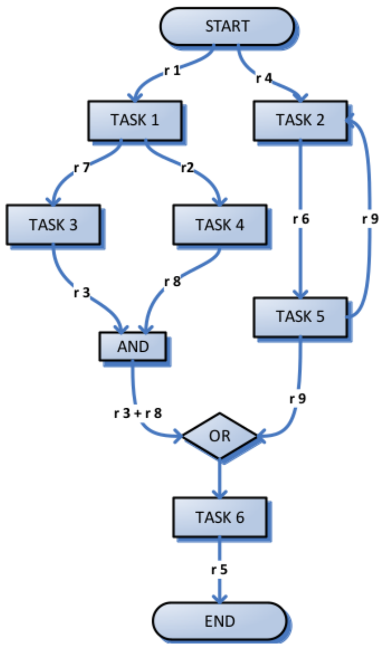


Figure 3.1. Example of the visual representation of a generic business process design

Mathematical parameters of the business process elements

Vergidis (2008) expressed the main elements of a process using mathematical parameters in order to represent a business process design in a quantitative way. Table 3.1 shows the encoding of the main parameters.

Parameter	Description	Parameter	Description
n_d	Number of tasks in the design	N_d	Set of the n d tasks
r_d	No. of resources in the design	R_d	Set of the r d resources
t_{in}	No. of task input resources	I_i	Set of the t in resources for a task i
t_{out}	No. of task output resources	O_i	Set of the t out resources for a task i
r_{in}	No. of process input resources	R_{in}	Set of the r in resources
r_{out}	No. of process output resources	R_{out}	Set of the r out resources
p	No. of task/process attributes	TA_i	Set of the task attribute values for a task i
		PA	Set of the p process attribute values

Table 3.1. Main process parameters

As it seems in Table 3.1 the parameters are related to each other. Particularly the set of n_d tasks that belong to a particular process design is $N_d = \{t_1, t_2, t_3, ..., t_{nd}\}$. The set of r_d resources in the design $R_d = \{r_1, r_2, r_3, ..., r_{rd}\}$ accommodates the subsets R_{in} and R_{out} that store the process input resources and process output resources respectively. The business process design utilizes all the resources in R_{in} and produces all the resources in R_{out}. Also, each task *i* in the design has t_{in} input resources stored in $I_i \subseteq R_d$ and t_{out} output resources stored in $O_i \subseteq R_d$. Finally, each task *i* has *p* attribute values stored in the TA_i set and the corresponding *p* process attributes are stored in the PA set. Figure 3.2 shows the business process design in figure 3.1 related to the parameters of table 3.1.

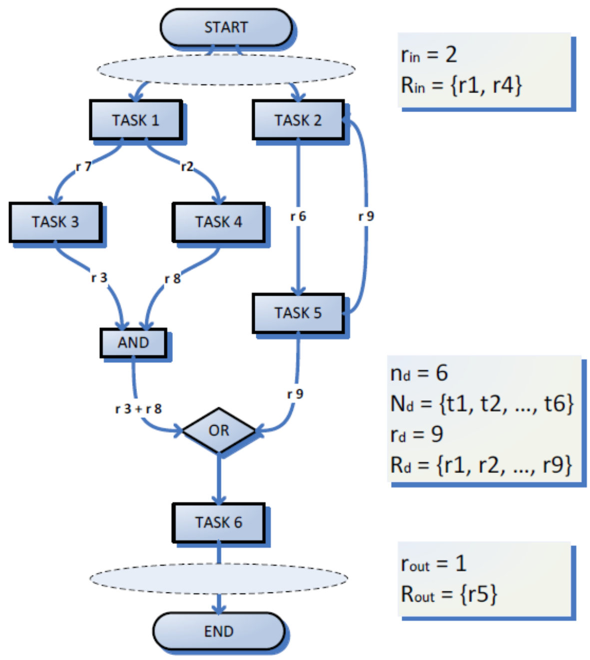


Figure 3.2. Parameters and visual representation of a process design

Based on these parameters Vergidis (2008) elaborated two matrices. The first matrix called *Task Attributes Matrix* (TAM) aims to gather the attribute values of the tasks so as the calculation of the process attributes and the evaluation of the design become easier. The way that the TAM calculates the process attributes is described later in this chapter. Table 3.2 provides an example of TAM for the generic design in Figure 3.1 assuming two attributes (A_1 and A_2).

Attributes / Tasks	A_1	A_2
Task 1	100	300
Task 2	120	302
Task 3	117	324
Task 4	178	308
Task 5	145	356
Task 6	157	389
PROCESS	817	1979

Table 3.2. Example of Task Attributes Matrix (TAM)

The second matrix called *Task Resources Matrix* (TRM) aims at capturing the task sequencing and the patterns formulated in the process design. To achieve this, the matrix maps the input and the output resources of the tasks in the process design. Thus, each cell in TRM represents the relationship between the task and the resource. The rules are as it follows:

- If $TRM_{ij} = 1$, the resource belongs to the set of input resources of the task
- If $TRM_{ij} = 2$, the resource belongs to the set of output resources of the task
- If $TRM_{ij} = 0$, the resource belongs neither to the set of input resources nor to the set of output resources of the task

Table 3.3 provides an example of TRM for the generic design in Figure 3.1. Figure 3.3 shows the TRM mapping 'Task 1' of this design.

Tasks \ Resources	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9
Task 1	1	2	0	0	0	0	2	0	0
Task 2	0	0	0	1	0	2	0	0	1
Task 3	0	0	2	0	0	0	1	0	0
Task 4	0	1	0	0	0	0	0	2	0
Task 5	0	0	0	0	0	1	0	0	2
Task 6	0	0	1	0	2	0	0	1	1

Table 3.3. Example of Task Resources Matrix (TRM)

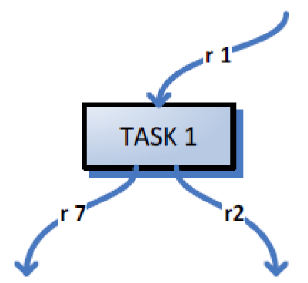
Visual representation	TRM mapping
	**$TRM_{11} = 1$** **$TRM_{12} = 2$** $TRM_{13} = 0$ $TRM_{14} = 0$ $TRM_{15} = 0$ $TRM_{16} = 0$ **$TRM_{17} = 2$** $TRM_{18} = 0$ $TRM_{19} = 0$

Figure 3.3. Example of TRM mapping based on 'Task1'

TRM provides the basis for reproducing the business process design based on the process requirements. However, the transformation of the quantitative representation of a business process into a visual diagram, is not trivial because the quantitative perspective cannot ensure the feasibility of the business process design based only on the TRM and the process input and output resources. For this reason Vergidis (2008) proposed an algorithm that can construct the business process diagram, given its quantitative representation, and check whether the result corresponds to a feasible business process or not. This algorithm is called Process Composition Algorithm (PCA) and will be presented below.

Process Composition Algorithm (PCA)

The Process Composition Algorithm (PCA) provides the bridge between the visual and the quantitative perspective. It composes the visual diagram of a process, stored in TRM, in a way that the design captured by both representation perspectives is feasible. According to Vergidis (2008) business process representation a business process design is considered as infeasible when:

1. One or more process input resources cannot be utilized from the tasks in the TRM
2. One or more process output resources cannot be produced from the tasks in the TRM
3. There is no task in TRM than can be attached to the process diagram based on its input and output resources

These cases of infeasibility result in a high probability of infeasible solutions found by PCA and this happens even for a large size of the task library. The first two cases where one process input cannot be utilized, or one process output cannot be produced, are inevitable. The third case of infeasibility may happen even in the case that all process inputs are utilized, and all process outputs are produced and is the most frequent. PCA attempts to tackle the infeasibility issues and construct a feasible process diagram.

1. Process input and output requirements (R_{in} and R_{out})

2. Participating tasks in the design (TRM)

3. Task library (N)

Figure 3.4. PCA requirements

Figure 3.4 shows the requirements of PCA. The process requirements in the form of the process input and output resources are required as the termination criteria. The algorithm attaches tasks to the process design until the process inputs are utilized and the process outputs are produced. The second requirement is TRM that contains the tasks that form the design for an individual solution. PCA will add tasks to the design from TRM and check whether they form a feasible solution. Finally, the task library is essential to modify or repair the design. The task library can be considered as a repository of tasks that can potentially participate in a business process design. Because of the high probability of infeasibility during the composition of a process design, PCA uses the task library to repair the design in order to make it feasible or to improve a feasible process design by replacing tasks with better attribute values.

1. Business process design (process graph)

2. Updated set of tasks in the design (N_d)

3. Degree of Infeasibility (DoI)

Figure 3.5. PCA outcomes

The outputs of PCA are shown in figure 3.5. The main output of the algorithm is the business process design which is composed and represented as a directed graph. The second outcome of PCA is the updated set of tasks that participate in the process design based on the execution of the algorithm. TRM represents the individual solution and PCA translates it into a process design. During the execution of PCA, TRM is updated for the following reasons:

1. The elimination of tasks in TRM that cannot be attached to the process diagram during its composition and
2. The replacement of tasks in TRM with tasks in the library that make the composed design feasible.

Last but not least, Degree of Infeasibility (DoI) is the third output of PCA. DoI was introduced by Vergidis (2008) as a metrics for process design infeasibility. Measuring the infeasibility of a design is of high importance in order to the different process designs to be compared and evaluated. DoI is based on three main factors and is calculated as:

$$DoI = 1 \cdot n_{in} + 5 \cdot (\sim r_{out}) + 3 \cdot (\sim r_{in}) \qquad \text{Equation 3.1}$$

For each infeasibility case, DoI assigns a different weight that reflects its relative importance and frequency. For every task inserted from the library of tasks in the process design, DoI is increased by 1 (n_{in} = total number of tasks inserted from the library). This infeasibility case is considered as a frequently occurring one during the design composition, hence its weight. For every process output resource not produced, DoI is increased by 5 (r_{out} = total number of process output resources not produced). Vergidis (2008) considers this case as the most important one for the feasibility of a process design. The production of all process outputs serves as the termination criterion of PCA for a feasible process design. Finally, for every process input resource not utilized, DoI is increased by 3 (r_{in} = total number of process input resources not utilized). This case is as important as the previous one although the weight here is less than the output resources. Vergidis (2008) deems that the production of all output resources means that at some point all process input resources are utilized. For one or more input resources to be missing it means that the corresponding tasks were omitted during the last stage of PCA and thus the penalty is less. As each individual solution – process design carries a DoI, the feasibility comparison among the designs generated by PCA, is straight-forward. A feasible process design has zero DoI. The main steps of PCA are presented below.

Main steps of PCA

Figure 3.6 displays the main steps of the Process Composition Algorithm (PCA). PCA constructs a process graph and traverses it to ensure that it meets the process requirements. In the graph, each task is represented as a node and there are two artificial nodes, the 'START' node with the process input resources and the 'END' node with the process output resources. These nodes facilitate the connection of the process input and output resources with the participating tasks in order to produce a process design that meets the process requirements. The graph is elaborated with the breadth–first strategy using the concepts of 'parent' and 'child' levels. The 'parent' level consists of the nodes already inserted in the graph and the 'child' level is the one where the new tasks are added in the design based on the output resources of the tasks in the 'parent' level. Once the elaboration of all tasks in the 'child' level is completed, it becomes 'parent' level for the graph elaboration to proceed.

PCA starts by inserting the artificial nodes 'START' and 'END' to an empty graph. The 'START' node is initially marked as the 'parent level'. Then, the algorithm visits all the nodes in parent level one by one in order to elaborate the child level. Once the child level elaboration is completed, the output resources of the recently attached tasks along with the unlinked output resources of previous tasks are checked to find out whether they contain the process output resources. In the case that not all the output resources are produced and there are unused tasks in TRM, the tasks in 'child' level become the new 'parent' level and the elaboration process is repeated. If there are no unused tasks in TRM then for every output resource that has not been produced there is a penalty attached to the design and DoI is updated accordingly.

In the case that –at some stage of the elaboration process– all the process output resources are produced, TRM and the graph are updated. The update process involves two parts: (i) the elimination from TRM of any tasks that have not been inserted in the process design, and (ii) the elimination of graph nodes (tasks) that do not contribute to the production of the process outputs. After the update, PCA checks whether all the process input resources are produced. Some of the tasks that were utilizing the process inputs might not have contributed to the process outputs and therefore are removed from the design. In the case that one or more process inputs are not utilized, there is a penalty attached to the design and DoI is updated accordingly. In the case that all the process inputs are produced, the design is marked as feasible.

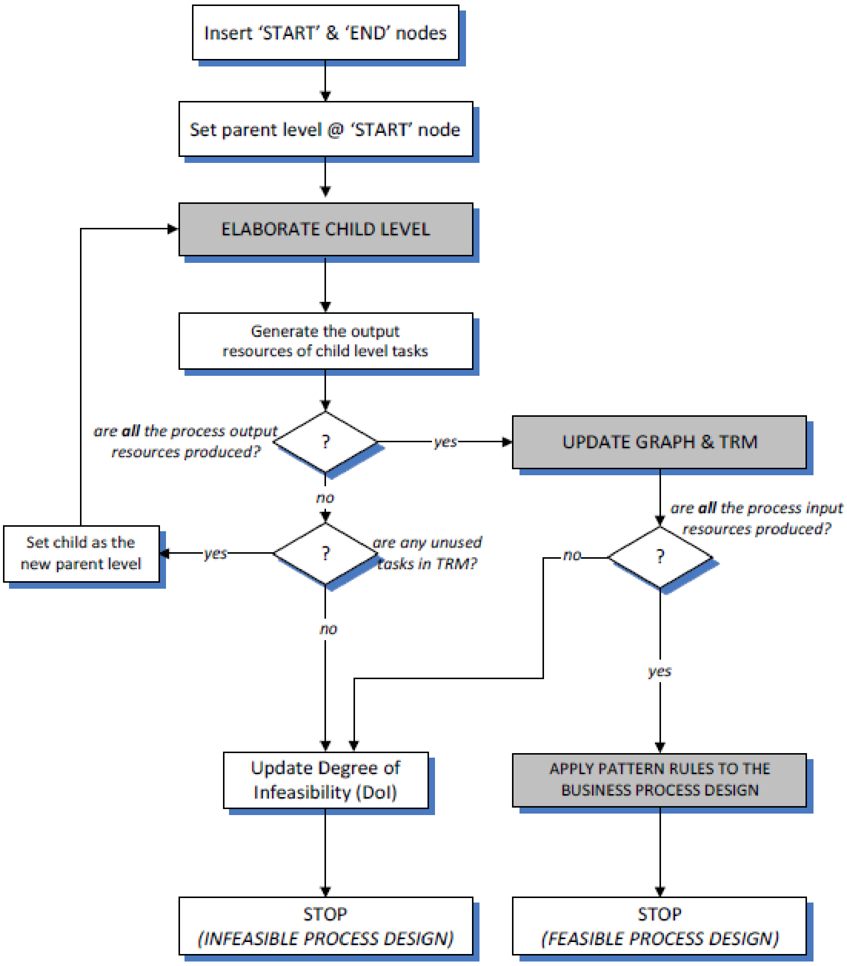


Figure 3.6. Main steps of the PCA

3.2 Problem formulation

This section introduces the formulation of the business process optimization problem as it was proposed by Vergidis (2008) on Business Process Optimization Framework (bpo^F). The problem formulation is based on the business process representation which was described in the previous section. Table 3.4 shows the parameters of the business process optimization problem.

Parameter	Description	Parameter	Description
n	Number of tasks in the library	N	Set of the n tasks
n_d	No. of tasks in the design	N_d	Set of the n_d tasks (subset of N)
n_{min}	Minimum number of tasks in the design	N_{in}	Set of library tasks to be included in the process design (subset of N)

r	No. of available resources	N_{ex}	Set of library tasks to be excluded for the process design (subset of N)
t_{in}	No. of task input resources	S_d	Set of the different process sizes
t_{out}	No. of task output resources	DoI	Degree of Infeasibility (as calculated by the PCA algorithm)
r_{in}	No. of process input resources	TAM	Matrix that stores the task attribute values for each of the n_d tasks in the process design
r_{out}	No. of process output resources	PA	Set of the p process attribute values
p	No. of task/process attributes		

Table 3.4. Parameters for business process design optimization problem

The aim of this problem is the generation of alternative optimized business process designs. For this reason, Vergidis (2008) introduced the library of available tasks. As it was mentioned in previous section, the library of tasks can be considered as a repository of tasks that can potentially participate in a business process design. This fact affected the mathematical parameters as new parameters were added. The most significant additions for the problem are the number of tasks in the library (n) and the number of resources of the tasks in the library (r). We will engage with them later in this dissertation.

The multi-objective problem formulation for business process optimization is as follows:

For a business process design with a set of n_d tasks and p process attributes:

- Minimize / maximize $(PA_1, PA_2, \ldots, PA_p)^T$
  Considering the following constraints:

Compulsory constraints	**Optional constraints**
1. $DoI = 0$	a) $n_d \geq n_{min} > 0$
2. $n \geq n_d > 0$	b) $n_d \in S_d$
3. $r \geq r_{in}, r_{out}, t_{in}, t_{out} > 0$	c) $N_d \cap N_{ex} = \emptyset$
4. $p \geq 2$	d) $N_{in} \subseteq N_d$

Table 3.5. Constraints of problem formulation

We assume that the process attributes are used as the optimization objectives. A process attribute (PA_j) can be calculated as an aggregate of the corresponding task attributes stored in TAM for all the n_d tasks in the process design according to the Equation 3.2 .

$$PA_j = \sum_{i=1}^{n_d} TAM_{ij} \qquad \text{Equation 3.2}$$

To clarify the content of the Table 3.5 we can say that the compulsory constraints are used to ensure that:

1. only feasible business process designs are evaluated. The only case that a design is feasible is when DoI equals to zero.
2. the available tasks in the library (n) are more than or at least equal to the tasks required to compose a design (n_d) and that both (n, n_d) are greater than zero.
3. all the resource-related parameters are greater than zero and the available resources (r) are more than those required by the process and task inputs and outputs.
4. there are at least two task/process attributes and thus the problem is multi-objective or at least bi-objective.

In the terms of the optional constraints of the Table 3.5, they are provided in order to make the problem more flexible in terms of business process designs generated. In particular optional constraints are used to:

a) set a lower limit (n_{min}) to the number of tasks that can formulate a design. In the case that $n_d = n_{min}$, an acceptable solution contains exactly n_d tasks in the design.
b) receive a design as acceptable only if its size belongs to a specific range of process sizes (S_d).
c) ensure that the solution does not contain any undesired tasks from the library.
d) enforce particular tasks to be included in the solution.

The problem formulation described above defines how an optimized business process design should be. In the next section we will describe how the framework generates those optimal results.

3.3 Framework (bpo^F) description

In this section we present a description of Business Process Optimization Framework (bpo^F). It can be considered that the framework consists of two main components which are the business process representation technique and a series of Evolutionary Multi-objective Optimization Algorithms (EMOAs). The framework utilizes these two components to generate alternative optimized designs. However the question that arises is the way it operates.

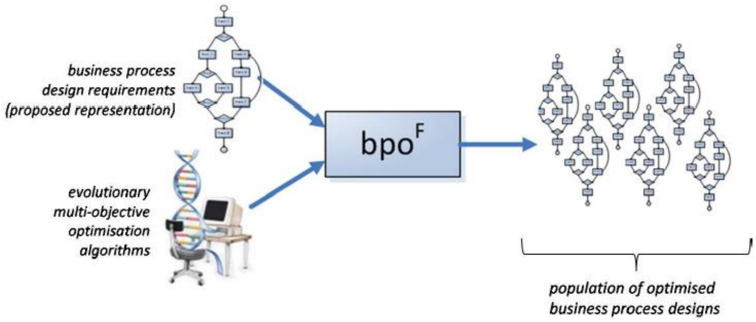


Figure 3.7. The business process optimization framework (bpo^F)

The framework needs four inputs to operate. In particular, the inputs are:

1. The process requirements for the design in the form of the required process inputs (R_{in}) and process outputs (R_{out}). All the generated designs must start from the same inputs and conclude to the same outputs.
2. The process size (n_d). The process size denotes the maximum number of tasks in the process designs.
3. The library of tasks (N). This set contains all the tasks that can potentially participate in a process design.
4. The process attribute functions. These functions are the formulas for each of the process attributes. The optimization framework uses these functions as optimization objectives.

Afterwards the framework employs a series of Evolutionary Multi-objective Optimization Algorithms (EMOAs) in order to optimize business process designs. The selected EMOAs are: NSGA2, SPEA2, PESA2 and PAES. In the context of this dissertation we are going to utilize and refer only to NSGA2. NSGA2 is considered to be a high-performing multi-objective optimization algorithm that has as its main parameters population size, number of generations along with crossover and mutation probabilities. The procedure of optimization is as follows:

1. Generate random population
   During this step a fixed number of sets of n_d tasks is created. This number equals the population size that the algorithm is working with. Each set contains n_d randomly allocated tasks from the task library (N) but with a constraint that a task must appear only once in the same set. This step occurs only once in the optimization process.
2. Check constraints
   In this step, the constraints (discussed in 3.2 section) are checked for each solution. To define the Degree of Infeasibility (DoI), TRM is formed and PCA is executed. PCA exports the diagrammatic version of the business process design, the DoI and the

updated set of tasks N_d that reflects the actual tasks in the solution. If any optional constraint is included, it will be checked.

3. Evaluate solution
   The step of the solution evaluation includes two stages: (i) TAM is created based on updated version of the solution and (ii) the various process attributes are calculated based on their functions. The solution evaluation is done after the constraint checking so as only the tasks that participate in the process design are taken into account in the evaluation process.
4. Perform crossover
   During this step the solutions undergo crossover. Crossover is a genetic operator that exchanges information between solutions. For the business process optimization problem, crossover occurs directly in the N_d set of each solution. Figure 3.8 illustrates how it works. The solutions are selected for crossover based on a given crossover probability defined by the EMOA. For each pair of solutions a unique crossover-point is defined based on a random number (between 1 and n_d -1).

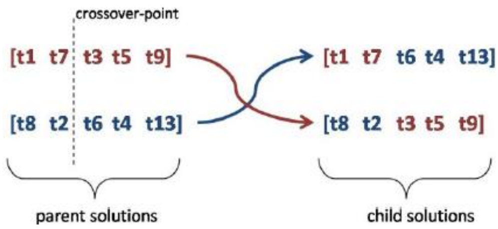


Figure 3.8. The 'process crossover' operator

5. Perform mutation
   The last step is mutation. Mutation is a genetic operator that randomly alters information in a chosen solution. As before the operator is applied on the N_d set of a solution and a chosen task is replaced by an arbitrary task from the library. The probability of mutation is again defined by the EMOA. Figure 3.9 illustrates an example.

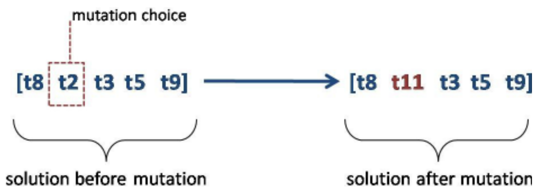


Figure 3.9. The 'process mutation'

In contrast to step 1 that occurs only once in the beginning of the procedure, steps 2-5 are repeated for a predefined number of generations. Figure 3.10 demonstrates the procedure of optimization that is described above.

Lastly when the optimization has finished, the outcome is the population of optimized business process design. Moreover, for each design, the framework produces:

1. The tasks in the design, stored in the N_d set.
2. The process graph, which is the diagrammatic representation of the design.
3. The Degree of Infeasibility (DoI), which for the optimized process designs should be equal to zero
4. The process attribute values, which are calculated based on the input functions.

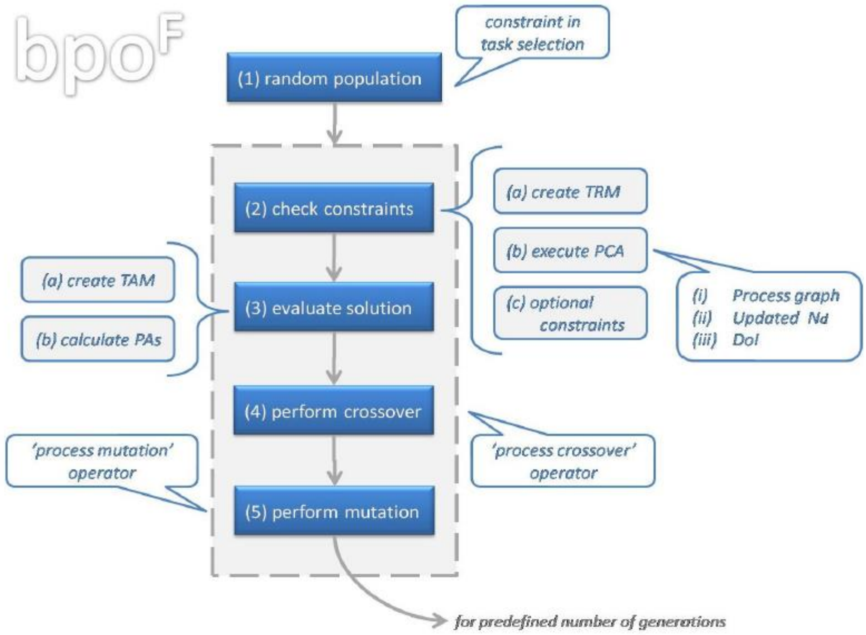


Figure 3.10. Main steps of bpo^F

3.4 Framework implementation

This section presents the implementation of the framework. The framework is programmed using the Java programming language and based on three Java libraries. In addition it is presented the procedure of setting up a business problem optimization test problem.

The framework was programmed as a combination of three Java libraries, two of them were open-source and available on-line and the third was developed by Vergidis (2008) for the purpose of the framework. Figure 3.11 shows the relationship between these three libraries based on their main packages.

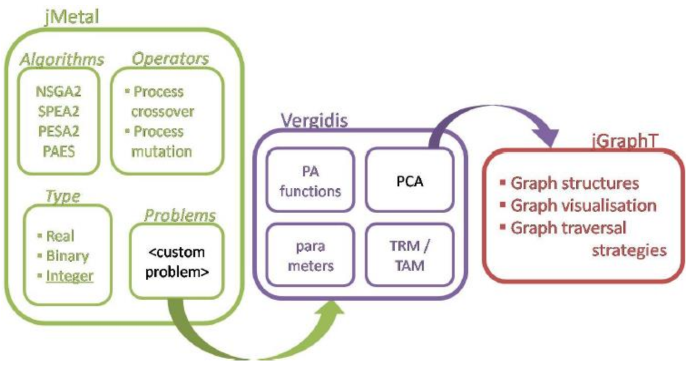


Figure 3.11. The three main Java libraries

As it seems in figure 3.11 the pattern begins from the jMetal library. This library implements a variety of EMOAs including those that are employed by bpo^F. The EMOAs are under the package 'Algorithms', while the genetic operators are developed under the package 'Operators'. Another package is 'Type' that implements the various types of problems to be solved (e.g. Real, Binary). In this package, the class for handling an Integer problem is programmed by Vergidis (2008) in order to handle the business process optimization problem. Finally, jMetal implements a variety of standard multi-objective optimization problems and allows for custom user-defined problems to be developed. It is in this package that a pointer towards the business process optimisation problem was developed. The pointer directs the execution to the Vergidis (2008) library where all the specific business process related components are programmed.

The second library of the framework is the Vergidis library. Vergidis (2008) developed this library exclusively for the business optimization problem. The library includes four main packages. The first package 'PA functions' defines the different functions for the process attributes that can be used as objective functions. The second package is called 'Parameters' and stores classes with various problem parameters (e.g. process size, library size, number of objectives, etc.) in order to test the framework for different problems. The package named 'TRM / TAM' includes the classes that convert a solution to the TRM / TAM representation. Last but not least is the 'PCA' package. This package implements the PCA algorithm and produces most of the framework outcomes.

The last library of the framework is the jGraphT library. This library provides a graph manipulation. Vergidis (2008) employed this library to construct in combination with PCA a business process design as a graph. Moreover, this library provides the framework with classes that visualize the graph. In the context of this dissertation we are not going to utilize this library as we do not need the diagrammatic representation for our study.

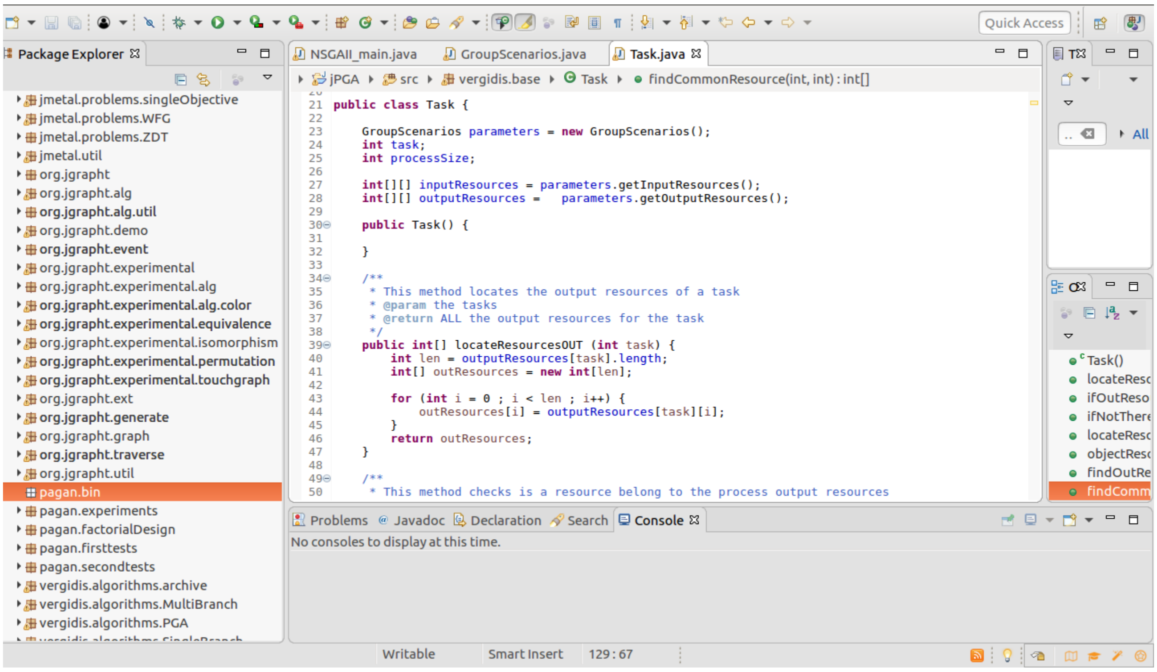


Figure 3.12. Screenshot of the Java programming environment

Setting up a bpo test problem

As the generation of scalable business process test problems was mentioned as an objective of this dissertation, it is important to introduce the procedure of setting up a bpo test problem at this point. When a new user of the framework begins to work with it, it is important to create a new library that will store packages relating with the problem that is examined. Any package could store classes with various problem parameters similar to the classes that are included in the 'Parameters' package of Vergidis library. Also, the algorithm parameters have to be specified. Figure 3.13 shows an example.

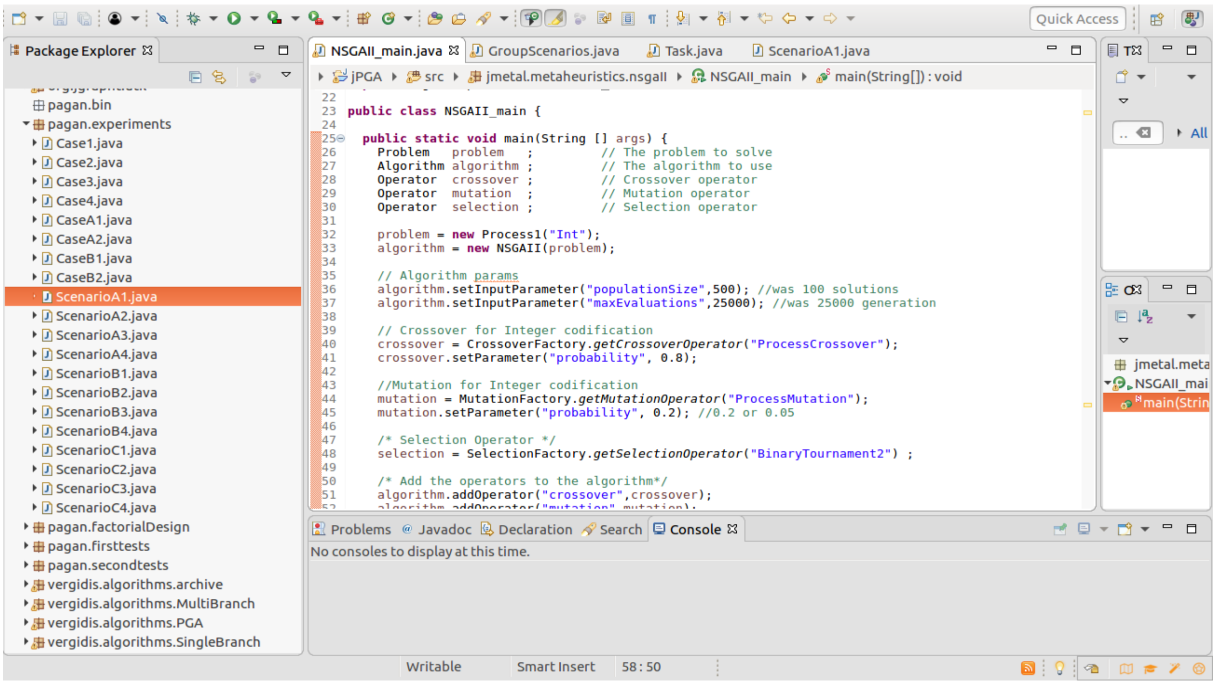


Figure 3.13. Screenshot of setting up NSGA2 parameters

Given the requirements of the problem (i.e. n, n_d, r, t_{in}, t_{out}, r_{in}, r_{out}, and specific limits for each attribute) the procedure consists of two steps, as follows:

1. Generation of task library

   In this step a library with a defined number of tasks (n) is generated. The parameters generated for each task are the set of input resources (R_{in}), the set of output resources (R_{out}) and the set of task attributes values (TA_i). These parameters have to be in accordance with compulsory constraints mentioned in section 3.2. In order to generate those, a spreadsheet software is employed. Figure 3.14 illustrates an example. The values of the parameters are produced with the Random function of the spreadsheet software for values defined from the problem requirements.

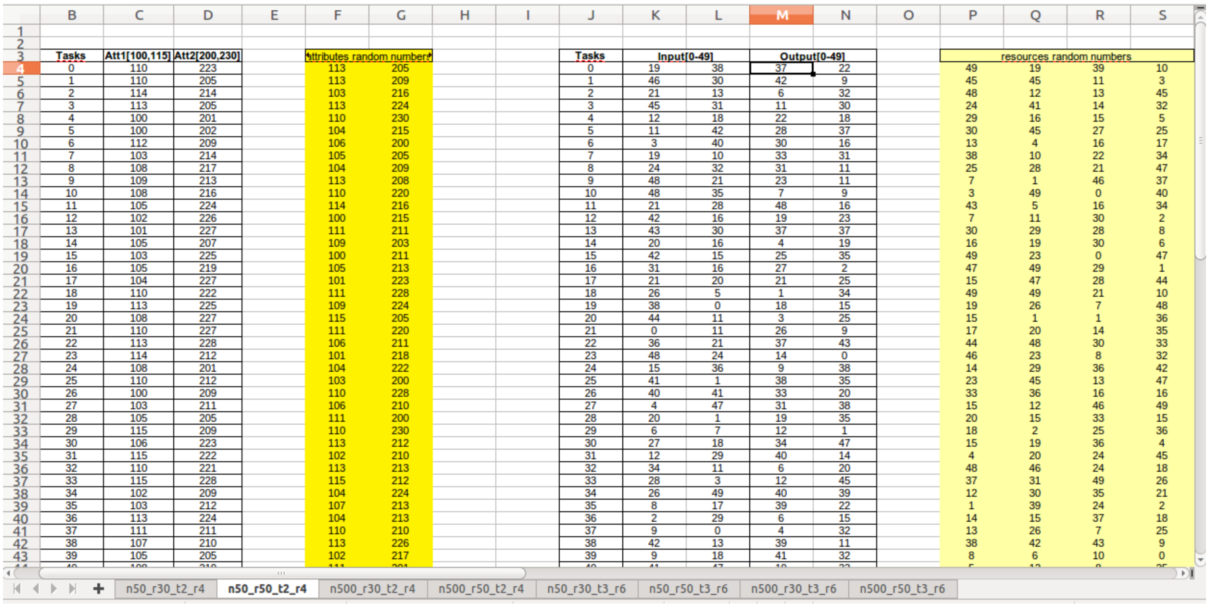


Figure 3.14. Screenshot of a task library

2. Construction of the appropriate class

   During this step a class in the package is created. This class depicts the business process test so all the problem parameters have to be written down. Thus, the data that were generated in the previous step are transferred into the framework. The rest of parameters are filled in by the user. Figure 3.15 shows an example of bpo test.

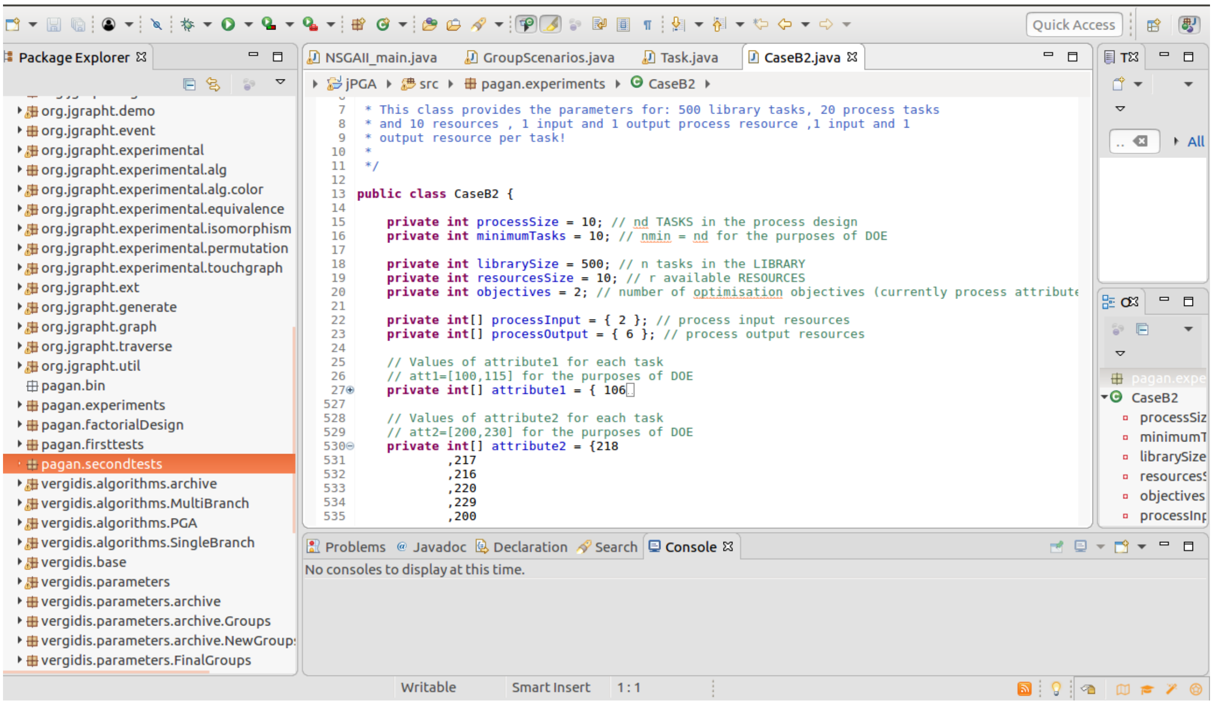


Figure 3.15. Screenshot of a bpo test

3.5 Validation of experimental results

This section reviews some experiments have been done by Vergidis (2008). The validation of experimental results indicates the accuracy and the reliability of Vergidis work. In addition, this type of experiments is a first look at the way we are going to investigate the framework in later chapters. In the context of this dissertation we utilize only the NSGA2 from the EMOAs.

Vergidis (2008) investigated the performance of the framework using a series of experimental scenarios. Particularly, he generated three group of experiments (scenarios) to discover the minimum library size (n) that the framework can operate with (scenario A), the maximum size of a business process design (n_d) that can be optimized (scenario B) and how the minimum number of tasks (n_{min}) in a business process design affects the number of optimized business process designs (scenario c).

For the purpose of the review three experiments are selected, one of each scenario. Table 3.6 presents the problem parameters for each experiment.

Parameter	Experiment 1	Experiment 2	Experiment 3
n	100	30	60
n_d	10	10	20
n_{min}	8	6	6

r	20	20	20
t_{in}/t_{out}	3/3	3/3	3/3
r_{in}/r_{out}	5/5	5/5	5/10
p	2	2	2
α	100-115	100-115	100-115
β	200-230	200-230	200-230

Table 3.6. Problem parameters for each experiment

As it seems in Table 3.6 the process/task attributes are α and β. Attribute α varies between values 100 and 115 and attribute β varies between values 200 and 230. The number of task input and output resources is the same. The number of process input and output resources is the same except the number of output process resources for the third experiment which is 10. Table 3.7 shows how the parameters of the NSGA2 were defined for these experiments.

Parameter	**NSGA2**
Population	500
Generations	25,000
Crossover prob.	0.8
Mutation prob.	0.2

Table 3.7. Parameter specification for the NSGA2

The experiments set up as it was described in the previous section. The execution was repeated three times in order to obtain a more precise estimate. Table 3.8 shows the results of the executions.

	Vergidis solutions	Test 1 solutions	Test 2 solutions	Test 3 solutions
Experiment 1	45	41	42	49
Experiment 2	8	7	8	8
Experiment 3	48	46	50	51

Table 3.8. Experimental results

The solutions that are mentioned in Table 3.8 are the alternative optimized business process designs the framework produced. Each repetition is referred to as 'Test'. As it seems in Figure 3.16 the review results are quite close to Vergidis results.

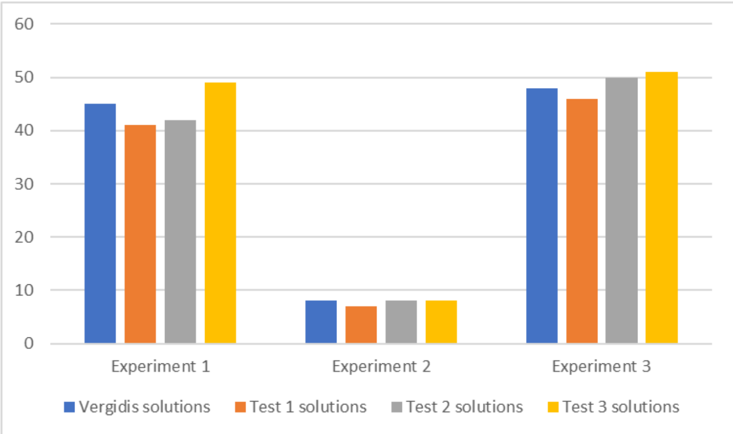


Figure 3.16. Experimental results

To interpret these results, we need to analyze them statistically. For this reason, the following equations are employed:

$$\bar{x} = \frac{\sum x_i}{N} \qquad \text{Equation 3.3}$$

$$\sigma_x = \sqrt{\frac{\sum (x_i - \bar{x})^2}{N-1}} \qquad \text{Equation 3.4}$$

$$\sigma_{\bar{x}} = \frac{\sigma_x}{\sqrt{N}} \qquad \text{Equation 3.5}$$

, where Equation 3.3 is the population mean, Equation 3.4 is the standard deviation and Equation 3.5 is the error of the mean.

The results now can be expressed by the following formula:

$$x = \bar{x} \pm \sigma_{\bar{x}} \qquad \text{Equation 3.6}$$

For each set of tests the values of these equations are calculated. Table 3.9 depicts the results of the calculations.

	$\bar{x}$	σx	$\sigma_{\bar{x}}$	$x = \bar{x} \pm \sigma_{\bar{x}}$
Experiment 1	44	4,35	2,52	44 ± 2,52
Experiment 2	7,66	0,57	0,33	7,66 ± 0,33
Experiment 3	49	2,64	1,53	49 ± 1,53

Table 3.9. Results of statistical analysis

The last column of the Table 3.9 can also be interpreted as the space that the actual value could fall with 68% probability (one sigma). The means can be used for the calculation of another interesting quantity named percentage difference and shown in the Equation 3.7

$$\alpha_i = \frac{|a_v - a_{\bar{x}}|}{a_v} \cdot 100\% \qquad \text{Equation 3.7}$$

, where α_v is the value from Vergidis results and $a_{\bar{x}}$ is the mean of the specific test.

This quantity provides us a comparison measurement between Vergidis (2008) and review results. Table 3.10 shows the findings from this quantity. As it seems the percentage differences are significant small, particularly below 5 %. Thus, we conclude that the results presented by Vergidis (2008) may be characterized as valid.

	αi
Experiment 1	2%
Experiment 2	4%
Experiment 3	2%

Table 3.10. Percentage difference between Vergidis (2008) & review results

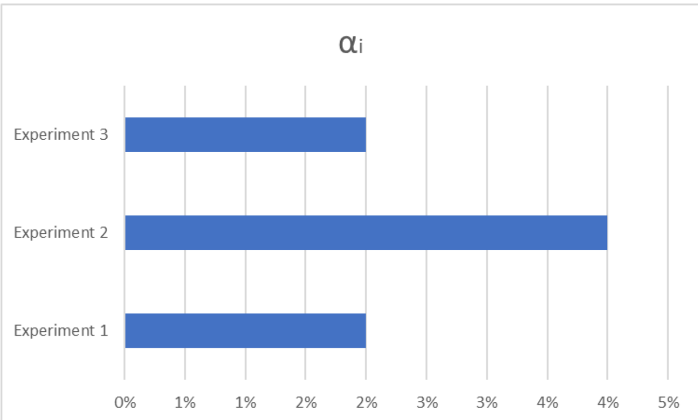


Figure 3.17. Percentage difference between Vergidis (2008) & review results

3.6 Summary

This chapter presented the business process optimization framework (bpo^F) that was proposed by Vergidis (2008). The proposed framework employs EMOAs to generate optimized results and is based on a representation that has mathematical characteristics. These characteristics are expressed by the problem parameters. From the study of bpo^F the need for a more detailed investigation of problem parameters is raised. The next chapter presents a statistical approach that is called Design of Experiments (DoE) and it can be employed for the investigation of these parameters.

Chapter 4 – Design of experiments

This chapter introduces the Design of Experiments. Design of experiments (DoE) can be considered as a systematic method to determine the relationship between factors affecting a process and the output of that process. In other words, it is used to find cause-and-effect relationships. The benefits of DoE are that it provides strategies to set up an experiment efficiently and effectively as well as statistical tools to analyze and interpret the results. In this chapter the basic principles of experimentation and the main concepts of DoE are presented. Moreover, main features and tasks of the Minitab statistical software that automates calculations are provided at the end of this chapter.

4.1 Introduction

An experiment is a test or a series of tests in which purposeful changes are made to the input variables of a process or a system so that we may observe and identify the reasons for changes that may be observed in the output response. We may want to determine which input variables are responsible for the observed changes in the response, develop a model relating the response to the important input variables and use this model for process or system improvement or other decision-making processes. So we can say that experiments are used for studying the performance of processes and systems.

According to Montgomery (2017), experimentation is a vital part of the scientific (or engineering) method. There are situations where the scientific phenomena are so well understood that useful results including mathematical models can be developed directly by applying well-understood principles. The models of such phenomena that follow directly from the physical mechanism are usually called **mechanistic models**. However, most problems in science and engineering require observation of the system at work and experimentation to explain information about why and how it works. Well-designed experiments can often lead to a model of system performance. These models are called **empirical models**. A well-designed experiment is important because the findings and the results that can be taken from that, depend on a great extent to the way the data are collected.

In general, we can say that we learn how systems or processes work through a series of activities in which we make assumptions about a process , perform experiments to generate data from the process and then use the information from the experiment to build new assumptions, which lead to new experiments, and so on. Thus, we can regard that experimental design is a fundamental tool to understand the way that systems or processes work. Montgomery (2017) supports that the application of experimental design techniques can result in

1. Improved process yields
2. Reduced variability and closer conformance to nominal or target requirements
3. Reduced development time
4. Reduced overall costs.

4.2 Fundamentals of designing an experiment

This section introduces the basic principles that we have to be aware of when we are going to design an experiment as they are provided by Montgomery (2017). At first, we can define the statistical design of experiments as the process of planning the experiment so that appropriate data will be collected and analyzed by statistical methods, resulting in valid and objective conclusions. The statistical approach to experimental design is necessary if we wish to draw meaningful conclusions from the data. Generally, when the problem involves data that are subject to experimental errors, statistical methods are the only objective approach to analysis. Thus, there are two aspects to any experimental problem: the design of the experiment and the statistical analysis of the data. These two subjects are closely related because the method of analysis depends directly on the design employed.

There are three basic principles for designing experiments:

- **Randomization** is the most important principle in experimental design. By randomization we mean that both the allocation of the experimental material and the order in which the individual runs the experiment to be performed are randomly determined. Statistical methods require that the observations (or errors) be independently distributed random variables. Randomization usually makes this assumption valid. In addition, an experiment that is randomized suitably can diminish the effects of extraneous factors that may be present.
- **Replication** is another principle in experimental design. By replication we mean an independent repeat run of each factor combination. Replication has two significant properties:
    1. The experimenter can obtain an estimate of the experimental error. This estimate of error becomes a basic unit of measurement for determining whether observed differences in the data are really statistically different.
    2. If the sample mean is used to estimate the true mean response for one of the factor levels in the experiment, replication permits the experimenter to obtain a more precise estimate of this parameter.

  We can say that replication shows sources of variability both between runs and within runs.
- **Blocking** is the last principle in experimental design. By blocking we mean a design technique that is used to improve the precision with which comparisons among

the factors of interest are made. We often use blocking to reduce the variability transmitted by nuisance factors. Nuisance factors can be defined as factors that influence the experiment response but we are not interested in them.

4.3 Process of designing an experiment

This section presents the steps of designing an experiment. The process of designing and afterwards analyzing an experiment requires from experimenters to have a clear idea of what will be studied, how the data will be collected and how these data will be analyzed. According to Montgomery (2017) the process of designing an experiment is as following:

1. **Recognition and statement of the problem.**
   The first thing that an experimenter should do is a list of specific problems or inquiries which would be answered by the experiment. Also, an explicit statement of the problem helps understanding the object of studying. An experiment can be executed for some reasons, for instance:
   - i. **Factor screening or characterization.**
     When something is new, it is usually important to learn which factors have the most influence on the response(s) of interest.
   - ii. **Optimization.**
     When the important factors would be characterized, we could find the settings or levels of them that result in desirable values of the response. Usually a screening experiment cannot produce optimal settings of the important factors. Thus, an optimization experiment follows up a screening experiment
   - iii. **Confirmation.**
     In a confirmation experiment, the experimenter tries to verify that everything operates in a way that is consistent with some theory or past experience.
   - iv. **Discovery.**
     In discovery experiments, the experimenter tries to determine what happens when we explore new materials, or new factors, or new ranges for factors.
   - v. **Robustness.**
     In these experiments, the experimenter tries to determine how the factors that could be controlled could be set in a way to minimize the variability transmitted into the response from factors that cannot be controlled very well.
2. **Selection of the response variable.**
   The next phase of designing an experiment is to select the response variable. The experimenter should be certain that this variable really provides suitable information. Often, the average or standard deviation (or both) of the measured characteristic will be the response variable. Also multiple responses are not unusual.
3. **Choice of factors and levels.**

In every experiment the factors can be classified as **potential design factors** or **nuisance factors.** The experimenter wishes to vary the potential design factors. In many cases there are many potential design factors and some further classification can take place as follows:

- *Design factors.* These factors are selected for study in the experiment.
- *Held-constant factors.* These factors may have a small influence in the response, but the experimenter is not interested in them, so he keeps them at a specific level.
- *Allowed -to -vary factors.* These factors have a small influence in the response, so the experimenter is not interested in them.

As regards to the nuisance factors, they may have significant effect on the response, but the experimenter may not be interested in them in that experiment. When the experimenter has selected the design factors, he must choose the levels at which runs will be made. It is ordinary to keep the factor levels low in a factor screening experiment.

4. **Choice of experimental design.**
When the experimenter chooses a design, he has to think about the sample size, the number of replications and if there will be any restriction as blocking. There are statistical software packages that can provide support to this decision.
5. **Performing the experiment.**
During the experiment's execution, the experimenter must monitor the process to ensure that everything is done correctly. Any error would destroy experimental validity. Some trial runs may be helpful.
6. **Statistical analysis of the data.**
Statistical methods should be used to analyze the data and to conduct safe results and conclusions. The statistical software packages can assist in data analysis. We will discuss more about these methods later.
7. **Conclusions and recommendations.**
After data analysis the experimenter should conclude in practical results and recommendations about the process and what else can be studied.

These steps are considered to be the base of every experiment. We are going to use them as a guideline in this dissertation.

4.4 Hypothesis testing

This section introduces the term of hypothesis testing, which can be considered as the reason of performing an experiment from statistical perspective. A statistical hypothesis is a statement either about the parameters of a probability distribution or the parameters of a model. A model describes the results of an experiment. An example of a model is what follows below:

$$y_{ij} = \mu_i + \varepsilon_{ij} \begin{cases} i = 1,2 \\ j = 1,2, \dots, n_i \end{cases} \qquad \text{Equation 4.1}$$

where y_{ij} is the jth observation from factor level i, μ_i is the mean of the response at the ith factor level, and ε_{ij} is the random error.

The hypothesis shows a conjecture about the experiment. For example, assume an experiment with a single factor was executed for two factor's levels and gave us two sets of values. The levels may be quantitative, such as values of temperature or time; or may be qualitative, such as types of machines. Each set can be considered as a level of the factor and each value is the result of an experiment run. A hypothesis can be

$$H_0 : \mu_1 = \mu_2$$
$$H_1 : \mu_1 \neq \mu_2$$

where μ_1 is the mean of the values for the first level and μ_2 is the mean of the values for the second level. The statement H_0: $\mu_1 = \mu_2$ is called the **null hypothesis** and H_1: $\mu_1 \neq \mu_2$ is called the **alternative hypothesis**. After doing the computations the experimenter will reject or fail reject the null hypothesis.

Generally, there are two types of errors that can be done in hypothesis testing. The first type or type I is when the null is rejected but it is true, and the second type or type II is when the null hypothesis is false but is not rejected. The probabilities of these errors are

$$\alpha = P(\text{type I error}) = P(\text{reject } H_0 \mid H_0 \text{ is true})$$
$$\beta = P(\text{type II error}) = P(\text{fail reject } H_0 \mid H_0 \text{ is false})$$

Another probability that is used often is the **power** of the test, where

$$\text{Power} = 1 - \beta = P(\text{reject } H_0 \mid H_0 \text{ is false})$$

The probability α is also called **significance level** of the test which is quite important in hypothesis testing and is the one that we will employ in this dissertation.

To better understand the significance level, assume we have a t-distribution with a population mean that equals to 260 (null hypothesis), and sample mean that equals to 330.6. Figure 4.1 shows the probability distribution plot of this assumption. Assume that the population mean occurred from past measurements and the sample mean has occurred from current measurements. What we want to determine is whether our sample mean indicates that there is a significantly different from the population mean.

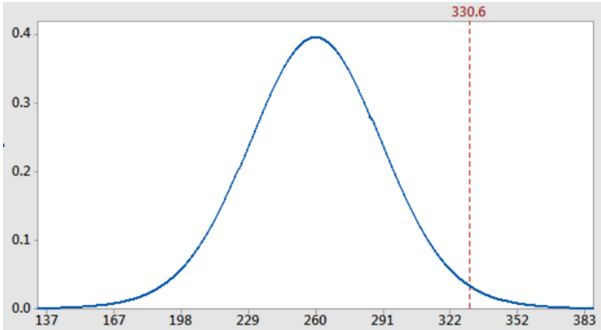


Figure 4.1. Population and sample mean

The significance level determines how far out from the null hypothesis value we'll draw that line on the graph. To graph a significance level of 0.05, we need to shade the 5% of the distribution that is further away from the null hypothesis (Figure 4.2).

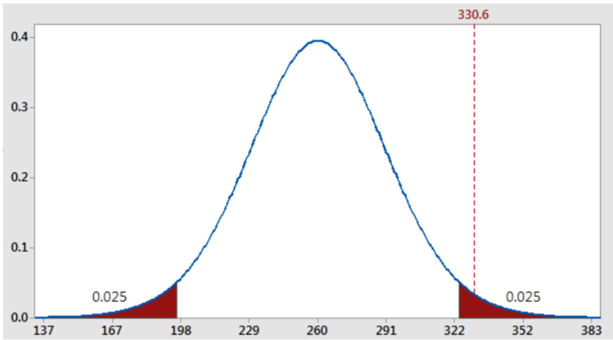


Figure 4.2. Significance level of 0.05

In the graph above, the two shaded areas are at equal distances from the null hypothesis value and each area has a probability of 0.025, for a total of 0.05. These shaded areas are called the **critical region**. If the population mean is 260, we would expect to obtain a sample mean that falls in the critical region 5% of the time. The critical region defines how far away our sample statistic must be from the null hypothesis value before we can say it is unusual enough to reject the null hypothesis. Our sample mean (330.6) falls within the critical region, which indicates it is statistically significant at the 0.05 level.

P-value

It is customary to inspect the hypothesis testing results by comparing the a-value (or significance level) with the **P- value**. The P- value can be defined as the smallest level of significance that would lead to rejection of the null hypothesis H_0. Thus, the experimenter can set a value for the significance level (most common value is 0.05) and then compute the P-value. If the P- value is less than or equal to the significance level, the null hypothesis will be rejected. In addition, P- value shows whether the computed data was just barely in the critical region or whether it was very far into this region. Figure 4.3 illustrates a P-value equal to 0.03 for the example that was described previously (sample mean= 330.6, population mean= 260). So in experiments with many factors, P-value can indicate how important a factor is. P- value can be easily computed by statistical software packages.

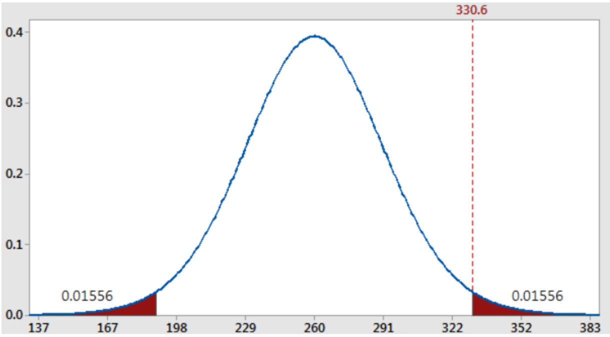


Figure 4.3. P- Value equal to 0.03

4.5 Experimental designs and statistical analysis

This section presents experimental designs and the appropriate statistical approach for analyzing them. In the context of this dissertation we are going to study and imply designs and statistical techniques related to factor characterization. The factors that we will study are parameters of the business process optimization problem. We will analytically refer to them further down. So designs that are related with other experimental purposes such as optimization, blocking or robustness are not going to be mentioned in this dissertation because they constitute a subject for a separate study. Regarding the factor characterization, there are two strategies to follow:

1. One factor at a time strategy
2. Factorial strategy

The designs for each strategy are presented below.

Single factor design

In many cases there are experiments, that we can handle only a factor a time. This fact can occur due to the nature of the object that we study. In other cases the experimenter through his experience and his knowledge about the object that he studies, may choose to experiment this way. To introduce the process of designing and analyzing this kind of experiment, suppose an experiment in which we would like to study the way a single factor affects the observed response. We would need to take n measurements of the observed response for the α levels of the factor. The levels may be quantitative, such as values of temperature or time; or may be qualitative, such as types of machines. The data could be gathered in a table as below (Table 4.1):

Level	Measurements				Totals	Averages
1	y_{11}	y_{12}	. . .	y_{1n}	y_1	$\bar{y}_1$
2	y_{21}	y_{22}	. . .	y_{2n}	y_2	$\bar{y}_2$
. . .	. . .	. . .	. . . .	. . .	. . .	. . .
α	$y_{\alpha 1}$	$y_{\alpha 2}$	. . .	$y_{\alpha n}$	y_α	$\bar{y}_\alpha$
					$\Sigma\Sigma y_{ij}$	$\Sigma\bar{y}_i$

Table 4.1. Data of a Single-Factor experiment

As mentioned before in this chapter, a model can be used to describe the observed measurements as the following:

$$y_{ij} = \mu_i + \varepsilon_{ij} = \mu + \tau_i + \varepsilon_{ij} \begin{cases} i = 1,2, \dots, a \\ j = 1,2, \dots, n \end{cases} \qquad \text{Equation 4.2}$$

where y_{ij} is the *ij*th observation, μ_i is the mean of the ith factor level, μ is a parameter common to all levels called the **overall mean**, τ_i is a parameter unique to the *i*th level called the ***i*th levels effect,** and ε_{ij} is a **random error** component that incorporates all other sources of variability in the experiment including measurement and variability arising from uncontrolled factors.
Equation 4.2 is called **one-way** or **single-factor analysis of variance (ANOVA)** model.

In this design, the goal from statistical perspective is to test the equality of the means for each level. In other words the hypothesis testing is:

$H_0 : \mu_1 = \mu_2 = \cdots = \mu_\alpha$
$H_1 : \mu_1 \neq \mu_2$ for at least one pair (i,j)

An equivalent way to write the hypothesis is with the use of the level effect τ_i :

$H_0 : \tau_1 = \tau_2 = \cdots = \tau_\alpha = 0$
$H_1 : \tau_1 \neq 0$ for at least one i

Thus, we speak of testing the equality of level means or testing that the level effects (the τ_i) are zero. The appropriate procedure for testing the equality of a treatment means is the analysis of variance. The **analysis of variance** shows the total variability in the data. The ANOVA is based on the sum of squares (Equation 4.3) and particularly in the partitioning of the total variability, as measured by the total sum of squares into its component parts. The fundamental equation of ANOVA is shown below (Equation 4.4).

$SS = \sum_{i=1}^{n}(y_i - \bar{y})^2$ Equation 4.3

$SS_T = SS_{Levels} + SS_E$ Equation 4.4

, where SS_T is the total sum of squares, SS_{Levels} is the sum of squares due to levels (i.e., between levels), and SS_E is the sum of squares due to error (i.e., within levels).

From the Equation 4.4 two important quantities called **mean squares** are extracted (Equation 4.5 & Equation 4.6). The ratio of these two quantities is called **F-test** (Equation 4.7). F-test plays an important role in the analysis of variance (ANOVA) and it can be used as an indicator for rejecting or fail rejecting hypothesis, alternatively to P- Value approach.

$MS_{Levels} = \frac{SS_{Levels}}{a-1}$ Equation 4.5

$MS_E = \frac{SS_E}{N-a}$ Equation 4.6

$F_0 = \frac{MS_{Levels}}{MS_E}$ Equation 4.7

, where N is the total measurements and α is the number of levels.

The rejection of null hypothesis with the value of a F- test (F_0) is occurred if

$$F_0 > F_{\alpha,a-1,N-a}$$

, where N is the total measurements and α is the number of levels.

The quantity $F_{\alpha,\alpha-1,N-\alpha}$ is called critical value and it is calculated from an F-distribution table that is contained in most statistic books.

Another useful tool of statistical analysis is the residuals. The residual can be defined as the difference between the observed value (i.e. the measurement) and the predicted value from the model. Figure 4.4 illustrates residuals for a random model. The residual for a measurement *j* in the factor level *i* can be expressed by the following equation:

$e_{ij} = y_{ij} - \hat{y}_{ij}$ Equation 4.8

, where $\hat{y}_{ij}$ is the predicted value of the model for the y_{ij} measurement.

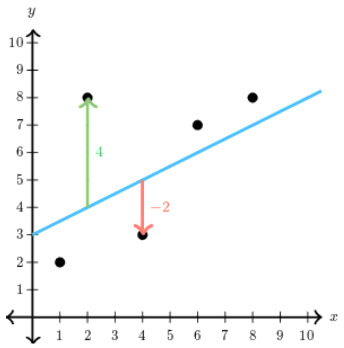


Figure 4.4. Example of residuals

Examination of the residuals is a basic part of an analysis of variance because it can show how adequately the measurements are described by the model (Equation 4.2) and that the errors in the model are normally and independently distributed. The examination is based on two main residual plots, the normal probability plot and the plot of residuals versus fitted values. The first is employed in order to verify that the residuals are normally distributed, while the second in order to verify the model adequacy. Particularly if the model is adequate, the plot of residuals versus fitted values should be structureless. An example of these plots is shown in Figure 4.5.

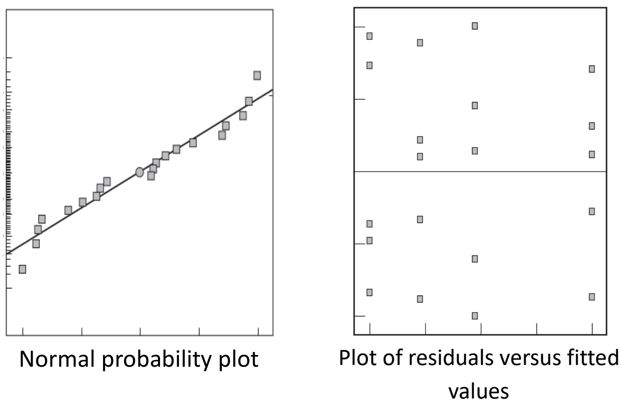


Figure 4.5. Example of residuals plots

Finally, for the interpretation of the results we are interested in finding an equation for the response variable of the experiment. This equation is an **empirical model or regression model** and the general approach to fitting empirical models is called **regression analysis**. For example, we may fit a linear model to the data (Equation 4.9) or a quadratic model (Equation 4.10). Examples of the diagrams of these models are shown in Figure 4.6 and Figure 4.7 respectively.

$$y = a_0 + a_1 x + \varepsilon \qquad \text{Equation 4.9}$$

$$y = a_0 + a_1 x + \alpha_2 x^2 + \varepsilon \quad \text{Equation 4.10}$$

,where α_0, a_1 and a_2 are unknown parameters that we will have to estimate and ε is a random error term.

In general, we would like to fit the lowest order polynomial because high-order polynomial increases complexity. These equations and the other things that we discussed above can easily be computed from the statistical software packages.

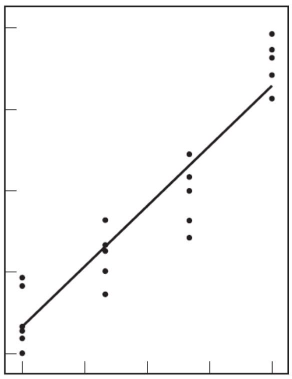


Figure 4.6. Diagram of a linear model

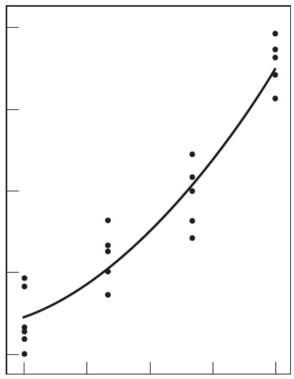


Figure 4.7. Diagram of quadratic model

To sum up, we not only presented the procedure of designing and analyzing experiments with a single factor, but we also introduced basic terms used in every design.

Factorial design

In an experiment that we are interested in studying the effect of two or more factors the most efficient way to do so, is the factorial design. An important advantage of this design is that it can show any interaction between the factors. The basic concept is to investigate all possible combinations of the levels of the factors and specify the effects of the factors on the observed response. The **effect** of a factor is defined as a change in response produced by a change in the level of the factor. In many cases, we will find that the difference in response between the levels of one factor is not the same at all levels of the other factors. When this occurs, there is an **interaction** between the factors.

To introduce this design, suppose we have an experiment with two factors A and B. The factor A has α levels, the factor B has β levels and we took n measurements for the observed response (n replications). The levels may be quantitative, such as values of temperature or time; or may be qualitative, such as types of machines. The data can be gathered in a table as below (Table 4.2).

	Factor B				
Factor A	Level	1	2	· · ·	β
	1	$y_{111}, y_{112}, \ldots, y_{11n}$	$y_{121}, y_{122}, \ldots, y_{12n}$	· · ·	$y_{1\beta 1}, y_{1b2}, \ldots, y_{1\beta n}$
	2	$y_{211}, y_{212}, \ldots, y_{21n}$	$y_{221}, y_{222}, \ldots, y_{22n}$	· · ·	$y_{2\beta 1}, y2_{\beta 2}, \ldots, y_{2bn}$
	. . .	. . .	. . .	. . .	. . .
	α	$y_{\alpha 11}, y_{\alpha 12}, \ldots, y_{\alpha 1n}$	$y_{\alpha 21}, y_{\alpha 22}, \ldots, y_{\alpha 2n}$	· · ·	$y_{\alpha\beta 1}, y_{\alpha\beta 2}, \ldots, y_{\alpha\beta n}$

Table 4.2. Arrangement for a Two-Factor Factorial Design

The measurements for the observed response can be described by a model, as follows:

$$y_{ijk} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} + \varepsilon_{ijk} = \mu_{ij} + \varepsilon_{ijk} \begin{cases} i = 1,2, \ldots, a \\ j = 1,2, \ldots, b \\ k = 1,2, \ldots, n \end{cases} \qquad \text{Equation 4.11}$$

where μ is the overall mean effect, τ_i is the effect of the *i*th level of the row factor A, β_j is the effect of the *j*th level of column factor B, $(\tau\beta)_{ij}$ is the effect of the interaction between τ_i and β_j ,and ε_{ijk} is a random error component. The effects can also be defined as **deviations** from the overall mean.

The hypothesis testing for this design would be about the equality of row level effects as follows

H_0: $\tau_1 = \tau_2 = \ldots = \tau_\alpha = 0$
H_1: at least one $\tau_i \neq 0$

and the equality of column level effects as follows

H_0: $\beta_1 = \beta_2 = \ldots = \beta_b = 0$
H_1: at least one $\beta_j \neq 0$

In addition, we are interested in determining interact between the factors. Thus,

H_0: $(\tau\beta)_{ij} = 0$
H_1: at least one $(\tau\beta)_{ij} \neq 0$

The results of hypothesis testing can come from a statistical procedure of analysis named **two-factor analysis of variance**. The quantities that we need for the statistical analysis of this design are gathered in the table (ANOVA table) below (Table 4.3).

Source of variance	Sum of Squares	Degrees of Freedom	Mean Square	F_0
A	SS_A	α-1	$MS_A = \frac{SS_A}{a-1}$	$F_0 = \frac{MS_A}{MS_E}$
B	SS_B	β-1	$MS_B = \frac{SS_B}{b-1}$	$F_0 = \frac{MS_B}{MS_E}$
Interaction	SS_{AB}	(α-1) (β-1)	$MS_{AB} = \frac{SS_{AB}}{(a-1)(b-1)}$	$F_0 = \frac{MS_{AB}}{MS_E}$
Error	SS_E	αβ(n-1)	$MS_A = \frac{SS_E}{ab(n-1)}$	
Total	SS_T	αβn-1		

Table 4.3. The Analysis of Variance Table for the Two-Factor Factorial

where A is the level effects of factor A, B is the level effects of factor B, Interaction is the effect that occurs when factors interact and degrees of freedom is the number of elements in the sum of squares that are free to vary. The other three quantities (Sum of squares, Mean Square and F_0) are similar to these of the single-factor analysis of variance. An example of a two-factor factorial experiment is shown in Figure 4.8. In this example the impurity of a chemical product is observed for the three temperature (factor A) levels and the five pressure levels (factor B). Figure 4.9 presents the ANOVA table of this example.

Temperature	Pressure					
(°F)	25	30	35	40	45	$y_{i.}$
100	5	4	6	3	5	23
125	3	1	4	2	3	13
150	1	1	3	1	2	8
$y_{.j}$	9	6	13	6	10	$44 = y_{..}$

Figure 4.8. Example of collected data

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F_0	P-Value
Temperature	23.33	2	11.67	42.97	0.0001
Pressure	11.60	4	2.90	10.68	0.0042
Nonadditivity	0.0985	1	0.0985	0.36	0.5674
Error	1.9015	7	0.2716		
Total	36.93	14			

Figure 4.9. Example of ANOVA table

These quantities are easily computed by the statistical software packages and the F- test can be used for rejecting or fail rejecting hypothesis as it was described for a single-factor design. In addition to the basic analysis of variance, the software packages display some other useful information, as the P-Value that is used to show the significance of every effect in a way that we have analyzed in a previous section, and the quantity **R-squared** (Equation 4.12). This quantity is interpreted as the proportion of the variability in the data explained by the ANOVA model.

$$R^2 = \frac{SS_{Model}}{SS_T} \qquad \text{Equation 4.12}$$

, *where* $SS_{Model} = SS_A + SS_B + SS_{Interaction}$.

After analysis of variance (ANOVA), the model adequacy can be checked using residual analysis and then a regression analysis can follow to express the results in terms of a regression model. These two procedures are the same as they were described in the previous subsection. The regression model and the residual plots are easily produced by the statistical software packages.

2^k Factorial design

The example with the two factors where are many levels, constitutes a general case of factorial design that is not used frequently. The most common case in an experiment is this with many factors. Therefore a special case of factorial design is used, the **2^k factorial design**. This design has *k* factors each at only two levels and the levels may be quantitative, such as two values of temperature or time; or may be qualitative, such as two machines, the "high" and "low" levels of a factor. A complete replication of such a design requires 2 × 2 × . . . × 2 = 2^k measurements of the observed response. So this design simplifies the way that the data needed are collected. For example, a 2^2 design needs four measurements for the four factor combinations:

1. Factor A at low level / Factor B at low level

2. Factor A at high level / Factor B at low level
3. Factor A at high level / Factor B at high level
4. Factor A at low level / Factor B at high level

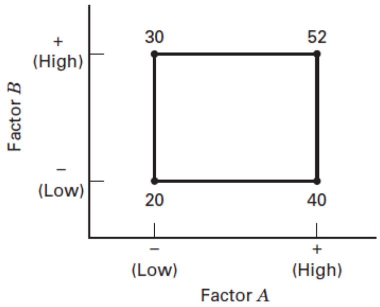


Figure 4.10. 2^k Factorial design with the response shown at the corners

Figure 4.10 illustrates the four combinations employing random numbers for the observed responses. As we can see the 2^k design is particularly useful because it provides the smallest number of runs with which *k* factors can be studied. In addition, in experiments involving 2^k design it is feasible for the software packages to calculate the magnitude and the direction of factor effects (negative value shows reduction - positive value shows increase) and determine along with ANOVA which variables are likely to be important. An example of estimated effects for a 2^3 design is presented in Figure 4.11. Thus, the 2^k design gives us the ability to study the effect of many factors easily and make it the ideal design for factor screening experiments. Finally, the procedure of analysis is similar to the other two designs that were presented before and it is summarized in Table 4.3.

Factor	Effect Estimate
A	−101.625
B	7.375
C	306.125
AB	−24.875
AC	−153.625
BC	−2.125
ABC	5.625

Figure 4.11. Example of estimated effects

Analysis Procedure for a 2^k Design
1.Collect data
2.Perform statistical testing
3.Interpret results

Table 4.4. Main steps of analysis procedure

This section presented the main designs for factor characterization. What has to be calculated is challenging but it can be done easily with the support of the statistical software packages. The next section presents the statistical software package that is employed in this dissertation.

4.6 Minitab

This section introduces the Minitab, the statistical software package that is employed in this dissertation. In general, there are a lot of statistical software that someone can find based on the main distinction of the way in which the source code is released, and that is open-source or licensed software. Also, statistical computing can be performed in the R programming language. Montgomery (2017) suggests three statistical software for the support and the performance of Design of Experiments and these are the following:

- Design – Expert
- JMP
- Minitab

All three statistical software are licensed software and an annual license costs a lot. According to the software's sites, Design – Expert and JMP are employed for professional use while the Minitab is quite widespread for statistics education as more than 4,000 colleges and universities worldwide use it, among which the Department of Applied Informatics of the University of Macedonia is. For this reason, the selected software for this dissertation is the Minitab.

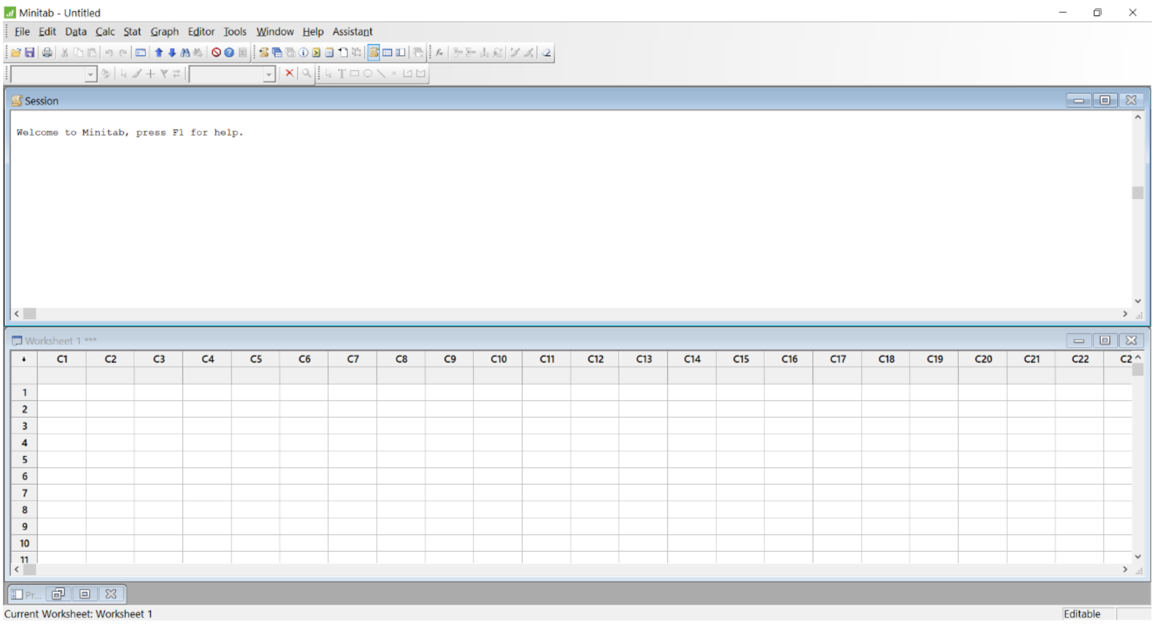


Figure 4.12. Screenshot of the Minitab environment

Minitab provides statistical guidance for a lot of statistical analysis approaches among which we find: Analysis of variance, DOE, Regression analysis, Time series etc. In addition, it provides a wide variety of graphs and many graph editing capabilities. Figure 4.13. illustrates the available features from the Minitab menu.

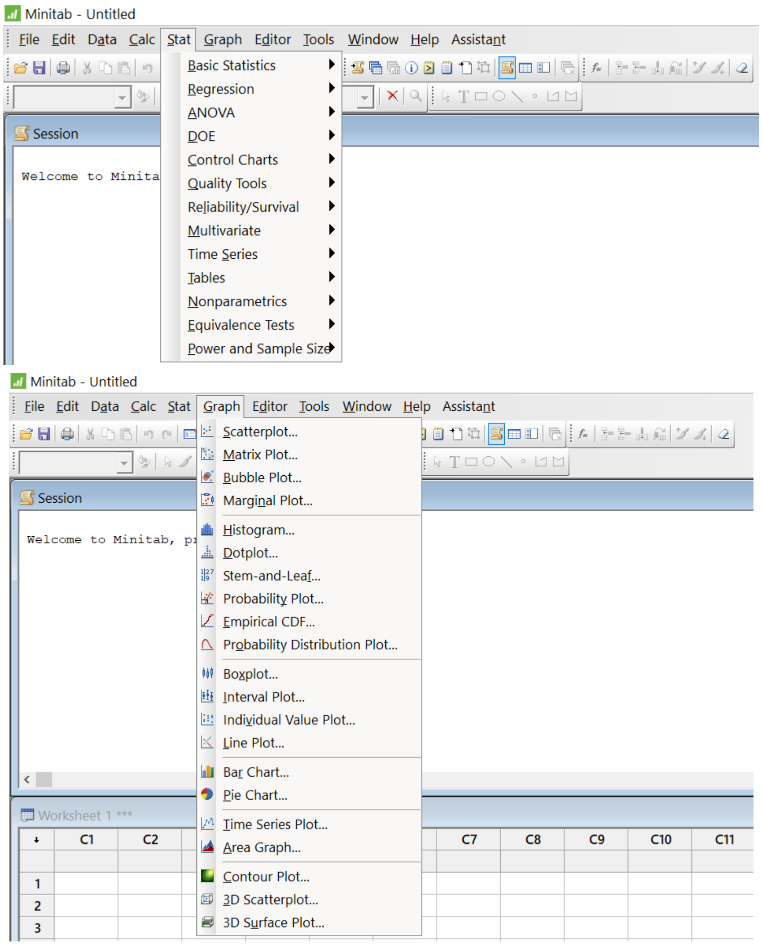


Figure 4.13. Minitab features

As it was mentioned before, in this dissertation we are interested in implying designs and statistical techniques related to factor characterization. Minitab simplifies the procedure of statistical analysis for DoE and provides a lot of graphs in order to interpret the results. Next the main steps for the generation of a Factorial Design are presented citing screenshots:

1. Creation of Factorial Design
   Figure 4.14 presents the commands you have to select in order to create a factorial design. After selecting these commands, a dialog box appears in which you have to select the type of the Factorial Design (px 2^3 design)

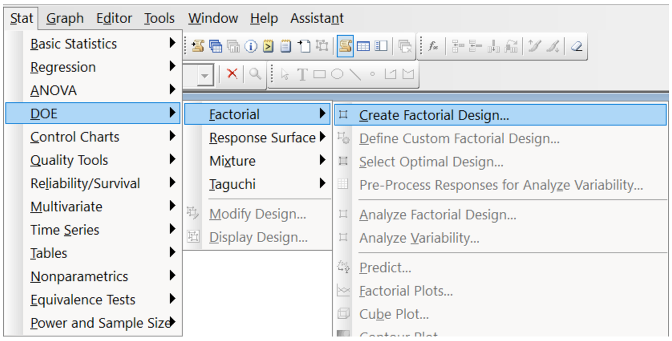
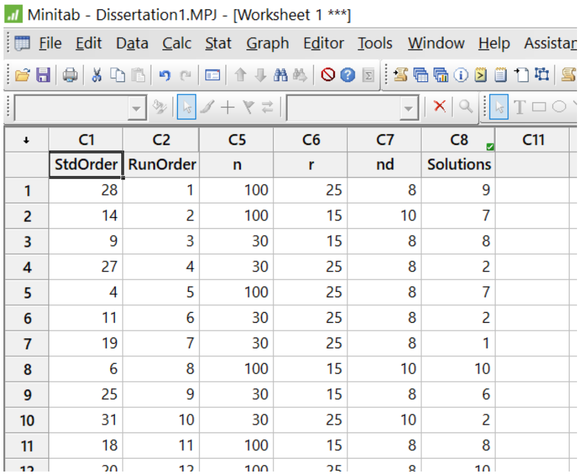


Figure 4.14. Commands for creation of factorial design

2. Entering the data
   The next step is to insert the data. As it seems in Figure 4.12, Minitab provides a worksheet in which the data are inserted. An example of inserted data is presented in Figure 4.15.



↓	C1	C2	C5	C6	C7	C8	C11
	StdOrder	RunOrder	n	r	nd	Solutions	
1	28	1	100	25	8	9	
2	14	2	100	15	10	7	
3	9	3	30	15	8	8	
4	27	4	30	25	8	2	
5	4	5	100	25	8	7	
6	11	6	30	25	8	2	
7	19	7	30	25	8	1	
8	6	8	100	15	10	10	
9	25	9	30	15	8	6	
10	31	10	30	25	10	2	
11	18	11	100	15	8	8	

Figure 4.15. Example of Minitab worksheet

3. Execution of analysis
   Figure 4.16 shows the commands you have to select in order to analyze a factorial design.

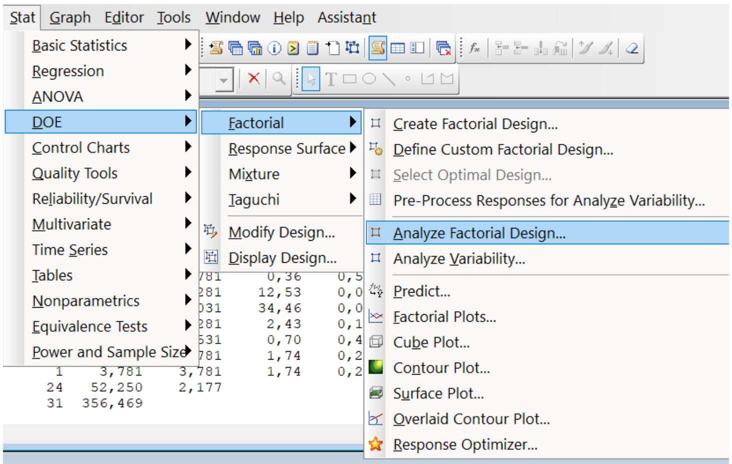


Figure 4.16. Commands for analyze factorial design

Finally, after these steps the Minitab performs the statistical analysis and produces the results. Figure 4.17 illustrates an example of produced results.

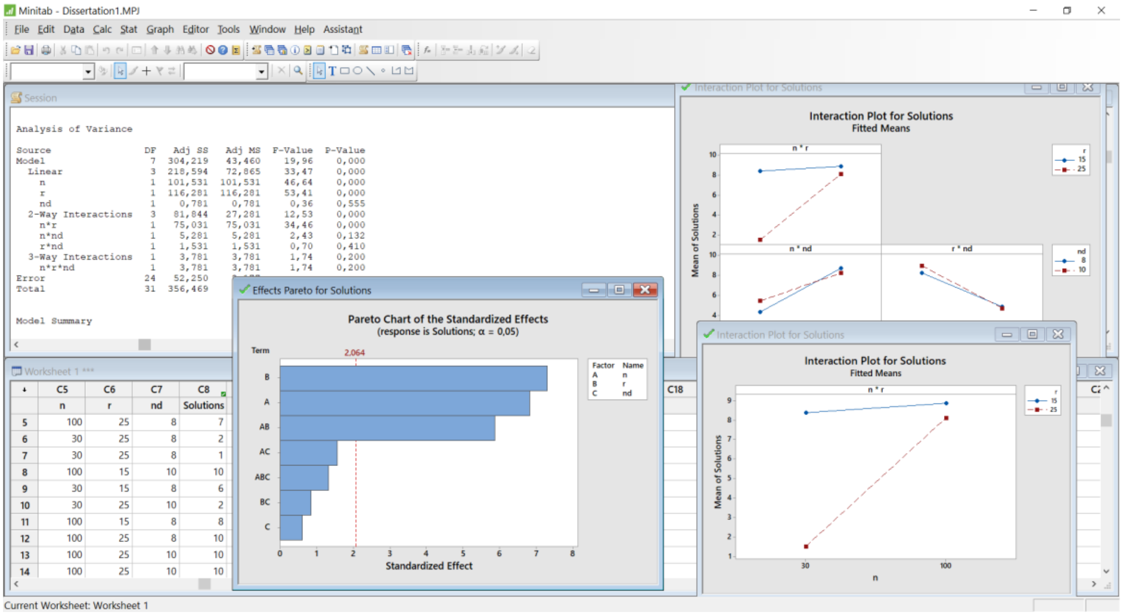


Figure 4.17. Example of results in the Minitab

4.7 Summary

This chapter presented the Design of Experiments. DoE provides to an experimenter approaches for an efficient study of the problem factors. As it was described in this chapter conjectures about the collected data can be tested with statistical methods and practical results, as the significance of a factor, can be concluded. The business problem optimization problem as it was described in the previous chapter includes many factors (problem parameters) and a systematic investigation of these factors is raised as a need. In this investigation, DoE could have a principal role. The next chapter presents the procedure that is employed for the first part of this investigation which determines the limits of the problem parameters that generate reliable results.

Chapter 5 – Generating scalable Business Process test problems

This chapter presents a series of scalable tests in order to investigate the parameters of the business process optimization problem. The tests aim to determine the limits of the parameters in which the bpo^F generates reliable results. The chapter starts by stating the purpose of the tests and the strategy that is followed for the execution of the tests. This strategy proposes the tests to be divided into two scenarios in order to investigate the majority of the parameters. Next, these scenarios are described analytically and the results of the tests for each one of the scenarios are presented.

5.1 Purpose and strategy of the tests

This chapter aims at determining the limits of the business process optimization problem parameters in which the business process optimization framework generates reliable results. As it was described in Chapter 3 the business process optimization framework (bpo^F) utilizes as main component the proposed representation which is described by mathematical parameters and the EMOAs to generate alternative optimized designs. Hence, these parameters are important to the business process optimization problem on the grounds that they affect the generated results of bpo^F. During the performance evaluation of the (bpo^F) that was done by Vergidis (2008), an investigation of the limits of the business process optimization problem parameters having used an experimental approach was presented. Particularly Vergidis (2008) tried to discover the minimum library size (n) that the framework can operate with, the maximum size of a business process design (n_d) that can be optimized and how the minimum number of tasks (n_{min}) in a business process design affects the number of optimized business process designs. This investigation can be characterized as limited and the need for a further investigation is raised. For this reason, a series of business process test problems is generated and is presented next.

Parameter	Description
n	Number of tasks in the library
r	No. of available resources
n_d	No. of tasks in the design
n_{min}	Minimum number of tasks in the design
t_{in}	No. of task input resources

t_{out}	No. of task output resources
r_{in}	No. of process input resources
r_{out}	No. of process output resources
p	No. of task/process attributes

Table 5.1. Main business process optimization problem parameters

For the limits' determination of the business process optimization problem parameters, a strategy was devised. This strategy is based on the experimental approach which Vergidis (2008) employed for the bpo^F performance evaluation and was presented in Chapter 3 (section 3.5). Particularly an experiment can be considered as the execution of the bpo^F for specified problem parameters and the record of the generated solutions. Table 5.2 presents an experiment that was carried out by Vergidis (2008).

Parameter	**Value**
n	100
n_d	10
n_{min}	8
r	20
t_{in}/t_{out}	3/3
r_{in}/r_{out}	5/5
p	2
α	100-115
β	200-230

Table 5.2. Example of Vergidis (2008) experiment

In order to investigate the majority of the parameters, two scenarios are proposed as follows:

1) Scenario A

   This scenario employs process designs that have one process input resource, one process output resource and tasks with one input resource and one output resource.

r_{in}	1
r_{out}	1
t_{in}	1
t_{out}	1

Figure 5.1 illustrates an example of this design. For this type of designs a series of scalable tests follows to determine the values of the *n* (number of tasks), the *r* (number of available resources) and the n_d (number of tasks in the design) that the bpo^F generate reliable results.

2) Scenario B

This scenario employs process designs that have five process input resources, five process output resources and tasks with three input resources and three output resources.

r_{in}	5
r_{out}	5
t_{in}	3
t_{out}	3

Figure 5.2 shows an example of this design. As in previous scenario, for this type of designs a series of scalable tests follows to determine the values of the *n* (number of tasks), the *r* (number of available resources) and the n_d (number of tasks in the design) that the bpo^F generate reliable results.

In both scenarios the tests are groups of experiments in which the parameters *n,r,* n_d vary in a scalable way in order to determine their values that generate reliable results.

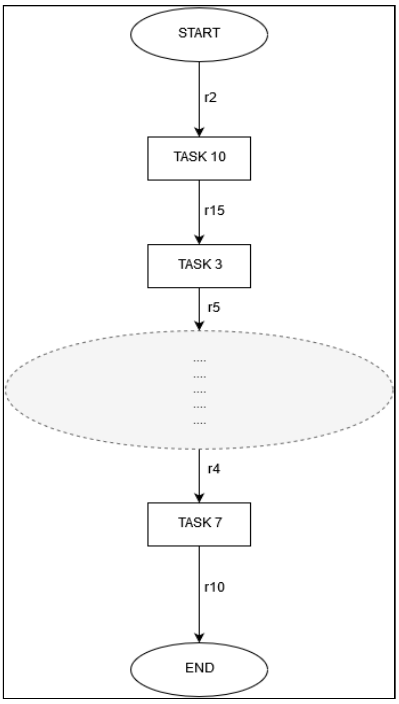


Figure 5.1. Process design of Scenario A

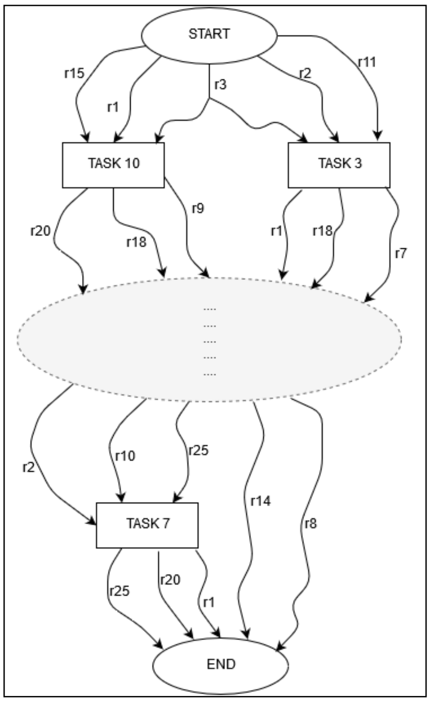


Figure 5.2. Process design of Scenario B

The benefit of the investigation division in two scenarios in which scalable tests are generated, is that not only the *n,r,* n_d parameters are studied but also the r_{in} , r_{out}, t_{in} , t_{out} parameters. Regarding the rest of parameters, n_{min} (minimum number of tasks in the design) and p (number of the task/process attributes), are going to be constant in all tests. Particularly:

- $n_{min} = n_d$
  We are not interested in solutions that have fewer tasks than the design requirement n_d.
- p=2
  The task/process attributes will be 'A' and 'B' and the spaces that vary are as follows:

$$A = [100,115] \qquad\qquad B = [200,230]$$

Also, as it was it was mentioned in Chapter 3, in this dissertation we utilize only the NSGA2 from the EMOAs. The parameters of the NSGA2 are going to be constant in all tests and are defined as follows:

Parameter	**NSGA2**
Population	500
Generations	25,000
Crossover prob.	0.8
Mutation prob.	0.2

To simplify the procedure of data gathering during the execution of the scalable tests, a further partition can take place. In the first part, the tests focus on the limits determination of *n* and *r* parameters while the n_d remains constant in a specified value. In the second part, using the determined limits of these parameters the tests focus on the limit determination of n_d parameter. If this partition does not occur, then a huge number of tests should take place in order to investigate all the possible combinations of these parameters. Finally, the way that the data are collected, is influenced by the Design of Experiments (DoE) and particularly from the table of Factorial Design (Table 4.2). The next section presents the problem tests that are generated in Scenario A.

5.2 Scenario A

The Scenario A includes the first business process problem tests. In these tests the process design has the simplest form that it can have. This fact is expressed by the value of the r_{in}, r_{out}, t_{in} , t_{out} parameters that equals to one. Figure 5.3 illustrates those points that should be taken into consideration for this design.

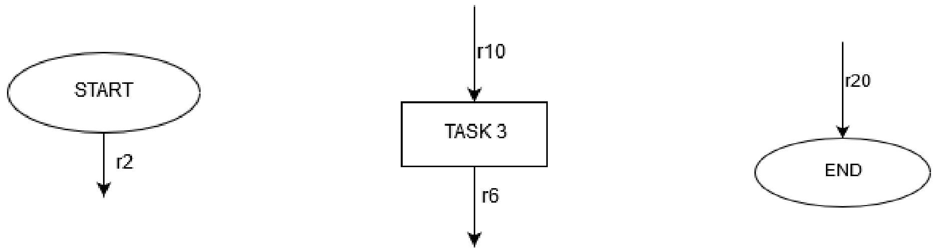


Figure 5.3. Process requirements for Scenario A

As it was mentioned before, the tests are divided into two parts. In the first part the n_d parameter, remains constant in a specified value and the *n*, *r* parameters begin to vary from specified values in a scalable way up until to finding value parameters for which the bpo^F does not generate results. In the second part by employing the lower and the upper limits of *n* and *r* parameters that were determined in the first part, the n_d parameter begins to vary from a specific value in a scalable way up to the point that it finds a value parameter for which the bpo^F does not generate results. Table 5.3 and Table 5.4 summarize this procedure for the respective parts.

n_d=10		**r**		
		10	20	...
n	50	$S_{n=50,r=10}$	$S_{n=50,r=20}$	...
	75	$S_{n=75,r=10}$	$S_{n=75,r=20}$	...
	100	$S_{n=100,r=10}$	$S_{n=100,r=20}$	...
	.	.	.	...
	.	.	.	...
	.	.	.	...

Table 5.3. First part of Scenario A

		n			
		n_-		*n_+*	
		r		**r**	
		r_-	*r_+*	*r_-*	*r_+*
n_d	15	$S_{nd=15,n=n-,r=r-}$	$S_{nd=15,n=n-,r=r+}$	$S_{nd=15,n=n+,r=r-}$	$S_{nd=15,n=n+,r=r+}$
	20	$S_{nd=20,n=n-,r=r-}$	$S_{nd=20,n=n-,r=r+}$	$S_{nd=20,n=n+,r=r-}$	$S_{nd=20,n=n+,r=r+}$
	.	.	.	.	.
	.	.	.	.	.
	.	.	.	.	.

Table 5.4 Second part of Scenario A

In both tables the number of solutions that the bpo^F generated is symbolized with the letter ‘S’ and the indicator shows the value of the parameters in this experiment. In Table 5.5 the lower limit of parameters *n* and *r* is symbolized with the minus sign ‘-‘ and the upper limit is symbolized with the plus sign ‘+’.

First part of the tests

In the first part the n_d parameter is selected to maintain the value ten (n_d=10). This process size can be considered as the minimum that such a simple design can have, like the ones that are being investigated in this scenario, so as to justify the use of bpo^F. The starting point of the investigation for the *n* parameter is the fifty tasks (n =50). This value is selected in order for enough available tasks to exist in the library for the generation of this type of process design. The starting point of the investigation for the *r* parameter is the ten resources (r=10). This value can be considered as a minimum number of available resources that the library can have in order for the bpo^F to generate optimized designs. Table 5.6 presents the tests that are generated for the first part of Scenario A and their results.

	n_d=10		**r**					
			10	20	30	40	50	60
Test1	**n**	50	0	1	1	-	-	-
Test2		75	0	1	1	-	-	-
Test3		100	0	2	1	0	-	-
Test4		150	0	1	2	0	-	-
Test5		200	-	4	5	2	1	0
Test6		300	-	4	3	2	2	0

Table 5.5. Scenario A (Part 1): Tests and results

As it seems in Table 5.6, a test was terminated when the bpo^F did not generate solutions for a value of *r* parameter. Finally, the tests stopped at n=300, a library with three hundred tasks. Considering real life business processes, this number of library tasks can be characterized as large. From the results that are presented in the Table 5.6, we conclude that the n parameter values that generate reliable results are

- n=200 for the *r* parameter limits: r=[20,30]
- n=300 for the *r* parameter limits: r=[20,30]

Reliable results are considered to be three or more solutions generated by the bpo^F.

Second Part of the tests

In the second part, we select a starting point in order to investigate the n_d parameter and we employ the limits of the *n,r* parameters that are extracted from the first part of the tests. As we mentioned before, we consider that the minimum value of the n_d parameter is ten (n_d=10). The starting point of the investigation for the n_d parameter is process sizes with fifteen tasks (n_d =15). The difference between the starting point and the minimum value of n_d parameter can be considered as necessary difference for such a simple design, like the ones that are being investigated in this scenario. The extracted limits are shown in Table 5.7. The lower limit of parameter *n* are two hundred tasks (n_-=200) and for this value, the lower limit of parameter *r* are the twenty resources (r_-=20) while the upper limit are the thirty resources (r_+=30) .The upper limit of parameter *n* are the three hundred tasks (n_+=300) and for this value the limits of parameter *r* happen to be the same as before, the lower limit are the twenty resources (r_-=20) and the upper limit are the thirty resources (r_+=30). Table 5.8 presents the tests that are generated for the second part of Scenario A and their results.

n		**r**	
n_-	200	r_-	20
		r_+	30
n_+	300	r_-	20
		r_+	30

Table 5.6. Scenario A: Limits of *n,r* parameters

	n	**r**	**n_d**	
			15	20
Test7	200	20	0	0
Test8	200	30	0	0
Test9	300	20	0	0
Test10	300	30	0	0

Table 5.7. Scenario A (Part 2): Tests and results

As it is shown in Table 5.8, the bpo^F did not generate results for the other process sizes that were investigated. The next section presents the problem tests that are generated in Scenario B.

5.3 Scenario B

The Scenario B includes tests for business process designs that have a complex form. The complexity is due to the branches that are formed by the value of the r_{in}, r_{out} parameters that equals to five and the value of t_{in} , t_{out} parameters that equals to three. Figure 5.4 illustrates those points that should be taken into consideration for this design.

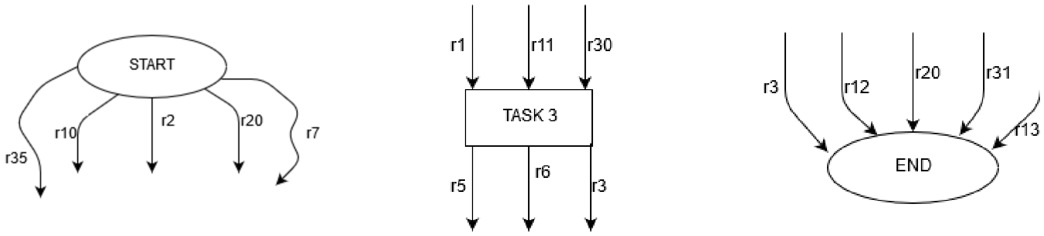


Figure 5.4. Process requirements for Scenario B

Similarly to Scenario A, the tests are divided into two parts. The procedure for the first part is exactly the same as it was in Scenario A: the n_d parameter, remains constant in a specified value and the *n*, *r* parameters begin to vary from specified values in a scalable way up until to finding value parameters for which the bpo^F does not generate results. In the second part there is a change in the procedure. Particularly by employing the lower and the upper limits of *n* and *r* parameters that were determined in the first part, the n_d parameter begins to vary from a specific value in a scalable way up to the point that it finds a value parameter for which the bpo^F does not generate results. This procedure takes place twice: the first time from a specific value downward in order to determine the lowest value that n_d can have and the second time from another specific value upward in order to determine the highest value that n_d can have. Table 5.9 and Table 5.10 summarize this procedure for the respective parts.

n_d=10		**r**		
		10	15	...
n	30	$S_{n=30,r=10}$	$S_{n=30,r=15}$	...
	100	$S_{n=100,r=10}$	$S_{n=100,r=15}$	...
	125	$S_{n=125,r=10}$	$S_{n=125,r=15}$	...
	.	.	.	...
	.	.	.	...
	.	.	.	...

Table 5.8. First part of Scenario B

		n			
		n-		*n+*	
		r		r	
		r-	*r+*	*r-*	*r+*
n_d	9	$S_{nd=9,n=n-,r=r-}$	$S_{nd=9,n=n-,r=r+}$	$S_{nd=9,n=n+,r=r-}$	$S_{nd=9,n=n+,r=r+}$
	8	$S_{nd=8,n=n-,r=r-}$	$S_{nd=8,n=n-,r=r+}$	$S_{nd=8,n=n+,r=r-}$	$S_{nd=8,n=n+,r=r+}$
	.	.	.	.	.
	.	.	.	.	.
	11	$S_{nd=11,n=n-,r=r-}$	$S_{nd=11,n=n-,r=r+}$	$S_{nd=11,n=n+,r=r-}$	$S_{nd=11,n=n+,r=r+}$
	12	$S_{nd=12,n=n-,r=r-}$	$S_{nd=12,n=n-,r=r+}$	$S_{nd=12,n=n+,r=r-}$	$S_{nd=12,n=n+,r=r+}$
	.	.	.	.	.
	.	.	.	.	.

Table 5.9. Second part of Scenario B

First part of the tests

In the first part the n_d parameter is selected to maintain the value ten (n_d=10). This process size can be considered as a required size for this type of designs like the ones that are being investigated in this scenario. The starting point of the investigation for the *n* parameter is the thirty tasks (n=30). This value is selected as the minimum number of available tasks to exist in the library for the generation of this type of process design. The starting point of the investigation for the *r* parameter is the ten resources (r=10). This value can be considered as a minimum number of available resources that the library can have in order for the bpo^F to generate optimized designs. Table 5.11 presents the tests that are generated for the first part of the Scenario B and their results.

	n_d=10		**r**								
			10	15	20	25	30	35	40	50	60
Test1	**n**	30	0	10	-	3	0	-	-	-	-
Test2		100	0	9	-	10	8	3	3	0	-
Test3		125	-	0	8	-	5	3	3	0	-
Test4		150	-	0	0	3	4	7	2	1	0
Test5		175	-	-	6	-	5	4	3	0	-
Test6		200	-	-	0	0	7	0	-	-	-

Table 5.10. Scenario B (Part 1): Tests and results

As it seems in Table 5.11, a test was terminated when the bpo^F did not generate solutions for a value of *r* parameter. In addition, if a starting value of *r* parameter did not generate solutions for two in a row values of n parameter, then the next test begun from the next *r* parameter value. Finally, the tests stopped at n=200, a library with two hundred tasks. Considering real life business processes and the requirements of the design that are being investigated in this scenario (r_{in}, r_{out}, t_{in} , t_{out} parameters), this number of library tasks can be characterized as large. From the results that are presented in the Table 5.11, we conclude that the bpo^F generates reliable results for all the *n* parameter values. Regarding the *r* parameter, the findings are:

- For n=30 the limits of *r* parameter are: r=[15,25]
- For n=100 the limits of *r* parameter are: r=[15,40]
- For n=125 the limits of *r* parameter are: r=[20,40]
- For n=150 the limits of *r* parameter are: r=[25,35]
- For n=175 the limits of *r* parameter are: r=[20,40]
- For n=200 the limits of *r* parameter are: r=[30]

As it was mentioned in previous section, reliable results are considered to be three or more solutions generated by the bpo^F.

Second Part of the tests

In the second part, we select two starting points in order to investigate the lower and the upper value that n_d parameter can have and we employ the limits of the *n,r* parameters that are extracted from the first part of the tests. The space between the two selected starting points should include the process size with ten tasks (n_d=10), the required size for this type of designs like the ones that are being investigated in this scenario. Thus, for the investigation of the lower n_d limit the starting point is process sizes with nine tasks (n_d =9) while for the upper limit the starting point is process sizes with eleven tasks (n_d =11). The extracted limits of *n,r* parameters are shown in Table 5.12. The lower limit of parameter *n* are thirty tasks (n_-=30) and for this value, the lower limit of parameter *r* are the fifteen resources (r_-=15) while the upper limit are the twenty five resources (r_+=25) .The upper limit of parameter *n* are the two hundred tasks (n_+=200) and for this value there is only one value of the parameter r that generates results, the thirty resources (r=30). Table 5.13 and Table 5.14 present the tests that are generated for the second part of Scenario B and their results.

n		**r**	
n_-	30	r_-	15
		r_+	25
n_+	200	r	30

Table 5.11. Scenario B: Limits of *n,r* parameters

	n	r	n_d								
			9	8	7	6	5	4	11	12	13
Test7	30	15	12	7	8	10	6	2			
Test8	30	25	1	2	2	0	-	-			
Test9	30	15							6	5	0
Test10	30	25							0	-	-

Table 5.12. Scenario B (Part 2): Tests and results (i)

	n	r	n_d													
			9	8	7	6	5	4	11	12	15	20	25	30	35	40
Test11	200	30	5	5	4	4	5	2								
Test12	200	30							4	6	6	3	5	2	1	1

Table 5.13. Scenario B (Part 2): Tests and results (ii)

As it was mentioned in the start of this section, a test was terminated when the bpo^F did not generate solutions for a value of n_d parameter. However, in the tests 7 & 11 the bpo^F generated solutions for all the n_d parameter values but the tests were terminated in the n_d value equal to four. This happened because a process size with four tasks (n_d=4) can be considered as the minimum size for this type of designs like the ones that are being investigated in this scenario. Also, the Test 10 was terminated in the n_d value equal to forty, because for three n_d values in a row the bpo^F did not generate reliable results. From the results that are presented above, we conclude that the n_d parameter values for which the bpo^F generates reliable results are:

- For n=30 and r=15, values of n_d parameter in the space n_d=[5,12]
- For n=200 and r=30, values of n_d parameter in the space n_d=[5,25]

5.4 Main remarks / Summary

This section summarizes the chapter and highlights the main findings. The chapter introduced a series of scalable business process tests in order to investigate the parameters of the business process optimization problem and determine their limits. Table 5.15 presents the parameter values that were investigated during the tests.

Parameter	Value	
n	50-300	30-200
r	10-60	10-60
n_d	10-20	4-40
t_{in}	1	3
t_{out}	1	3
r_{in}	1	5
r_{out}	1	5

Table 5.14. Parameter values during tests

The investigation is divided in two scenarios which were based on the form of the process designs. The form is expressed by the parameters group r_{in}/r_{out} & t_{in}/t_{out}. Thus, two values of these parameters were investigated. For each one of the scenarios, a series of scalable tests for the *n,r,* n_d parameters followed. The results of the tests proved that the bpo^F can generate solutions for both forms of process designs that were investigated. However, not all the produced results can be considered reliable, since in many cases the generated solutions were one or two. The *n,r* parameter values for which the bpo^F generates at least three solutions are shown in the Table 5.16.

	Parameter		
	n	*r*	n_d
Scenario A $\left\{\frac{r_{in} = r_{out} = 1}{t_{in} = t_{out} = 1}\right\}$	200	20-30	10
	300	20-30	
Scenario B $\left\{\frac{r_{in} = r_{out} = 5}{t_{in} = t_{out} = 3}\right\}$	30	15-25	10
	100	15-40	
	125	20-40	
	150	25-35	
	175	20-40	
	200	30	

Table 5.15. *n,r* parameter values that generate reliable results

The n_d parameter investigation confined in the lower and in the upper value of *n,r* parameters that are shown in the table above, as the full investigation was too hard to be done. The n_d parameter values for which the bpo^F generates at least three solutions are shown in the Table 5.17.

	Parameter		
	n	*r*	n_d
Scenario A $\left\{\frac{r_{in} = r_{out} = 1}{t_{in} = t_{out} = 1}\right\}$	200	20	10
		30	10
	300	20	10
		30	10
Scenario B $\left\{\frac{r_{in} = r_{out} = 5}{t_{in} = t_{out} = 3}\right\}$	30	15	5-12
	30	25	10
	200	30	5-25

Table 5.16. n_d parameter values that generate reliable results

To sum up, the main finding that emerged from the tests is that the bpo^F can generate much more reliable results for the various *n, r,* n_d parameter values for the form of process designs that are expressed by the parameters group r_{in}/r_{out} & t_{in}/t_{out} , at the values $r_{in}/r_{out}=5$ & $t_{in}/t_{out}=3$ (Scenario B) than the form of process designs that are expressed by this parameters group at the values $r_{in}/r_{out}=1$ & $t_{in}/t_{out}=1$ (Scenario A). Finally, the parameter limits that occur from the reliable results of the tests are summarized in the Table 5.18. The next chapter presents the application of Design of Experiments into the business process optimization problem.

Parameter	**Value**			
n	200	300	30	200
r	20-30	20-30	15-25	30
n_d	10	10	5-12	5-25
t_{in}	1	1	3	3
t_{out}	1	1	3	3
r_{in}	1	1	5	5
r_{out}	1	1	5	5

Table 5.17. Business process problem parameter limits

Chapter 6 – Parameter characterization using DOE

This chapter presents the application of the statistical method Design of Experiments (DoE) into the business process optimization problem. Considering the results of the tests performed in Chapter 5, this chapter moves the investigation of the business process optimization problem parameters a step further. Based on the DoE, as it was described in Chapter 4, this chapter seeks to discover the parameters that have a significant influence on the results that the business process optimization framework (bpo^F) generates. This procedure is analyzed step by step following a guideline for the DoE and important conclusions for the business optimization problem parameters are extracted.

6.1 Introduction

This chapter aims at characterizing the business process optimization problem parameters using Design of Experiments. As it was mentioned in Chapter 4, DoE can be considered as a method to determine the relationship between factors affecting a problem and the output of that problem. By the term characterization we mean the determination of the factors that have the most influence on the results. The subject of this dissertation, the business process optimization problem, is a problem that includes many factors, which are the parameters. The finding of the parameters that most influence the results, that is the solutions that are generated by bpo^F ,has not been investigated before.

In the previous chapter, an effort was made to investigate the business process problem parameters in detail and to determine the parameters limits in which the bpo^F generate reliable results. A series of scalable business process test problems was generated based on the experimental approach which Vergidis (2008) employed for the bpo^F performance evaluation and which was presented in Chapter 3. Table 6.1 summarizes the parameter limits that were extracted from the tests. Considering these results and based on the Vergidis (2008) experimental approach, the investigation goes on and moves a step further to the characterization of the business process optimization problem parameters using DoE. The expectations of employing DoE for the business process optimization problem are:

To determine

- the effects of the parameters
  (i.e. the change in the number of solutions that the bpo^F generates when a parameter changes from a value to another)

- the effect magnitude
- how likely these effects are to be important
- a potential interaction between the parameters
  (i.e. the change in the number of solutions that the bpo^F generates when a parameter changes from a value to another is to depend on the value of another parameter).

Parameter	**Value**			
n	200	300	30	200
r	20-30	20-30	15-25	30
n_d	10	10	5-12	5-25
t_{in}	1	1	3	3
t_{out}	1	1	3	3
r_{in}	1	1	5	5
r_{out}	1	1	5	5

Table 6.1. Parameter limits that occurred from the tests in Chapter 5

This chapter is based on a guideline for the Design of Experiments that was introduced in the Chapter 4 (section 4.3). The main steps of this guideline are the following:

1. Recognition of the problem for which DoE is used
2. Selection of the response variable
3. Choice of factors and levels
4. Choice of experimental design
5. Performing the experiment
6. Statistical analysis of the data
7. Conclusions and recommendations

While the first two steps are obvious since the reason for using DoE is the factor characterization and the response variable is the solutions that are generated by bpo^F, the next steps have to be elaborated. Hence, the next section engages steps 3-6 and the last section step 7. Finally, some terms of DoE which were introduced in Chapter 4 and are going to be greatly used in this chapter have to be clarified:

- Factor is a parameter of the problem (i.e. n, r, n_d)
- Level is a value that a parameter can take (n= 30, n=100, etc.)
- Replication is an independent repeat run of each factor combination

6.2 Application of DoE

This section applies the Design of Experiments (DoE) following the guideline that presented in section 4.3 in order to characterize the factors of business process optimization problem. As the first two steps have been clarified in the previous section, the next steps that have to be taken, are elaborated below.

Factor	Description
n	Number of tasks in the library
r	No. of available resources
n_d	No. of tasks in the design
n_{min}	Minimum number of tasks in the design
t_{in}	No. of task input resources
t_{out}	No. of task output resources
r_{in}	No. of process input resources
r_{out}	No. of process output resources
p	No. of task/process attributes

Table 6.2. Main business process optimization problem factors

Choice of factors and levels

In this step we have to select which factors we are going to study and their levels. As it was described in section 4.3 there are two main categories of factors: (i) the *Design factors* which are factors that are selected for study, and (ii) the *Held-constant factors* which are factors that remain constant at a specific level and do not participate in the procedure.

In order to categorize the factors of the business optimization problem we have to go back to the tests that were conducted in the previous chapter. A main finding that emerged from the tests is that the bpo^F can generate much more reliable results for the various *n, r,* n_d factor levels for the types of process designs that are expressed by the factors group r_{in}/r_{out} & t_{in}/t_{out} , at the levels $r_{in}/r_{out}=5$ & $t_{in}/t_{out}=3$ than the types of process designs that are expressed by this factors group at the levels $r_{in}/r_{out}=1$ & $t_{in}/t_{out}=1$. Since the levels of this factors group that were investigated did not produce a comparable number of reliable results, during the DoE procedure it is preferable to use only those levels of this factors group in which bpo^F performed better.

Thus, this factors group (r_{in}, r_{out}, t_{in}, t_{out}) is going to be in the *Held-constant factors* category at the levels $r_{in}/r_{out}=5$ & $t_{in}/t_{out}=3$ and is not going to participate in the factor characterization. In addition, the factors that were not investigated during the tests, n_{min} and *p* (task/process attributes) are going to be in the same category, with the values of levels maintained constant as follows:

- $n_{min} = n_d$
  We are not interested in solutions that have fewer tasks than the design requirement n_d.
- p=2
  The task/process attributes will be ‘A’ and ‘B’ and the spaces that vary are as follows:

$$A = [100,115] \qquad\qquad B = [200,230]$$

The rest of the factors, *n ,r,* n_d, are going to be the subject of the study, so they belong in the *Design factors* category. Table 6.3 presents the categorization of the factors.

Category	Factor
Design factors	n , r , n_d
Held-constant factors	r_{in}, r_{out}, t_{in}, t_{out} , n_{min}, p

Table 6.3. Categories of Factors

In order to choose the levels of *Design factors* category we have to consider the reason for employing DoE and the results of the tests. According to Montgomery (2017) if the reason of employing DoE is the factor characterization, the number of levels should be kept low. Thus, two levels for each factor are considered to be enough. Regarding the results of the tests we have to select factor levels for which the bpo^F generated reliable results. Moreover, we need to make sure that the selected levels of the *r,* n_d factors have generated results into the respective selected levels of the *n* factor. Based on these, Table 6.4 shows the selected levels of the factors.

Factor	Level	
	(-)	(+)
n	30	100
r	15	25
n_d	8	10

Table 6.4. Factor levels

In the Table 6.4 the low level of factors is symbolized with the minus sign ‘-‘ while the high level is symbolized with the plus sign ‘+’. As it seems in this table, the low level of factor *n* is the thirty tasks while the high level is the one hundred tasks, the low level of factor *r* is the fifteen resources while the high level is the twenty five resources and the low level of factor n_d is process sizes with ten tasks while the high level is process sizes with ten tasks.

An issue that occurs with these selected levels has to do with the high level of the *n* factor. For the reasons that were explained in the previous chapter, the tests for the investigation of the n_d factor confined in the lower and in the upper level of *n,r* factors. In order to employ the level of *n* factor equal to one hundred tasks, we have to confirm that the bpo^F generates reliable results for the selected levels of n_d factor at both levels of *r* factor. Table 6.5 presents the results of the supplementary tests. As it seems in this table, bpo^F generates reliable results for these factors levels and as a result the selected levels can be employed.

n	r	n_d	
		8	10
100	15	8	7
100	25	9	7

Table 6.5. Test results for the high *n* factor level

Choice of experimental design

Considering the Design factors and their levels that were selected in the previous step, in this step we have to choose the appropriate design and other things that are related with this, like the number of replications which indicates the sample size and if there will be any restriction as blocking. In our case the number of Design factors are three and the number of their levels are two, so the appropriate design is a **2^3 factorial design**. As it was described in Chapter 4, the 2^k designs can be considered as subsets of Factorial designs and are particularly useful because they provide the smallest number of runs with which *k* factors can be studied. Moreover for the 2^k designs, the calculation of the magnitude and the direction of factor effects and as well as their interactions, is feasible. Table 6.6 shows the combinations of levels factor for which we have to take measurements and under the DoE terminology it is called 'Design Matrix'. In this table the low level of factors is symbolized with the minus sign '-' while the high level is symbolized with the plus sign '+'.

	Factor		
Run	*A*	*B*	*C*
1	-	-	-
2	+	-	-
3	-	+	-
4	+	+	-
5	-	-	+
6	+	-	+
7	-	+	+
8	+	+	+

Table 6.6. Design Matrix

Finally, in our study the number of replications is going to be four, a number considered enough in order to have a good estimation of the factor effects. Also, there is not going to be any restrictions.

Performing the experiment

In this step we have to collect the data according to the requirements formed by the previous steps. During our study the experiment accounts for the bpo^F execution and the data that have to be collected or the measurements that need to be recorded account for the solutions that are generated by the bpo^F. As it was mentioned in previous chapters in this dissertation, we utilize only the NSGA2 from the EMOAs. The parameters of the NSGA2 are going to be constant during the procedure of DoE and are defined as follows:

Parameter	**NSGA2**
Population	500
Generations	25,000
Crossover prob.	0.8
Mutation prob.	0.2

Table 6.7 presents the measurements that were conducted in order to study and characterize the factors of business process optimization problem.

	Factor			**Solutions**			
Run	*n*	*r*	n_d	*Replicate1*	*Replicate2*	*Replicate3*	*Replicate4*
1	30	15	8	8	6	8	7
2	100	15	8	8	11	7	11
3	30	25	8	2	2	1	1
4	100	25	8	9	7	10	7
5	30	15	10	8	9	11	10
6	100	15	10	7	10	9	8
7	30	25	10	2	2	1	1
8	100	25	10	10	10	5	7

Table 6.7. Sample of bpo solutions

In the table above each run expresses an execution of bpo^F in which the factors of the business process problem are in the displaying levels. In addition, each run is repeated four times or according to the DoE terminology four replications are performed and the solutions that were generated by bpo^F are shown in the respective cells. The next part of the DoE procedure is the statistical analysis and it is presented below.

Statistical analysis of the data

This is considered to be the most important step in the DoE procedure. In this step we have to apply the statistical methods that were described in Chapter 4 in order to analyze the data and as a result to characterize the factors. The statistical analysis relies on the Minitab statistical software. Minitab provides a worksheet in which the data that was collected in the previous step are inserted. In addition, it randomizes the order of the runs in order to ensure the randomization, the most important principle in any experimental design. Table 6.8 shows how Minitab randomized the order of the measurements that were conducted in the previous step.

Run	**n**	**r**	**n_d**	**Solutions**
1	100	25	8	9
2	100	15	10	7
3	30	15	8	8
4	30	25	8	2
5	100	25	8	7
6	30	25	8	2
7	30	25	8	1
8	100	15	10	10
9	30	15	8	6
10	30	25	10	2
11	100	15	8	8
12	100	25	8	10
13	100	25	10	10
14	100	25	10	10
15	100	25	10	5
16	100	15	10	9
17	100	25	10	7
18	30	25	8	1
19	30	25	10	1
20	100	15	8	11
21	30	15	10	8
22	100	15	8	7
23	100	25	8	7
24	100	15	8	11
25	30	15	10	9
26	30	25	10	1
27	30	15	8	8
28	100	15	10	8
29	30	15	8	7
30	30	15	10	11
31	30	25	10	2
32	30	15	10	10

Table 6.8. The bpo sample in Minitab

As it was described in Chapter 4, the statistical analysis tests a hypothesis through the analysis of variance (ANOVA). The testing of the hypothesis in our study is the following

$$Ho: \beta_1 = \beta_2 = \beta_3 = \beta_{12} = \beta_{13} = \beta_{23} = \beta_{123} = 0$$

$$H1: \text{at least one } \beta \neq 0$$

where β is the effect between the two levels while the indicator shows the factor or the interaction between the factors. In order to test this hypothesis and estimate the factor effects we have to generate the 2^3 factorial design from the Minitab menu. The estimated factors effects along with their P-values are shown in the Table 6.9 and Table 6.10 presents the ANOVA. The significance level is defined to be 5% (a= 0.05) so a P-value of less than 0.05 (<0.05) implies significance.

Term	Effect	P-Value
n	3,562	0,000
r	-3,813	0,000
n_d	0,312	0,555
n*r	3,062	0,000
n*n_d	-0,813	0,132
r*n_d	-0,438	0,410
n*r*n_d	0,688	0,200

Table 6.9. Factors effects

Source	DF	SS	MS	F-Value	P-Value
Model	7	304,219	43,460	19,96	0,000
Linear	3	218,594	72,865	33,47	0,000
n	1	101,531	101,531	46,64	0,000
r	1	116,281	116,281	53,41	0,000
n_d	1	0,781	0,781	0,36	0,555
2-Way Interactions	3	81,844	27,281	12,53	0,000
n*r	1	75,031	75,031	34,46	0,000
n*n_d	1	5,281	5,281	2,43	0,132
r*n_d	1	1,531	1,531	0,70	0,410
3-Way Interactions	1	3,781	3,781	1,74	0,200
n*r*n_d	1	3,781	3,781	1,74	0,200
Error	24	52,250	2,177		
Total	31	356,469			

Table 6.10. ANOVA table

In Table 6.9 the sign ‘-‘ indicates that the number of solutions is reduced while the factor changes from the low level to the high level. The terms with the sign ‘*’ express the interactions of the factors. In Table 6.10 the DF is the degrees of freedom, the SS is the sum of squares, the MS is the mean squares, the F-Value is the value of F- test and the model is the ANOVA model which includes all the main (linear) effects and the interactions. These terms were detailed in Chapter 4. Finally, the ***P-value cannot be equal to zero***, but Minitab expresses the very small values (< 0,0001) in this way.

As it seems in both tables there are three effects that can be characterized as highly significant due to the fact that their P-values are very small. These effects are:

i. the effect of *r* factor
ii. the effect of *n* factor
iii. the interaction between *n* and *r* factors.

Figure 6.1 illustrates the magnitude and the importance of the effects employing a Pareto chart and Figure 6.2 illustrates the influence of each effect on the mean number of solutions where the significance of these three effects is distinguished by the way that the number of solutions varies. The null hypothesis can be rejected considering the significance of these three effects or considering the P-Value of the model which is very small.

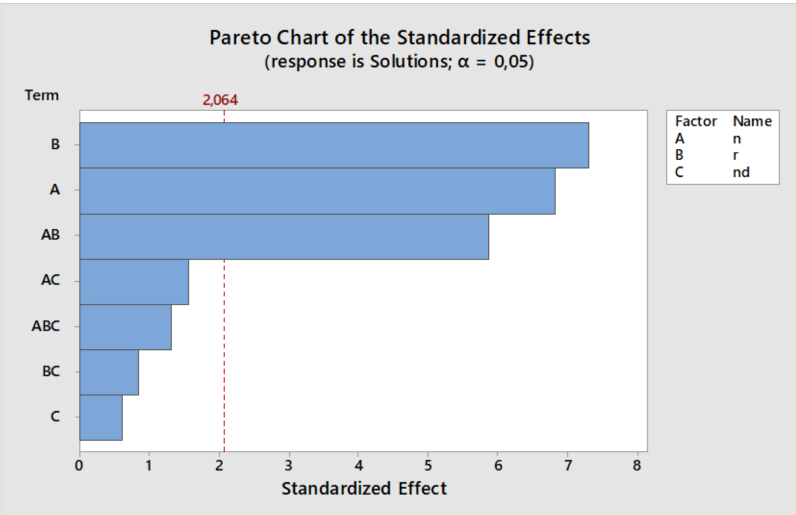


Figure 6.1. Pareto chart of the effects

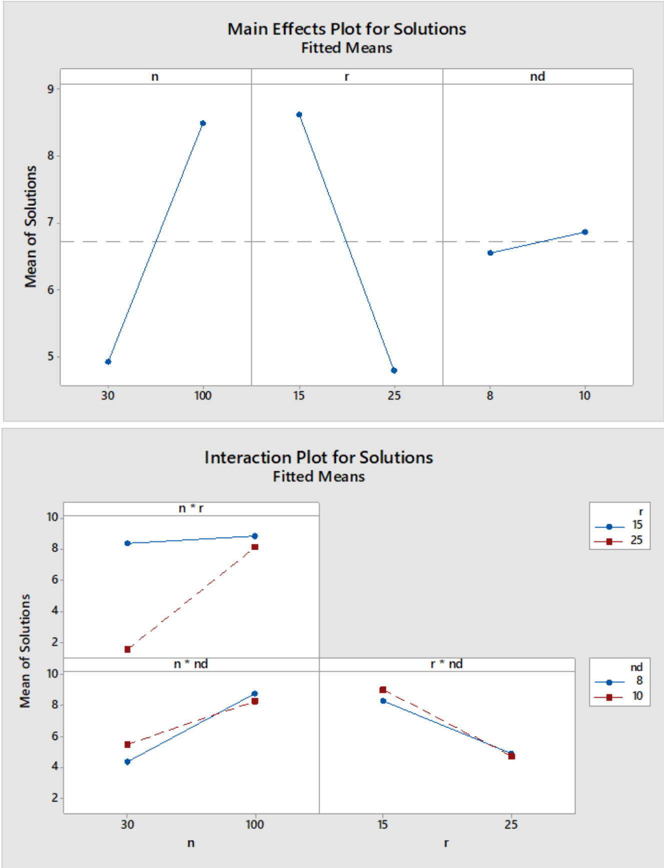


Figure 6.2. Mean number of solution and effects

Based on the ANOVA, the variability in the data can be explained by the quantity R-squared that was calculated by the Minitab:

$$R^2 = \frac{SS_{Model}}{SS_T} = 0{,}8534$$

Thus, the R-squared indicates that the factors *n, r,* n_d and their interactions explain by up to 85,34 % the variability in the number of solutions that the bpo^F generated. In addition, Minitab produced a regression equation in order to describe the relationship between the response and the terms in the model. This regression equation is:

Solutions = -16,6 + 0,334 n + 0,593 r + 4,34 n_d - 0,0089 n*r - 0,0509 n*n_d - 0,171 r*n_d + + 0,00196 n*r*n_d

Equation 6.1

This step and as a result this section concludes with two residuals plots in order to estimate how adequate the regression and the ANOVA models are. Figure 6.3 shows the normal probability plot and the plot of residuals versus fitted values of the model. The normal probability plot verifies that the residuals are normally distributed as they approximately follow the straight line and the plot of residuals versus fitted values shows that the residuals are structureless, i.e. without any pattern, so the model can be considered adequate.

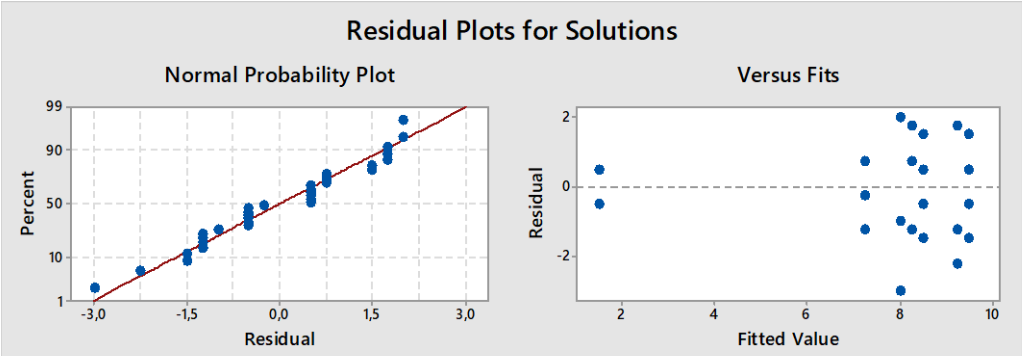


Figure 6.3. Residual plots

6.3 Main remarks / Summary

This section highlights the main findings and summarizes the chapter. The chapter introduced the Design of Experiments (DoE) method for the business process optimization problem. In order to characterize the business process optimization problem factors (parameters) a guideline for the DoE was followed step by step. In the beginning, the factors and their levels as well as the appropriate design resulted after considering the reliable results of the tests from Chapter 5. These are presented in Table 6.11.

Design	2^3 factorial		
Factors	n	r	n_d
Levels	$n_- = 30$ $n_+ = 10$	$r_- = 15$ $r_+ = 25$	$n_{d-} = 8$ $n_{d+} = 10$

Table 6.11. Factors, levels & design

Next the bpo^F was executed for the above factor levels and the measurements were inserted into Minitab in order to perform the statistical analysis for the 2^3 factorial design. Figure 6.4 illustrates a cube plot that is produced by the Minitab in order to present the mean of the measurements (mean number of solutions that the bpo^F generated) for each level combination of the factors. The statistical analysis that was generated by the Minitab shows that there are three effects that are statistically significant for these levels of the factors. Table 6.12 summarizes these effects and their magnitudes. In addition, Minitab calculated that the R-squared, the percentage of variation in the results that is explained by the factors, is 85 % and generated the regression model (Equation 6.1).

Factor	Effect
r	-3,813
n	3,562
Interaction between n & r	3,062

Table 6.12. Significant effects

To sum up, we conclude that the expectations posed in the beginning of the chapter about the employment of DoE in the business optimization problem have been fulfilled. The conclusions that resulted from the study presented in this chapter about the parameter spaces selected based on the reliable results of the investigation that took place before, are the following:

- The change in the value of *n* , *r* parameters has a significant influence on the number of solutions that the bpo^F generates.
- The *r* parameter has the biggest influence.
- The parameters *n* , *r* , n_d describe by up to 85% the variability in the number of solutions that the bpo^F generates.

The next chapter provides an overview of this dissertation and a critical discussion on the limitations, the contribution and the potential for future research.

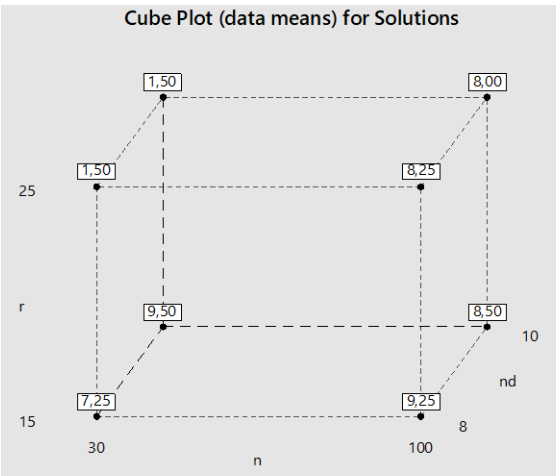


Figure 6.4. Mean number of solutions for each level combination

Chapter 7 – Conclusion

This chapter summarizes this research, discusses the contribution and the limitations of this research and provides suggestions for future work.

7.1 Thesis overview

The aim of this thesis as stated in Chapter 1, is the systematic investigation of the business process optimization problem parameters as they were introduced at the business process optimization framework (bpo^F) (Vergidis,2008).

Vergidis (2008) proposed an approach for the Evolutionary Multi-objective business process optimization, the business process optimization framework (bpo^F). The bpo^F utilizes as a main component the proposed business process representation and EMOAs in order to generate alternative optimized designs. The business process representation is described by mathematical parameters and the composition of a business process design is based on an algorithm that is named Process Composition Algorithm (PCA). During the execution of the bpo^F, the PCA composes feasible business process designs based on the parameters as well as the EMOAs use them in order to optimize the business process designs. Hence, the parameters are very important to the business process optimization problem on the grounds that they affect the generated results of bpo^F.

In order to understand the business process optimization framework (bpo^F), basic concepts about the business process optimization had to be understood. For this reason, the first objective of this research was to study and understand the business process optimization. This objective was carried out in Chapters 2 & 3. In Chapter 2 an overview of the definitions of business processes, the business process modeling and evolutionary approaches was provided. In Chapter 3 the business process optimization framework (bpo^F) was analyzed in-depth. The second objective was the review of the results that Vergidis (2008) reported. The validation of these results would indicate the accuracy and the reliability of Vergidis' work. This objective was presented in Chapter 3.

The third objective was to generate scalable business process test problems. From the study of the bpo^F the need for a further and more precise investigation of the problem parameters occurred. For this reason, a series of scalable business process tests was employed. Based on these tests the next objective was to determine the limits of the problem parameters for which the bpo^F generates reliable results. These two objectives took place in Chapter 5. In order to investigate the majority of the parameters and determine the parameter limits the tests were divided in two scenarios based on the group of parameters that express a process design. The rest of parameters varied in a scalable way and were evaluated based on the effectiveness of bpo^F.

The final objective of this thesis was to characterize the significance of the problem parameters and determine their influence on the results that are generated by the bpo^F. In order to do this, the statistical approach DoE was employed. DoE is a statistical method which can be used for the investigation and the determination of the relationship between the factors affecting a process and the output of that process. In Chapter 4 the main concepts of DoE were presented while Chapter 6 presented the application of DoE into business process optimization problem.

7.2 Research contribution

This research contributes to the business process optimization framework (bpo^F) that was proposed by Vergidis (2008) providing a complete and extended investigation of the business process optimization problem parameters. Through this investigation the following are determined:

- The parameter values for which the bpo^F generates reliable results and as a result the parameters limits.
- The parameters that have a significant influence on the results that the bpo^F generates.
- The parameter that has the biggest influence.
- The magnitudes of these influences.
- The proportion of variability on the results that the bpo^F generates, which is described by the n , r , n_d parameter group.

7.3 Research limitations

To begin with, there are two main limitations regarding this research. The first limitation refers to the way the problem parameters were investigated. Particularly, the parameters group r_{in}/r_{out} & t_{in}/t_{out} that expresses the form of process designs is confined only in the investigation of two cases. This limitation resulted from the difficulty to find parameter values for which the bpo^F will generate a satisfying number of solutions and which will also express different forms of process designs. The second limitation is about the parameter spaces employed during the DoE procedure. Taking into account the results of the tests, the Factorial Design needed the selected spaces for the r, n_d parameters to generate results into the respective selected space of the *n* parameter. This fact restricted the investigated parameter values into short spaces, as it was too difficult to find larger spaces from the tests results.

7.4 Future work

This research presented a complete and extended investigation of the business process optimization problem parameters. This investigation had two directions. The first one was a series of scalable tests for the business process optimization problem and the second direction was the application of the statistical approach DoE on this problem. A future research on the business process optimization problem parameters can rely on these directions.

More particularly, regarding the first direction, the scalable tests, all the possible combinations of the n, r, n_d parameters in the spaces of the n, r parameter examined in this research can be investigated. In addition, the investigation of n, r parameters values can be studied in larger spaces. As regards to the second direction, in a future research, the *Response Surface Design* could be applied in this problem. The *Response Surface Design* is mainly employed in order to find the factor settings that produce the "best" response. This would be translated in the business optimization problem as the values of parameter spaces for which the bpo^F generates the biggest number of optimized designs.

References

Abate, A.F., Esposito, A., Grieco, N. and Nota, G. (2002), 'Workflow Performance Evaluation Through WPQL', in *Proceeding of the 14th International Conference on Software Engineering and Knowledge Engineering Vol. 27*, ACM Press, New York, pp. 489-495.

Agerfalk, P., Goldkuhl, G. and Cronholm, S. (1999), 'Information Systems Actability Engineering - Integrating Analysis of Business Process and Usability Requirements', in *Proceedings of the 4th International Workshop on the Language Action Perspective on Communication Modelling* Copenhagen, Denmark, pp. 1245-1252.

Aguilar-Saven, R.S. (2004), 'Business Process Modelling: Review and Framework', *International Journal of Production Economics*, Vol. 90, pp. 129-149.

Biazzo, S. (2002), 'Process Mapping Techniques and Organizational Analysis', *Business Process Management Journal*, Vol. 8, No. 1, pp. 42-52.

Castellanos, M., Casati, F., Umeshwar, D. and Ming-Chien, S. (2004), 'A Comprehensive and Automated Approach to Intelligent Business Processes Execution Analysis', *Distributed and Parallel Databases*, Vol. 16, pp. 1-35.

Davenport, T.H. (1993), *Process Innovation: Reengineering Work Through Information Technology*, Harvard Business School Press, Boston.

Davenport, T.H. and Short, J.E. (1990), 'The New Industrial Engineering: Information Technology and Business Process Redesign', *Sloan Management Review*, pp. 11-27.

Fan, Y.S. (2001), *Fundamental of Workflow Management Technology*, Springer-Verlag, New York.

Georgoulakos, K., Vergidis, K., Tsakalidis, G., and Samaras, N. (2017), 'Evolutionary Multi-Objective Optimization of business process designs with pre-processing', *2017 IEEE Congress on Evolutionary Computation,* pp. 897-904.

Grigori, D., Casati, F., Castellanos, M., Dayal, U., Sayal, M. and Shan, M.-C. (2004), 'Business Process Intelligence', *Computers in Industry*, Vol. 53, pp. 321-343.

Guha, S., Kettinger, W.J. and Teng, J.T.C. (1993), 'Business Process Reengineering: Building a Comprehensive Methodology', *Information Systems Management*, Vol. 10, pp. 13-22.

Gunasekaran, A. and Kobu, B. (2002), 'Modelling and Analysis of Business Process Reengineering', *International Journal of Production Research*, Vol. 40, No. 11, pp. 2521-2546.

Hammer, M. and Champy, J. (1993), *Reengineering the Corporation: A Manifesto for Business Revolution*, N. Brealey, London.

Havey, M. (2005), *Essential Business Process Modelling*, O'Reilly, U.S.A.

Hofacker, I. and Vetschera, R. (2001), 'Algorithmical Approaches to Business Process Design', *Computers & Operations Research*, Vol. 28, pp. 1253-1275.

Irani, Z., Hlupic, V. and Giaglis, G.M. (2002), 'Business Process Reengineering: An Analysis Perspective', *International Journal of Flexible Manufacturing Systems*, Vol. 14, pp. 5-10.

Johanson, H.J., McHugh, P., Pendlebury, A.J. and Wheeler, W.A.I. (1993), *Business Process Reengineering - Breakpoint Strategies for Market Dominance*, Wiley, Chichester.

Kim, C.H., Weston, R.H., Hodgson, A. and Lee, K.H. (2003), 'The Complementary Use of IDEF and UML Modelling Approaches', *Computers in Industry*, Vol. 50, pp. 35-56.

Koubarakis, M. and Plexousakis, D. (2002), 'A Formal Framework for Business Process Modelling and Design', *Information Systems*, Vol. 27, pp. 299-319.

Lakin, R., Capon, N. and Botten, N. (1996), 'BPR Enabling Software for the Financial Services Industry', *Management Services*, Vol. 40, pp. 18-20.

Lindsay, A., Downs, D. and Lunn, K. (2003), 'Business Processes - Attempts to Find a Definition', *Information and Software Technology*, Vol. 45, pp. 1015-1019.

Luttighuis, P. O., Lankhorst, M., van de Wetering, R., Bal, R., and van den Berg, H. (2001), 'Visualizing Business Processes', *Computer Languages*, Vol. 27, pp. 39-59.

Melao, N. and Pidd, M. (2000), 'A Conceptual Framework for Understanding Business Process Modelling', *Information Systems*, Vol. 10, pp. 105-129.

Montgomery, D. C. (2017), *Design and analysis of experiments*, John wiley & sons, Arizona.

Pall, G.A. (1987), *Quality Process Management*, Prentice-Hall, Englewood Cliffs, NJ, USA.

Sadiq, W. and Orlowska, M. (2000), 'Analyzing Process Models Using Graph Reduction Techniques', *Information Systems*, Vol. 25, No. 2, pp. 117-134.

Shen, H., Wall, B., Zaremba, M., Chen, Y. and Browne, J. (2004), 'Integration of Business Modelling Methods for Enterprise Information System Analysis and User Requirements Gathering', *Computers in Industry*, Vol. 54, pp. 307-323.

Smith, H. (2003), 'Business Process Management - the Third Wave: Business Process Modelling Language and Its Pi-Calculus Format', *Information and Software Technology*, Vol. 45, pp. 1065-1069.

Soliman, F. (1998), 'Optimum Level of Process Mapping and Least Cost Business Process Re-Engineering', *International Journal of Operations and Production Management*, Vol. 18, No. 9/10, pp. 810-816.

Stock, J.R. and Lambert, D.M. (2001), *Strategic Logics Management*, McGraw-Hill, Irwin.

Stohr, E.A. and Zhao, J.L. (2001), 'Workflow Automation: Overview and Research Issues', *Information Systems Frontiers*, Vol. 3, No. 3, pp. 281-296.

Tinnila, M. (1995), 'Strategic Perspective to Business Process Redesign', *Business Process Re-Engineering and Management Journal*, Vol. 1, No. 1, pp. 44-59.

Tiwari, A. (2001), ‘Evolutionary computing techniques for handling variables interaction in engineering design optimisation’. PhD Thesis, SIMS, Cranfield University.

van der Aalst, W.M.P., ter Hofstede, A.H.M. and Weske, M. (2003), 'Business Process Management: A Survey', *Lecture Notes in Computer Science*, Vol. 2678, pp. 1-12.

Vergidis, K. (2008), 'Business Process Optimisation using an Evolutionary Multi-Objective Framework. ' PhD Thesis, Cranfield University.

Volkner, P. and Werners, B. (2000), 'A Decision Support System for Business Process Planning', *European Journal of Operational Research*, Vol. 125, pp. 633-647.

Wang, M. and Wang, H. (2005), 'Intelligent Agent Supported Business Process Management', in *Proceedings of the 38th Hawaii International Conference on System Sciences*, Hawaii, pp. 567-577.

Zakarian, A. (2001), 'Analysis of Process Models: A Fuzzy Logic Approach', *International Journal of Advanced Manufacturing Technology*, Vol. 17, pp. 444-452.