Optimization methods for inventive design

Lei Lin

UNIVERSITÉ DE STRASBOURG

ÉCOLE DOCTORALE ED 269

LGECO Laboratoire du GEnie de la COnception EA3938

THÈSE présentée par :

Lei LIN

soutenue le : **1 avril 2016**

pour obtenir le grade de : **Docteur de l'Université de Strasbourg**

Discipline/ Spécialité : Génie Industriel / Conception

Optimization methods for inventive design

(Méthodes d'optimisation pour la conception inventive)

DIRECTEUR DE THÈSE:

Monsieur DE GUIO Roland	Professeur, INSA de Strasbourg

RAPPORTEURS :

Monsieur CASTAGNA Pierre	Professeur, Université de Nantes
Monsieur LE LANN Jean-Marc	Professeur, INP-ENSIACET

EXAMINATEURS :

Monsieur CASCINI Gaetano	Professeur, Politecnico di Milano
Madame RASOVSKA Ivana	Maître de conférences, Université de Strasbourg

INVITÉS :

Monsieur BARTH Marc	Maître de conférences HDR, INSA de Strasbourg
Monsieur ROSE Bertrand	Maître de conférences HDR, Université de Strasbourg

Acknowledgements

This dissertation was enabled due to the help of many individuals from China and France during the last three years.

First of all, I would like to thank to the China Scholarship Council for the financial support during my PHD studies in the laboratory of design engineering (Laboratoire de GEnie de la COnception) at the National Institute of Science and Technology of Strasbourg (INSA de Strasbourg in french).

I would like to express my sincere gratitude to my supervisor Professor Roland De Guio, for his continuous support within my PHD studies and research, for his motivation, enthusiasm and immense knowledge. Thank you for your time, your patience, for the guidance in the area of the inventive problem solving (TRIZ) from both the theoretical and also from the practical point of view. All support you provided me during these three years made me grow and continue in my academic life. It was a great pleasure to learn from you in different domains, especially in the above mentioned TRIZ. I am fully confident that I will benefit from your help in the rest of my career.

I would also like to express my sincere gratitude to my co-supervisor Ivana Rasovska. Thanks for your fruitful discussions and suggestions you gave me when I was puzzled and lost in my thesis. In addition, I really appreciate your help in my life, I still remember when I first arrived to Strasbourg from China and you came to the airport to meet me and took me to my studio by car. As a foreigner student who nearly could not speak French in that time, were all things difficult for me. Thank you for all you have done for me during my stay in Strasbourg.

I would also like to express my appreciation to Sébastien Dubois. My task wouldn’t be accomplished without your help in TRIZ. I have greatly benefited from your explanations of the dialectical thinking. I would also like to thank to Dmitry Kucharavy, for your courses of TRIZ, which help me so much in discovering of TRIZ.

I would especially like to thank Dr. Zhang Qiang, Dr. Li Kai, Dr. He Huan, and Dr. Yan Wei who gave me a lot of support for my life in France. Thanks to Dr Thong Chai for sharing the living experience and idea with me. I would also like to thank my Chinese friends in Strasbourg, (Zhao Peng, Feng Xin, Zhao Enyu, Zang Dejin, Hao Xin, Li

Xiuming, Yu Wei, Hou Jing, Wang Chunliang, Ding Tianyu, Chen Jie, Sun Zhen, Zhang Zhongrui), I enjoyed the happy life with them in France.

Last, but not least, I would like to thank my parents for their unconditional support throughout my whole life.

Thank you very much all of you!

Lin lei

10th February 2016

Contents

Abstract

The thesis deals with problems of invention where solutions of optimization methods do not meet the objectives of problems to solve. The problems previously defined exploit for their resolution, a problem extending the model of classical TRIZ in a canonical form called "generalized system of contradictions." This research draws up a resolution process based on the loop simulation-optimization-invention using both solving methods of optimization and invention. More precisely, it models the extraction of generalized contractions from simulation data as combinatorial optimization problems and offers algorithms that provide all the solutions to these problems. In addition, it provides heuristics to select variables and its relevant values involved in generalized contradictions and/or useful for optimization. The contributions concern theory and practice of the inventive design. The thesis also explores cross-fertilization between optimization and TRIZ.

Keywords: TRIZ, inventive design, optimization.

List of Figures

List of Tables

List of Appendixes

Chapter 1 Introduction

1-1 Context

The works presented in this chapter are part of research efforts conducted over the last fifteen years in the LGECO laboratory within the context of applying the theory of inventive problem solving, also known as TRIZ (*Teorija Reshenija Izobretateliskih Zadatch*), and its extensions in different domains. TRIZ is a set of methods and tools organized into one system that was developed by Genrich Altshuller [1] in order to facilitate the invention of physical objects. The underlying principle behind the combination of methods that constitute TRIZ is based on a set of fundamental assumptions derived from dialectics and analysis concerning the evolution of technical systems. As a significant portion of innovative research carried in this field is of the increasing order, the research aims at improving different components of existing approaches. Some foundations of TRIZ, such as dialectical thinking, are very generic. Several studies have attempted an analogous approach of Altshuller's work for other application domains such as management, advertising, and logistics. If these research works had achieved some success, recurrent difficulties raised in implementation of the problem formulation through their underlying contradiction will be solved by the same approach. However, this was also our problem when the laboratory initiated a research work on the identification of contradictions that underlie simulator system limits by developing numerical models of system behavior. In order to accomplish this, the definition of the concept of contradiction, as defined by TRIZ, was too vague to be used with computer tools. The definition of the concept of contradiction could be the origin of difficulties for human users to define and understand contradictions. Because of these practical and theoretical reasons, the definition of contradiction has been revised, and a generalized contradiction model has been proposed. This new contradiction model is even more difficult to identify and understand by humans; however it is sufficiently precise to be processed by a computer as soon as we have enough data regarding the system behavior.

The first objective of this thesis is to propose the methodology, tools and algorithms to identify these contradictions from experimental data or system simulation data, for understanding system problems and searching for their solutions. The second objective is to use this new tool to analyze and explain certain practical difficulties that humans encountered when identifying contradiction. Finally, a research on formalization of generalized contradictions has reinforced our idea that there is a link between optimization theory and inventive problem solving through the concept of contradiction. Furthermore, the tools developed in this thesis aim at exploring the link between optimization theory and inventive problem resolution by using the concept of generalized contradictions and the Pareto line.

In order to demonstrate the generality of the approach, illustrative examples from the thesis are drawn from the design of physical objects (e.g., the case of an electrical circuit breaker) to the process design of internal logistics (e.g., the case of an inventory Kanban system). The following paragraphs of this introduction discuss the background of our research work, especially the concept of contradictions in TRIZ, and the general framework of the proposed inventive design problem solving process. The specific model used in our research is the generalized system of contradictions involving generalized technical and physical contradictions. We use the formalism of the experimental design as the common model permitting support of optimization and inventive methods used to illustrate, identify, and extract generalized contradictions. The challenges of our research are then presented through questions on three different levels:

- Design theory through contradictions as the foundation concept of the TRIZ theory.
- Practical consequences of a new contradiction definition for the inventive problem solving.
- Exploration of the relation between optimization methods and TRIZ in order to develop cross-fertilization from theoretical and practical points of view.

To answer these questions, a research method is proposed and questions are answered completely or partially through the utilization of the proposed algorithms.

Real examples are then used to illustrate the proposed algorithms. The last part of this chapter is dedicated to the thesis organization.

1-2 Background

1-2.1 Inventive design problem solving

Problem solving is a common activity for a number of domains, and its crucial role in design is particularly recognized in [2],[3]. In general, problem solving cannot be distinguished from problem formulation. A respectable formulation of a problem means approximating a solution. However, it is difficult to determine a well-formulated problem. This concept presumes that some problems are not well formulated or are not real problems. Thus, a question then arises as to what constitutes a real problem. The different types of answers to this question arise from heterogeneous ways for tackling the concept of a problem, its formulation, and thus, the way to manage its resolution process [4]. The creative or inventive design problems were identified as either ill-defined or ill-structured by Reitman in [5]. This definition means the start state of problem solving activities are not completely specified, the goal state could be changed or reformulated in time, and the transformation function is also completely unspecified. Moreover, Bonnardel in [2] considers design problems as being open-ended as they do not have a single solution but a set of possible solutions.

Thus, solution synthesis is a result of the choice of one solution from several solutions. Often very little information regarding a design problem exists that indicates problem solving requires a significant structuring of the problem itself [6]. Problem structuring is a process of drawing upon external information to compensate for missing information and using that information to construct the problem space [7]. The process of problem structuring begins with an interpretation of the problem situation–definition of problem parameters and functions. Then the generation of design requirements and constraints follows. These requirements and constraints are used to specify the design assignment (i.e., defining the problem space) as well as to describe and explore aspects of the desired solution (i.e.,

exploring the solution space). As problem resolution aims at developing a well-formulated problem, it is necessary to ensure the evolution of the first understanding of the problem as well as the first model of the problem. Our goal is then to propose a method in which an initial model of the problem could be changed in order to pursue its resolution.

The concept and model of problem are directly linked to the nature of the considered knowledge. Thus, in the domain of problem solving for technical system designs, it is important to clarify the type of knowledge and model relevant to the resolution process. The inventive problem is a design problem that has no solution based on the known conditions. Inventive design, which tries to solve the inventive problems, is a specific activity that differs from the traditional design performed in research and development departments [8]. An invention supposes to invent something, i.e., to propose something new or something not known until now. Inventive problems can concern any known field, technical or not, and specifically some principle, product or fact relevant to that field. Designing a new technical system means pursing the evolution of a technical system [9].

Two types of situations can lead to this evolution process: increasing the system efficiency by optimization of its parameters or redesigning the system when new parameters are introduced during the resolution process, e.g., when a working principle is changed. The hypothesis is that two types of problem solving techniques can be used to solve inventive problems: optimization methods to optimize the system parameters and inventive methods to change the problem model. The logical succession is that optimization techniques are used first because of the nature of the formalized process required for problem optimization. If no solution is found, the designer will resort to inventive solving techniques, which are much more creative but not well formalized yet. In this thesis, these problems, which cannot be solved by optimization methods and require more profound changes of the models, will be considered problems of invention.

Rosenman and Gero in [10] proposed a system of design classification that decomposes into three different subgroups based on different problem spaces in

which problem resolution is researched and solving methods are used to resolve the problem:

- **Routine design** proceeds within a well-defined state space, i.e. all design variables and their possible domains or ranges being known and the problem being one of possible instantiations. The tools used can include optimization methods.
- **Innovative design** proceeds within a space of known solutions which is extended by developing variations or adaptations to existing designs, i.e. the domain or ranges of values of existing design variables is extended. Over-constrained satisfaction methods could be used to solve this type of problem.
- **Creative design** implies the formulation of a state space that may include an extended state space of possible solutions or creating a new state space. For innovative and creative design, inventive resolution tools are used.

Based on the definition of creative design as a process of searching solutions beyond the known design space, many authors include the reformulation of design problems into the process of creative design. Different theories regarding the expansion of the problem space and the principles of determining solutions behind this space have been proposed in [11],[12],[13],[14]. On one hand, the theories have brought about changes in determining the problem structure as well as its reformulation. Additionally, different theories have brought about the expansion and evolution of the problem space based on used solving methods. Boden in [15] proposes to explore and transform a conceptual space that unifies and structures a domain of thinking. Exploration of this space identifies the limits and points where possible transformations allow expanding the search space such as removing or negating a constraint. Maher et al. in [13] proposed a model for design exploration based on a computational coevolution of the design and solution space by using a modified genetic algorithm. The problem space is defined as a fitness function (i.e., representation of functional requirements) and the solution space is defined as a set of design genes (i.e., design solutions characterized by parameters).

Two systems will evolve in response to each other, the features and constraints in the current solution become new criteria that redefine the problem space. This concept

is based on a genetic algorithm and is part of evolutionary design where the current state of the theory is presented in [13]. Hatchuel in [12] proposed a theory of expandable rationality where learning devices generate new problems and concepts that lead to unexpected expansions of initial concepts. He proposed a unified design theory where the design process is modeled as a co-expansion of two spaces: concepts and knowledge. Another interesting theory was proposed by Dorst in [11]: where the nature of design problems was studied within the framework of Simon's theory of ill-structured problems and their solving as well-structured problems. The design is based on the notions of paradox and discourse. A paradox is defined as a complex statement that consists of two or more conflicting statements. The model of paradox and discourse requires a redefinition of a problematic situation in order to determine a solution. Discourses are the elementary statements that comprise a paradox. This method describes the complete structure of the terms and relationships that lie at the basis of the thinking and discussions within an area of human activity.

This is quite similar to the contradictions within the TRIZ dialectical approach. Altshuller [16] proposed the theory of inventive thinking TRIZ, which tackles creative technical problem solving. For him, creativity was to find a way where compromise would not be recognized as a solution. The statement of the problem is accomplished by goals (i.e., what must be achieved) and means (i.e., how to achieved the goal and what must be done). The problem resolution process will lead to the acquisition of new knowledge, or at least represent the knowledge based on a new understanding, in order to shift from a problem perspective towards a solution focus. The solution is searched in the design space where possible variants of design problems and the solutions exist. The reason why we cannot solve the problem is that the requirements are in contradictions. Altshuller proposed an ideal machine with determined parameters as a prototype that indicates the most promising direction to search for a solution. Technical contradictions then indicate the obstacles that must be overcome to achieve the solution.

1-2.2 Optimization versus invention

The goal of our research work is to find a solution to design problems by browsing a design problem space and to use both optimization and inventive solving principles. This problem space is defined in [17] in terms of states of problem solving, where operators and evaluation functions move the problem solving from one state to another. We analyze the approaches used by different solving methods to explore the problem space, which operators are used for, and where an adequate solution to the design problem appears in the problem space. Two types of design problems are suggested. Once the problem has been modeled, the optimization design problem is searching for values for a fixed set of variables. This approach prompts the objectives to arrive at an optimal value without changing the model or non-creative design. The optimization methods have been proven to be effective in many situations, but not effective for inventive design problems that require improving a system by adding new variables or new relationships between variables. Optimization algorithms browse a space of potential solutions, which is limited by the stated problem space. If no solution is found, the classical optimization algorithms are not able to continue to explore the solution space. For this case, the inventive solving theory TRIZ proposes methods to change the stated problem model thereby defining a new problem space.

Our approach of model change is based on the best results developed from the optimization methods and represented by the Pareto front [18]. By considering an arbitrary optimization problem with k objectives, where all objectives are minimized and equally important, i.e., no additional knowledge about the problem is available. We can then assume that a solution to this problem can be described in terms of a decision vector in the decision space X. Subsequently, a function can be developed that evaluates the quality of a specific solution by assigning it an objective vector in the objective space Y (Figure1).

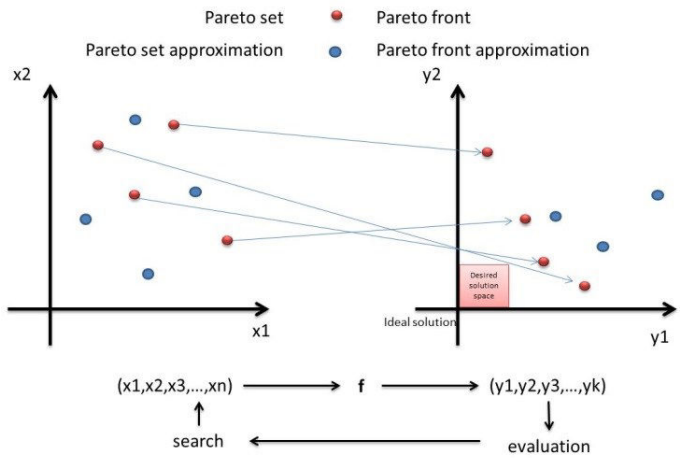


Figure1: Illustration of general multi-objective optimization problem

Following the well-known concept of Pareto dominance, an objective vector Y is said to dominate other objective vectors if all components of the considering vector are as good as the components of other objective vectors and at least one component of Y is better. Accordingly, we can say that one solution is better than another solution, i.e., x1 dominates x2, if f(x1) dominates f(x2). Here, optimal solutions, as defined by solutions not dominated by any other solution, may be mapped to different objective vectors. In other words, there may exist several optimal objective vectors representing different trade-offs between the objectives.

Our goal is to go beyond this limit represented by the Pareto front and obtain results from the desired objective space. For this purpose, we use the best solutions obtained by the optimization methods and issued from the Pareto front representing a conflict of performance. As an example (Figure 2), for solution 1, the evaluation parameter PE1 is better than that for solution 2; however, the evaluation parameter PE 2 is worse and vice versa. This conflict in the evaluation (objective) space is a limit expressed by technical contradiction. This conflict also represents an entry point to use the dialectical approach with the view of contradiction corresponding to this conflict as shown in Figure 2. The subsequent question is to determine how to translate this conflict through the search space system parameters, which correspond to the physical contradiction within the TRIZ theory.

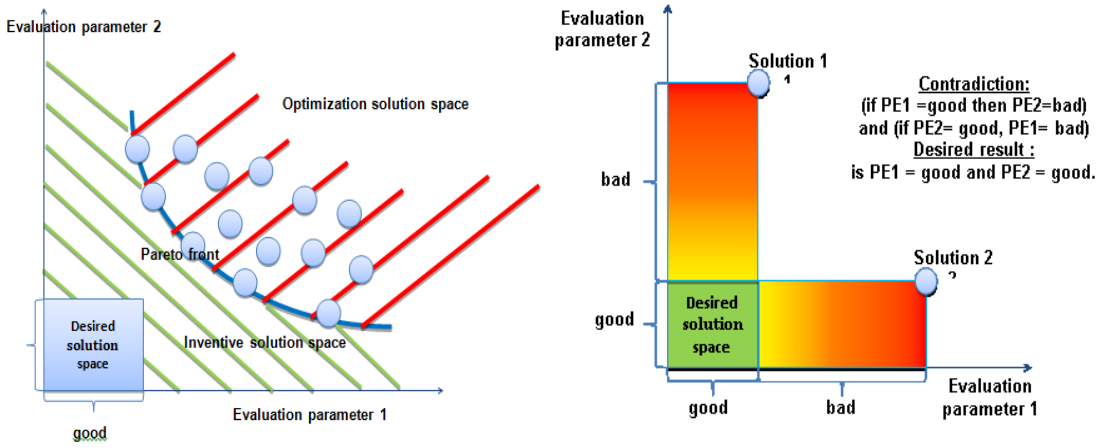


Figure 2: From optimization to invention through a contradiction

1-2.3 Dialectical approach of TRIZ theory

The dialectic approach is a philosophical method known since antiquity where one searches for arguments to solve disagreements. This approach provides a deep understanding of problems by using a universal method involving the evolution of the system and its context. The evolution of the system is characterized by changes that are initiated by contradictions. The development of a dialectical system, as discussed by Hegel [19], presents the emergence of a logical contradiction and its subsequent sublation that signifies the movement of mind or development of a material condition.

Based on dialectics and its underlying principles, a TRIZ model of contradictions is a key element for several inventive problem solving methods in the area of inventive engineering design. The OTSM-TRIZ [20] is a dialectical based theory where any existing problem has to be recognized as existing because of a set of contradictions coming from the confrontations of system parameters.

The motivations to use the dialectical approach are twofold. On one side, contradictions permit a better understanding and formulation of the problems regarding inventive design. Alternately, the problem solving procedure can be perceived strictly as a procedure for solving contradictions. Another argument to use the dialectical approach is the possibility to couple the optimization and inventive solving principles to improve the performance of the design problem solving process.

The contradictions must be clear and understandable in the early stage of the resolution process so as to choose one contradiction to resolve. The practice has shown that a bad choice of a contradiction can lead to a decrease in the effectiveness of the problem solving process. Hence, the extraction and the interpretation of contradictions play an essential role in the dialectical approach. TRIZ does not resolve the question of the appropriate choice of contradiction. Our hypothesis explaining this particular practical problem of contradiction choice, where the contradiction concept within TRIZ, should be reworked and clarified.

As mentioned before, TRIZ is a theory initiated by Altshuller [21],[22], which is dedicated to the synthesis of methods for the resolution of inventive problems, i.e., for problems for which the solver does not know of any solution. There exist three main axioms: the consideration of generic tendencies to describe the technical systems evolution, the contextual aspect of a problem, and the contradictory aspect of a problem. The first two axioms provide clarification of the core problem and input into evaluation of a potential solution, whereas the third axiom is directly linked to the representation of the problem. The last axiom identifies the contradictions inherent to a problematic situation inside a system. In its original definition, contradictions within TRIZ are defined at three different levels. These three levels correspond to a progressive understanding of the problem origin:

- **Administrative contradiction** is a situation where an objective is given, but not satisfied. This level corresponds to the identification of problem existence where something has to be changed. However, the solver is not aware of the required process.
- **Technical contradiction** (TC) expresses the opposition between two parameters where one factor improves while another one deteriorates. The technical contradiction is directly linked with the impossibility to satisfy two specification parameters simultaneously.
- **Physical contradiction** (PC) reflects the problem of a system parameter that must exist in two different states simultaneously. This contradiction identifies the core problem where one system parameter must be in different states to satisfy different specification parameters.

The idea of contradiction has been reinforced within the boundaries of the general theory of advanced thinking (OTSM-TRIZ) based on dialectical thinking, which is typically well adapted to solving technical design problems [20]. The problem is stated in the shape of the contradictions used for finding a contradiction-free model within the framework of the given objectives [23]. The administrative contradiction is not kept within the border of OTSM-TRIZ, as this contradiction definition only refers to the objective where no corresponding solving tool exists. Two types of contradiction are proposed in OTSM-TRIZ: the contradiction of a system and the contradiction of the parameter, which respectively generalize the TRIZ technical and physical contradictions. Moreover, a system of contradiction is proposed within the framework of OTSM-TRIZ to build coherence between the levels of contradiction of the system and contradiction of parameter. OTSM-TRIZ stated an element-name of property-value (ENV) model to describe the contradiction. For unifying the notation, we propose a design model, which includes three types of components:

- **Elements** [24]: Elements are constituents of a system. For example, the hammer drives the nail, i.e., an element = a hammer.
- **Parameter** [24]: Parameter describes elements by assigning them a specificity that which reflects an explicit knowledge of the area observed. There are two categories of parameters:
    - Action parameter (AP): an AP is also called a control parameter, and its states can be modified by designer.
    - Evaluation parameter (EP): an EP was observed to evaluate positive or negative system performance based on the designer's required criteria.
- **Value** [24]: Values are primarily adjectives used to describe a parameter such as the weight of the anvil should be heavy. In this case, Value=heavy.

In Figure 3, two values of action parameter form a physical contradiction. For example when AP= VALUE 1, EP1 improves; however, the EP2 condition will worsen. This technical parameter is known as technical contradiction 1. However, when AP=

VALUE 2, the EP2 improves while EP1 worsens, which is known as technical contradiction 2. TC 1 and TC 2 are mutually complementary.

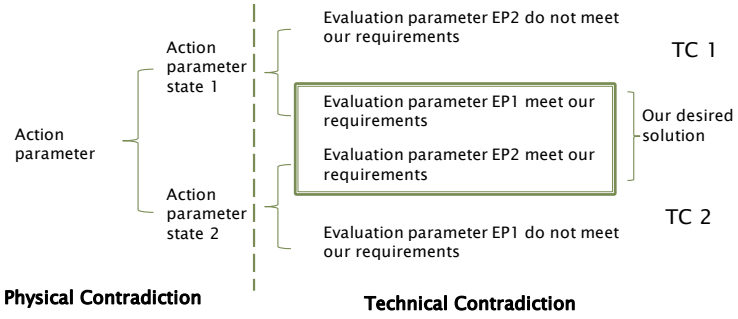


Figure 3: OTSM-TRIZ system of contradiction

Note that during the evolution of TRIZ methodology, technical contradictions first appeared in conjunction with suitable solution tools. Subsequently after a number of years, the conception of physical contradiction arose with a new family of solution tools. For Altshuller, the physical contradiction reflected a deeper contradiction in the sense of dialectics. The assumption is that behind every technical contradiction was hiding a more fundamental physical contradiction.

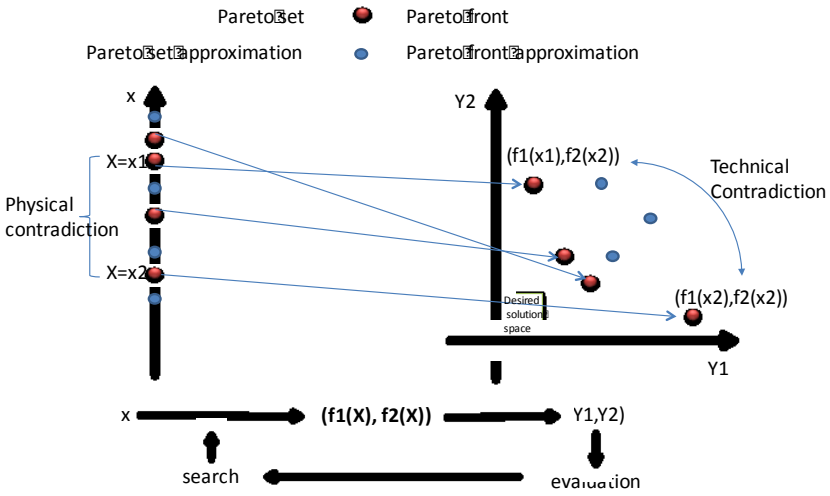


Figure 4: TC and PC in the context of optimization problem space

The Figure 4 illustrates the technical and physical contradiction in the context of the optimization problem space. The technical contradiction appears in the objective space as a conflict between two evaluation parameters while the physical contradiction appears in the decision space and expresses the fact that the action parameter X should have two different values (x1 and x2) at the same time to satisfy Y1 and Y2 (f1(x1),f2(x2)).

A significant number of action and evaluation parameters related by numerous links arose with the evolution of more complex design problems. As a result, it is difficult to browse the design space because of design constraints, which narrow the opportunities for obtaining the desired solution [25]. In order to fill the gap between contradiction gathering and representation, the concept of representing a design problem as a network of contradictions while using semantic rules to drive a design using a network was introduced in [26]. Additionally, the network problem was proposed in [27],[28], which can be transferred into a network of contradictions. A formal definition of a contradiction and its potential variations was proposed in [29]. The concept of a contradiction cloud as a three-value graphical representation of a set of elementary contradictions was presented in [24]. A significant number of these proposals are based on technical contradictions and contribute to transferring a complex problem into a classical TRIZ contradiction, which can be solved by TRIZ inventive principles.

As far as classical TRIZ contradictions are concerned, there are several limitations and gaps in their definition and utilization. We note in certain situations the proven absence of contradictions, appearing from the available relations between the variables of the system, which corresponds to the contradiction definition provided by the classical TRIZ approach [30]. The observed general trend can be summarized as follows: the more experiments and knowledge regarding a system that exists, the lower the chance of finding a technical contradiction (i.e., an input for inventive problem-solving methods). Another inconvenience is that the classical technical contradiction considers only two evaluation parameters. Supposing that there exists a technical contradiction that can be solved, nothing can be said about the satisfaction of the other evaluation parameters. Moreover, there is a lack of the explicit definition of the context required to validate the contradictions for the solving method.

In order to address these limitations, the concept of a generalized system of contradictions (GSC) involving generalized technical contradictions (GTC) and generalized physical contradictions (GPC) was proposed in the previous work [31] as an enhanced equivalent to the classical TRIZ contradictions. These generalized concepts avoid situations where no classical TRIZ technical and physical

contradictions exist as it was mentioned before. The GTC model in Figure 5 replaces two evaluation parameters defined in a classical technical contradiction with two concepts of evaluation parameters. A concept consists of an evaluation parameter or a logical disjunction of several evaluation parameters. One parameter can only participate in one of the two concepts involved in a GTC. The desired result is the simultaneous satisfaction of the two concepts. In each concept, there is at least one or more evaluation parameters where the solution of each generalized technical contradiction should satisfy all the evaluation parameters associated with the two concepts. Thus, the result will be improved over the case of classical technical contradictions. Note that the classical TRIZ contradiction is a special case of generalized contradiction.

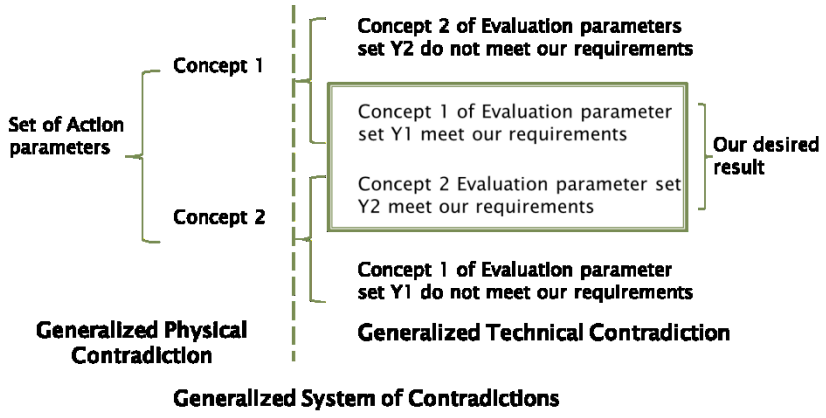


Figure 5: Generalized System of Contradiction

Another advantage of the generalized technical contradiction is that it bridges the multi-optimization and contradiction as shown in Figure 6. Contrary to the classical TRIZ contradictions which account for just two EPs, the generalized system of contradictions considers additional evaluation parameters and defines the validity of the context of contradictions through the values of the other action parameters.

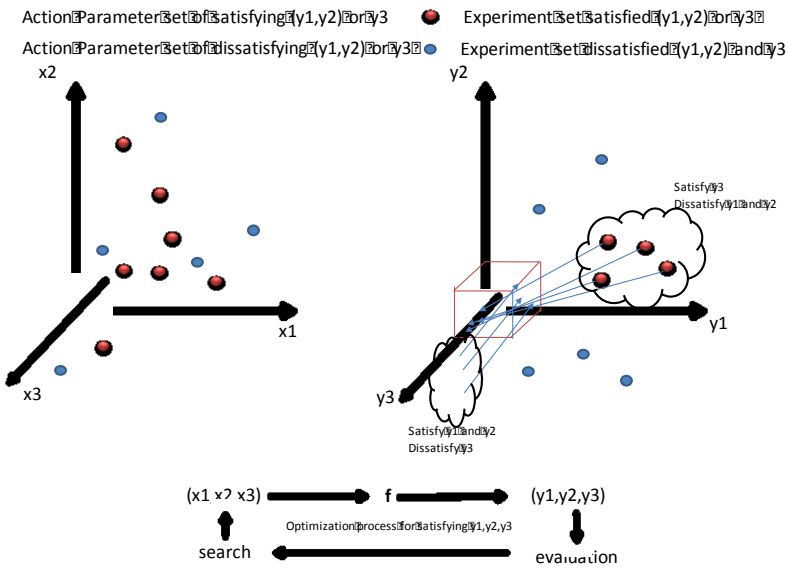


Figure 6: GTC and GPC in the context of optimization problem space

1-2.4 Theoretical framework for design problem solving

As said previously, the solving process of inventive design problems implies the evolution of the system and two categories of evolution are possible: the system efficiency is improved through optimization of system parameters or redesign of the technical system as an answer to system changes. The first category of technical evolution uses optimization solving principles whereas the redesign effort uses inventive solving principles. Several practical indications have shown that it may be less expensive to use an inventive problem solving method even if the problem can be solved by optimization. However, in many situations both approaches are required to provide satisfactory solutions and should be used in sequence. At the beginning of the design process, one cannot predict the type of solving principle required.

A general framework for design problem solving process based on the simulation—optimization—invention loop used in our research works is proposed and shown in Figure 7.

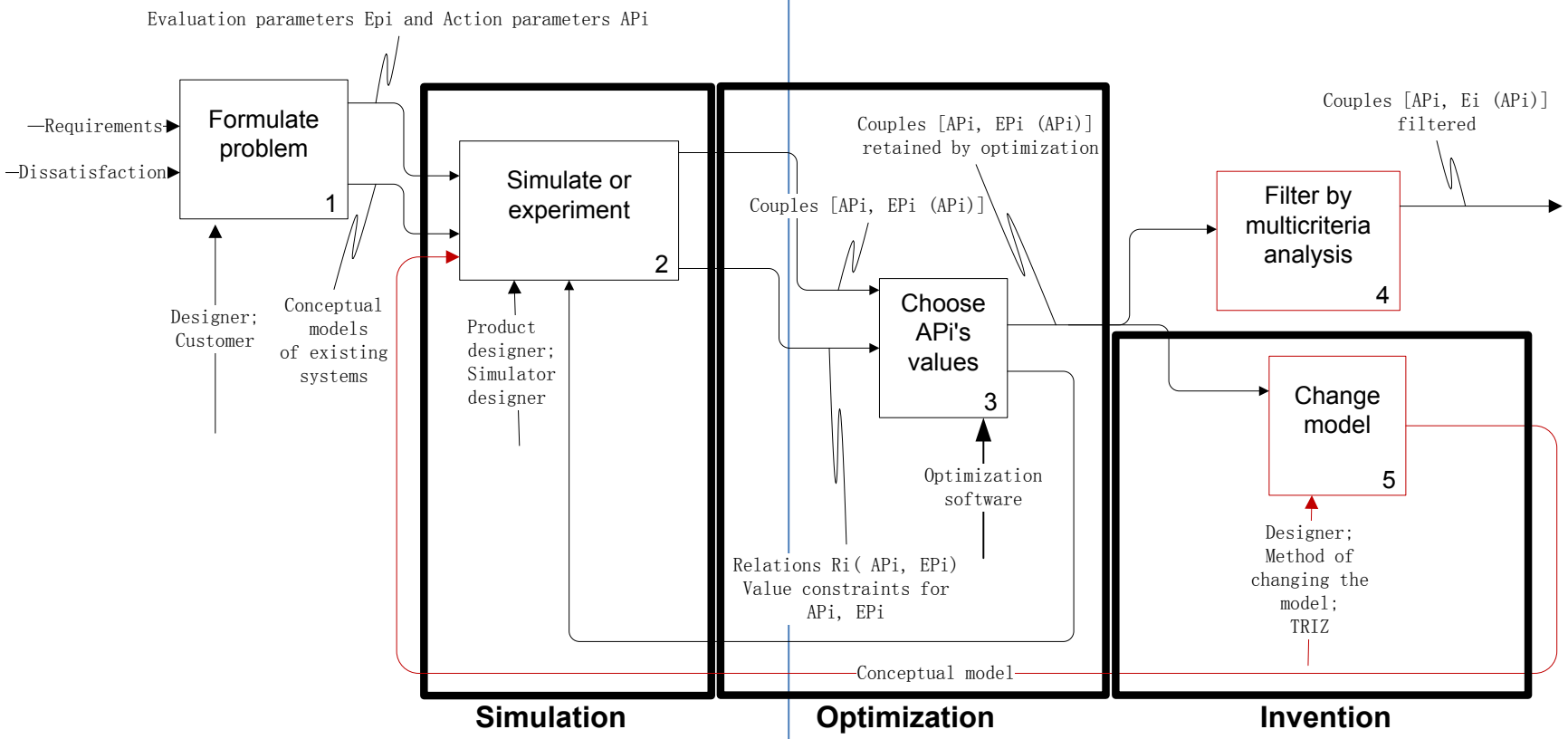


Figure 7: General framework for inventive design problem solving

This solving approach is possible when simulation or experimental means are available and involves five functions (represented by boxes in Figure 7) which can be independently performed using different methods and tools such as design of problem models (e.g., quality function deployment, design of experience, …); simulator models (e.g., CAD, Witness), simulation algorithms (e.g., stochastic optimization, genetic algorithms), multi-criteria decision analysis, or inventive changing model methods such as TRIZ.

The first function represents the problem formulation and definition through the requirements and the dissatisfaction of the customer. The evaluation parameters (EPi) are used to describe the objectives and are measured to check whether the customer's requirements are satisfied. The action parameters (APi) with their possible values represent system (decision) variables on which one can act. Some relations (Ri) between system variables and parameters are described through the system constraints. The second box generates the experiments based on a valuation of possible solutions, which are obtained by the aid of a simulator or by physical experimentation. Occasionally, a significant number of action parameters and their values are accessible, but it becomes impossible to process all data. Thus, the third function should choose relevant action parameters for the valuated experiments with possible solutions such as design of experiments or optimization algorithms. When satisfactory couples (APi, EPi(APi)) are obtained (i.e., their evaluation parameters achieve the expected values) or when the time allowed for experiment has expired; the results are filtered by a multi-criteria analysis as represented by the fourth box. When the evaluation parameters do not achieve the expected values after the time allowed for experiment or because of proven limitations of the system as determined by the experiments, it is necessary to change the conceptual model as represented by the fifth box to achieve the requirements. The use of a dialectical inventive solving approach is proposed. Then the loop starts again with a new model by performing Function 2 to modify the conceptual model.

The contribution of this thesis concerns the enhancement of the methodology proposed in [32] for performing the fifth box involving the change of the conceptual model. For inventive design problems, it is necessary to develop the first understanding of the problem and the first model of the problem. The goal is to

compute the research extraction and identification of the meaningful contradictions, which will lead to model changes and further the problem resolution. Figure 8 summarizes the proposed methodology for model change.

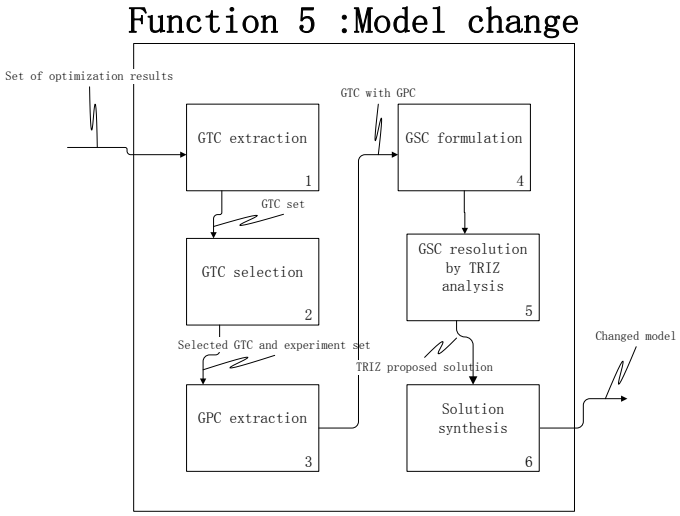


Figure 8: Approach for model change

Once experiments have been performed and no satisfying solution was determined by optimization methods, the experimental data serves as an input into the search of contradictions. The contradictions should be solved to continue the problem evolution resulting in changes to the system model in order to satisfy the customer's requirements. First, the GTCs are extracted using the proposed exhaustive search algorithm. As the number of GTC can be very large, the question of selecting the GTCs for consideration is crucial. We propose to choose the GTCs that are built from points situated on the data Pareto frontier, i.e., the GTCs that are not dominated by any other point. Once a GTC is selected, the related general physical contradictions are searched. A second algorithm for GPC extraction should be used in order to complete the system of contradictions. Then the GSC is solved by TRIZ inventive solving methods.

1-2.5 Contradiction model for optimization and inventive approach

The common representation model of a design problem is necessary to enable shifting from optimization representation models to inventive models. The representation model should support the simulation-optimization-invention loop and

should enable the simultaneous use of both optimization and inventive solving strategies. In order to extract the generalized contradictions, information regarding the technical system is required. The use of the experimental design is a good starting point as it involves a strategy for gathering empirical knowledge on the studied technical system. In other words, the gathering of knowledge based information on the analysis of the experimental data and expressed in the rectangular experiments table (Table 1), not in theoretical models. In general, we generate the experiments in the table with raw data. If it is possible we develop a complete design of experience, or if this is not possible because of too many variables, we will then randomize according to uniform law among all possible experiments from the research space.

Table 1: A table of experiments

	x_1	...	x_l	y_1	...	y_i	...	y_r
e_1	v_{11}		v_{1l}	z_{11}		z_{1i}		z_{1r}
e_2	v_{21}		v_{21}	z_{21}		z_{2i}		z_{2r}
...								
e_{k-1}	v_{k-11}		v_{k-1l}	z_{k-11}		z_{k-1i}		z_{k-1r}
e_k	v_{k1}		v_{kl}	z_{k1}		z_{ki}		z_{kr}

As shown in Table 1, the rows of the table represent the experiments and each column corresponds to different process variables expressing one system parameter. In each experiment as noted by E, one or more process variables or factors are changed in order to observe the effects these changes have on one or more response variables or outputs. The factors are the controlled parameters usually noted as X and correspond to the action parameters in the GSC model. The outputs are the measured parameters usually noted by Y and correspond to the evaluation parameters in the GSC model.

Once the experiments are complete, we can begin to organize and interpret the data. First, the response variables are transformed into a binary system in order to simplify the extraction problem. The purpose is to obtain two sets of evaluation parameters that are in contradiction. This contradiction is translated as two orthogonal blocks of

ones obtained by permutations of the rows and columns of the rectangular matrix. Different methods of data analysis can be used to identify the blocks in the matrix.

The properties of the Generalized System of Contradictions can be characterized by the set of definitions that enables the extraction of the GSC from the experimentation table. An experimentation table can be characterized by:

- A set of action parameters X=(x0, x1, …, xn),
- A set of domains D=(D0, D1, …, Dn) where Di defines the possible range of values for xi,
- A set of evaluation parameters Y=(y0, y1, …, yp) characterized by binary values, either 1 if yi is satisfied, or 0 if yi is not satisfied, and
- A set of experiments E=(e0, e1, …, em). An experiment ei is a particular instantiation of the action parameters: (ai1, ai2, …, ain), such that aij α Dj combined with the induced values of evaluation parameters (zi1, zi2, …, zip) resulting in the binary values of zij = 1 if yj is satisfied by experiment ei, or zij = 0 if yj is not satisfied by experiment ei.

The goal is to satisfy all the evaluation parameters. However, the situation should be considered where such a solution does not exist in the considered table above, i.e., that no experiment enables the satisfaction of all the evaluation parameters.

A Generalized System of Contradictions seeks to identify the following table of experiments:

- Three sets of evaluation parameters Y_0, Y_1 and Y_2, such that $Y_0 \cap Y_1 = \varnothing$, $Y_1 \cap Y_2 = \varnothing$, $Y_0 \cap Y_2 = \varnothing$, $Y_0 \cup Y_1 \cup Y_2 = Y$, $Y_1 \neq \emptyset$ and $Y_2 \neq \emptyset$.
- Three sets of experiments E_0, E_1 and E_2 such that $E_0 \cap E_1 = \varnothing$, $E_1 \cap E_2 = \varnothing$, $E_0 \cap E_2 = \varnothing$, $E_0 \cup E_1 \cup E_2 = E$, $E_1 \neq \emptyset$ and $E_2 \neq \emptyset$.

Moreover ;

- E_1 is a set of experiments for which all the evaluation parameters of Y_1 are satisfied.
- E_2 is a set of experiments for which all the evaluation parameters of Y_2 are satisfied.

Such a definition provides a path for reorganizing the experimentation table by permutations of the rows and of the columns in order to group the previously defined Ei and Yi [33](Table 2).

Table 2: GSC representation in experimentation output

X	Y_1	Y_2	Y_0
E_1	$E_1 \times Y_1$: $z_{ij}=1$	$\forall e_i \in E_1$ $e_i \times Y_2$: $\exists j / z_{ij}=0$	$E_1 \times Y_0$
E_2	$\forall e_i \in E_2$ $e_i \times Y_1$: $\exists j / z_{ij}=0$	$E_2 \times Y_2$: $z_{ij}=1$	$E_2 \times Y_0$
E_0	$E_0 \times Y_1$	$E_0 \times Y_2$	$E_0 \times Y_0$

Based on this definition, it is possible to show that existing TRIZ-based models are particular models derived from the GSC. The classical TRIZ contradictions are a particular case of GSC (see Table 3), where:

card $Y_{i,\ i=\{1,2\}}=1$, and

$\varepsilon\ (x_i, v_1, v_2)\ \alpha\ (X, D_i, D_i) / (x_i=v_1 \bullet (Y_1, Y_2)=(1,0))\ \&\ (x_i=v_2 \bullet (Y_1, Y_2)=(0,1))$

Table 3**:** Classical TRIZ contradiction

							Y1	Y2	Y0		
		x1	…	xg	…	xn	ys	yu	yw	…	yh
E1	ei	ai1		v1		ain	1	0			
	…										
	ej	aj1				ajn					
E2	ek	ak1		v2		akn	0	1			
	…										
	el	al1				aln					
E0	eq	aq1		aq2		aqn					
	…										
	er	ar1		ar2		arn					

These details indicate the generic aspect of the GSC, which enables definition under certain conditions of existing TRIZ-based contradictions. The classical TRIZ model of contradiction is straightforward to solve as it is easier for a human expert to

interpret. However, the GSC is more difficult to interpret and solve, thereby presenting a problematic situation for which no solution is known.

The extraction of the GSC in the experimentation table is described as a set of equations characterizing the blocks of the matrix [34]:

$$z_{ij} = 1 \text{ or } 0$$

$$\sum_{i/e_i \in E_k ; j/y_j \notin Y_k} \overline{z_{ij}} \geq 1$$

$$\forall (e_i, y_j) \in E_{k\neq 0} \times Y_{k\neq 0}; z_{ij} = 1$$

$$\forall (e_i, y_j) \in E_0 \times Y_{k\neq 0} : \sum_{j/y_j \in Y_k} \overline{z_{ij}} \geq 1$$

$$\forall (e_i, y_j) \in E_{k\neq 0} \times Y_0 : \sum_{i/e_i \in E_k} \overline{z_{ij}} \geq 1$$

The matrix is divided into nine blocks, and in Table 2, we have formulated the features into blocks. In the blocks where E1 × Y1 and E2 × Y2; all of the elements are equal to 1. In the remaining 4 blocks associated with Y1 and Y2, there must be at least one element equal to zero in each row.

To extract classical TRIZ contradictions, the following set of equations is resolved:

$$z_{ij} = 1 \text{ or } 0$$

$$\sum_{i/e_i \in E_k ; j/y_j \notin Y_k} z_{ij} = 0$$

$$\forall (e_i, y_j) \in E_{k\neq 0} \times Y_{k\neq 0}; z_{ij} = 1$$

$$\forall e_i \in E_0 : z_{ij, j/y_j \in Y_1} + z_{ik, k/y_k \in Y_2} \leq 1$$

$$\forall e_i \in E_1 : a_{ij} = v_1$$

$$\forall e_i \in E_2 : a_{ij} = v_2$$

$$v_1 \neq v_2$$

The differences between these sets of equations required to extract the different types of TRIZ-based contradictions show that the classical TRIZ system of contradiction is the most constrained system. Such a contradiction has the limitation of not fitting the equivalence (i.e., no solution) ⇔ (a contradiction exists). These descriptions of the existing TRIZ-based contradictions show that it is possible to

define generic contradictions. Additionally, our interest for the GSC is based on the satisfaction of the previously defined equivalence.

1-3 Problematic concerns

In the current practice, the system of contradictions is identified by interviewing human experts. In our previous research, we have shown that there are some cases in which no classical TRIZ contradiction exists and the problems still cannot be solved by optimization. Therefore, the concept of a generalized system of contradictions was proposed. These generalized contradictions are typically not searched by human experts as their expression is too difficult to interpret by the human mind. For a human expert it is simple to validate a generalized contradiction; however, it is much more difficult for a human expert to define a generalized contradiction. Moreover, looking for simple technical or physical contradictions as represented by the classical TRIZ model of contradictions, the human practitioner could be faced with reaching their own expertise limits when the system is too complex or when they have no relevant knowledge about the system.

This thesis contributes to answer the problematic concerns as situated into three different levels. The following discussion specifies the three levels and related questions.

The first level of questions regarding the design theory is concerned with the concept of contradiction, which is one of the foundations of TRIZ:

Question 1: When no classical technical or physical contradiction exists, do the generalized technical and/or physical contradictions exist and are there a significant number of those contradictions? Can we always extract the generalized contradictions intrinsically the same way as Pareto, for example, from the behavioral representation and the objectives of the system? If so, then what are the consequences?

Question 2: How can the generalized contradictions be exhaustively identified and extracted?

Question 3: Once all generalized contradictions are known, how can the relevant contradictions be chosen or defined? Alternately, how can a relevant contradiction be defined in a straightforward manner?

The second level of questions regarding the methodology concerns the practical consequences of the new definition of contradictions for the inventive problem solving:

Question 4: Once the contradictions have been identified and extracted, how can we use them in the inventive problem solving process?

Question 5: How can we extract the relevant contradictions without exhaustive research, which is often too expensive and time consuming despite a posteriori filtering?

Question 6: Can we use the concept of a generalized contradiction to express the implicit knowledge from a system expert?

Finally, the third level of questions discusses the exploration of the relationship between optimization methods and TRIZ in order to develop cross-fertilization from a theoretical and/or a practical point of view.

Question 7: Can we use methods and concepts from the optimization in order to facilitate the identification of generalized contradictions?

Question 8: Is there a relationship between the Pareto concepts and the generalized contradictions?

Question 9: If this link exists, could it be used to identify generalized contradictions?

Question 10: Alternately, if this link exists, could it be exploited to facilitate the optimization process?

One of the objectives of this thesis is to answer the previous questions. The research strategy adopted to address these questions is described in the next section.

1-4 Research method

In order to answer the questions proposed in the previous section, it is necessary to build an exhaustive extraction tool to identify and extract the generalized technical and physical contradictions from system data. To do this, the problems of identifying generalized technical and physical contradictions are modeled in the form of combinatorial optimization problems along with solving the algorithms, which are

proposed for each case. Once these algorithms have been realized, they can be used for empirical studies that are necessary to answer the questions Q1 and Q2. Furthermore, they also help us to develop hypotheses required to answer questions Q4 and Q6. We previously answered the question Q7 in a preceding section.

The exhaustive methods have their limitations. The methods can be implemented for a system with a limited number of variables because of the complexity of time calculation. Alternately, the number of generalized contradictions is significant as indicated by the answers to Q1 and Q2. This limit is reflected in the number of action parameters. In order to reduce the limitations related to the number of variables, our strategy is to analyze the data to identify the action parameters and their values involved in the concepts of generalized physical contradictions before determining these concepts. This can simplify the system by reducing the number of action parameters by only considering the influencing parameters. The exhaustive algorithm can then be used for the simplified system with a reduced number of variables.

This approach of data preprocessing permits the reduction of the number of action parameters and their values. Then identifying the contradictions is realized by providing the answers to questions Q8 and Q9. An alternative solution to this sequence, which is not developed in this thesis, is to design a heuristic based research algorithm of relevant contradictions by using the data without including the exhaustive research of contradictions. The development of these algorithms requires the identification of specific properties to the relevant contradictions. The search for these properties can be performed by experimentation, using real cases involving human inventive design experts coupled with the analysis of the exhaustive search results. In this thesis, we used an academic school example from the logistics domain as the test case to answer question Q5. In the context of this thesis, we do not expect to provide a complete answer to question Q4; however, we believe a contribution to the answer of Q4 can be accomplished by testing the previous heuristics on the real or academic examples with the appropriate number of variables. Question Q10 is discussed according to the synthesized results developed for questions Q8 and Q9.

1-5 Organization of the thesis

In order to remind the previous research method and the answers to the above questions, the thesis chapters are organized as follows. Chapter 2 and 3 deal respectively with the identification and extraction of generalized technical and physical contradictions. The identification problem is formulated as an optimization problem, specifically a binary programming problem that can be organized into sub-problems based on the original problem properties. This sub-problem is proven NP-hard (non-deterministic polynomial-time hard). The combinations of sub-problem solutions provide the generalized technical contradictions. The exhaustive search algorithm is proposed based on the input data issued from the experimental table of the physical system from the simulators. The time and space complexity as well as the limitations of the algorithm are presented and evaluated based on the example of an electrical circuit breaker.

We can relate several generalized physical contradictions to one chosen generalized technical contradiction to form a generalized system of contradictions. Thus, Chapter 3 proposes an algorithm to search all the generalized physical contradictions related to one chosen generalized technical contradiction. The limitations in terms of number of variables and their possible values to be processed are discussed as well as the contributions to our problematic concerns and potential applications. The algorithm is illustrated for the examples of the electrical circuit breaker and the single inventory Kanban system.

The use of exhaustive search algorithms proposed in Chapters 2 and 3 has its limitations related to the number of possible variables that can be processed because of the computing time, which increases exponentially with the number of parameters. For the GTC research, the algorithm can only evaluate 15 evaluation parameters while for the GPC research the algorithm can only evaluate 12 action parameters with binary values. Nevertheless, the use of the existing algorithms provides practical evidence that only a few action parameters within the model are involved in the description of the physical contradictions. The purpose of chapter 4 is to define reduced sets of action parameters that are relevant candidates for generalized physical contradictions or eliminate those that are not defined

beforehand. This may allow the use of exhaustive physical contradiction search algorithms for systems described by more than 12 action parameters including two values, or facilitate a human search of the physical contradictions. To accomplish this task, a search of the parameters is stated as a set of classification problems where an adaptation of a support vector machine (SVM) feature selection algorithm is proposed to address the problems. The limitations of this algorithm are also discussed. Finally, strategies for using the proposed SVM algorithm within the GPC extraction context are suggested.

Chapter 5 provides a synthesis on how to combine the algorithms within the inventive solving process that is illustrated using the example of a double Kanban system.

Finally, chapter 6 summarizes the contributions and limitations of our work, discusses the answers to the 10 questions posed previously in this introductory chapter and proposes a prospective for future research.

Chapter 2 Extraction of generalized technical contradiction

In this chapter, the problem of how to identify and extract the generalized technical contradictions in the exhaustive manner is discussed. First, the brief state of the art of different approaches and methods of technical contradiction extraction is introduced and the limitations and gaps are identified. Our methodology has a goal to fill this gap starting with the identification problem, which is described and formulated as a binary programming problem whose resolution is organized into a set of sub-problems. This combinatory problem is NP-hard and the solutions to this problem are the generalized technical contradictions. The research space of these contradictions is defined by the number of system parameters and the number of experiments involved. The theoretical assumptions and the binary programming model were used to formalize the extraction of the generalized technical contradictions. The exhaustive search algorithm is proposed in order to identify all generalized technical contradictions.

Data derived from the experiment of designing the physical system or data from the simulator were utilized. The time and space complexity of the algorithm as well as the limitations were analyzed. The illustration of the algorithm was performed using the example of the electrical circuit breaker. To be more general in our experimentation, we randomly generated a population of binary matrices for comparison to the results from the electrical circuit breaker. A statistical analysis was then performed to interpret the general results and identify the principal influencing factors affecting the number of identified technical and generalized technical contradictions. This exploitation of the algorithm will permit answering several questions posed within our problematic concerns.

2-1 State of the art in the technical contradiction extractions

As presented in the background of Chapter 1 several models of contradictions were introduced as provided by [16],[21],[20]. Many of these proposals are based on the concept of TRIZ technical contradictions. A technical contradiction occurs between two system evaluation parameters and exists if the improvement of one parameter leads to degradation of the other.

Different authors [35],[36],[37] proposed formal procedures on how to identify and extract the contradictions based on text or patent analysis. Cascini et al. in [35] proposed a specific algorithm by using computer aided analysis of patents based on textual description, which allowed highlighting relevant details of a patent. Specifically, the algorithm highlighted relevant design parameters, improved features, or the motivation for the patent. The goal was to speed up the identification of contradictions solved by the invention to assess its invention level for correlation with other evolutionary parameters. The algorithm used text-mining tools that have achieved relevant capabilities for extracting useful information from a large set of documents. However, no specific means are available to support the analysis of patents based on the aim of identifying the contradictions underlying a given technical system.

The automatic discovery and the classification of TRIZ contradictions from patent texts is one of the primary natural language processing challenges for TRIZ [37]. This would enable the emergence of systems capable of providing real-life examples on the conflict currently being resolved by the innovator. Though some approaches addressing the problem of contradiction mining exist, their accuracy is low and the methods can be considered relatively naive. The complexity of the task requires modern approaches that can combine statistical knowledge with domain thesauri or ontologies analysis.

The Knowledge Media Institute team in the context of Tech It Easy project [37] has built an experimental system that uses information extraction techniques to detect key sentences in a patent text that are likely to represent a contradiction. However, reliable discovery and classifying of TRIZ contradictions was not planned and so was not achieved by that specific system.

Different authors use optimization and simulation methods to extract the contradictions from the technical system analysis such as Cugini et al. in PROSIT approach [38] where the optimums are searched for each parameter (situation) and the human experts manually select the contradictions. Even if the simulation is used to retrieving the data, the contradictions are manually identified and extracted by human experts. The mathematically formalized knowledge is used to disclose geometrical contradictions, which is a specific type of physical contradiction, by using a topological optimization algorithm.

Some authors explore problem formulation and contradiction statements based on networks of non-formalized data [39],[20],[24],[25],[27],[26],[28],[30],[40]. The concept of representing a design problem as a network of contradictions and using semantic rules to drive a design using this network was introduced by Cavallucci et al. [26]. A formal definition of contradiction based on the ontology and its potential variations was proposed by Rousselot et al. [29]. The concept of a contradiction cloud as a three-value graphical representation of a set of elementary contradictions was presented by Cavallucci et al. [24]. An algorithm for extracting the most important contradiction in a network of problems was proposed by Baldussu et al. [25].

Despite the fact that different models of contradictions exist, there are gaps in the formalized procedures utilized for contradiction gathering and extraction. The methods presented are linked to qualitative data analysis; however, we are proposing a quantitative analysis tool based on the exhaustive search principle. To our knowledge there is no automatic or formalized tool that allows for exhaustive identification and extraction of contradictions.

2-1.1 Generalized contradictions' extraction within experimental table

Our methodology for design problem solving is based on the general framework involving optimization and inventive methods. The generalized system of contradiction (GSC) is proposed as an inventive model for a problem concerning a technical system. The GSC consists of generalized technical contradictions (GTC) involving system evaluation parameters on one side and generalized physical contradictions (GPC) adding the system action parameters on the other side. To

extract the contradictions, information about the technical system is required. The outputs of the experimental design developed for study of the technical system are used to gain the necessary information. For the purpose of our study, these results were transformed and summarized into a binary rectangular matrix. The first step of our approach is to identify and extract the GTCs. Assuming the GTCs are not related to action parameters, the binary matrix can then be reduced to its evaluated form denoted as Z, as shown in Figure 9 a).

$$\begin{array}{c} \\ e_1 \\ e_2 \\ \dots \\ e_m \end{array} \begin{array}{c} \begin{array}{cccc} y_1 & y_2 & \dots & y_n \end{array} \\ \begin{pmatrix} z_{11} & z_{12} & \dots & z_{1n} \\ z_{21} & z_{22} & \dots & z_{2n} \\ \dots & \dots & \dots & \dots \\ z_{m1} & z_{m2} & \dots & z_{mn} \end{pmatrix} \end{array}$$

	y1	y2	y3	y4	y5	y6
e1	1	0	1	1	1	1
e2	0	1	0	0	1	1
e3	1	0	1	0	0	0
e4	1	1	1	1	0	0
e5	1	0	1	0	1	1
e6	0	1	0	1	1	1
e7	1	0	1	0	0	0
e8	1	0	0	1	1	1
e9	0	1	0	1	1	1

(a) (b)

Figure 9: Generic matrix Z (a) and an example (b)

The rows of the matrix represent the experiments while the columns correspond to the evaluation parameters. The generic term zij of matrix Z equals 1 when the experiment ei meets the design requirements for the evaluation parameter yj otherwise zij equals zero. The design goal is to satisfy all of the evaluation parameters, i.e., to find any row of ones. When it is not possible to obtain such an output by using only the action parameters, inventive methods must then be implemented. Identifying GTCs in the customized experimental table representation requires defining the properties of the GTCs. GTCs are then characterized by a set of definitions, which allow the extraction of GTCs from this experimental table. Assuming no experiment satisfies all of the evaluation parameters at the same time, identifying a GTC in such a table involves searching for the following:

- Three sets of evaluation parameters (Y_1, Y_2, Y_0) whose union is the entire set of evaluation parameters and whose intersection is an empty set.
- Three sets of experiments (E_1, E_2, E_0) whose union is the entire set of experiments and whose intersection is an empty set.

- The satisfaction of the first set of evaluation parameters Y1 for the first set of experiments E1 and the satisfaction of the second set of evaluation parameters Y2 for the second set of experiments E2.
- At least one evaluation parameter from the second set of parameters Y2 that is not satisfied in each experiment of the first and third sets of experiments E1 and E0.
- At least one evaluation parameter from the first set of parameters Y1 that is not satisfied in each experiment of the second and third set of experiments E2 and E0.

Table 4: Generalized technical contradiction expressed in experimental table

	Y_1	Y_2	Y_0
E_1	$E_1 \times Y_1 :$ $z_{ij} = 1$	$\forall e_i \in E_1$ $e_i \times Y_2 : \exists j, z_{ij} = 0$	
E_2	$\forall e_i \in E_2$ $e_i \times Y_1 : \exists j, z_{ij} = 0$	$E_2 \times Y_2 :$ $z_{ij} = 1$	
E_0	$\forall e_i \in E_0$ $e_i \times Y_1 : \exists j, z_{ij} = 0$	$\forall e_i \in E_0$ $e_i \times Y_2 : \exists j, z_{ij} = 0$	

Table 4 shows the matrix of the experimental table derived by grouping the columns and the rows. The matrix has been divided into 9 blocks, and in the table, we have formulated the features into blocks. In the blocks E1×Y1 and E2×Y2, all of the elements are equal to 1. In the remaining 4 blocks associated with Y1 and Y2, there must be at least one element equal to 0 in each row.

2-1.2 Difficulty of the problem extraction: the NP-hard problem

To identify the generalized technical contradictions, we are first interested in the size of our search space. In other words, we are interested in answering the following question: how many candidate solutions can we find when searching for generalized technical contradictions? If this problem is viewed as a searching problem, we can use a brute force algorithm to browse the solution space by an enumeration method. Although enumeration is feasible for some small matrices, it can be difficult or even impossible for large matrices. In fact, the search space grows exponentially with the

number of rows M and number of columns N of the matrix Z. The following section discusses the analysis of the complexity of the search space.

Quantifying a search space is mainly a combinatorial problem as because all possible matrix blocks can be expressed using Stirling numbers of the second kind [45]. In actuality, the sets Y_0 and E_0 may be empty. Therefore, row combinations and column combinations can be addressed using two types of combinatorial models. In the first case, when $Y_0 = \phi$, the problem can be described as a type of Ball-Box Matrix [45], which describes the situation in which N different balls (where N>2) are placed in two identical boxes and neither box is empty. In this case, the number of combinations, expressed as a Stirling number of the second kind, is *s*(N, 2). In the second case, when $Y_0 \neq \phi$, the problem can be described by a Ball-Box Matrix that describes the situation in which N different balls are placed in three identical boxes and no box is empty. However, in this case we must select 2 of 3 boxes Y_1 and Y_2. Thus, the result is $s(N,3)\times C_2^3$. The same process applies to the column number. Thus, the number of combinations of the column numbers is $s(N,3)\times C_2^3 + s(N,2)$. In addition, when we permute the two column sets and two row sets for $E_1 \times Y_1$ and $E_2 \times Y_2$, only two distinct cases arise. To summarize, we conclude that the search space size is

$$\left(s(M,3)\times C_2^3 + s(M,2)\right)\times\left(s(N,3)\times C_2^3 + s(N,2)\right)\times 2.$$

According to the equation for Stirling numbers of the second kind [45],

$$s(N,k) = \frac{1}{k!}\sum_{t=0}^{k}(-1)^t\binom{k}{t}(k-t)^N \tag{1}$$

$$s(N,2) = 2^{N-1} - 1 \tag{2}$$

Using Equations (1) and (2) and supposing $N \geq 3, M \geq 3$, we obtain the size of our search space:

$$\left(\frac{3^N+1}{2} - 2^N\right)\times\left(\frac{3^M+1}{2} - 2^M\right)\times 2 \tag{3}$$

Clearly, the browsing of the search space is an NP-hard problem.

2-1.3 Different possible methods to use in our approach

The central problem in GTC identification is the contradictory relation between different evaluation parameters. This is similar to the problem of clarifying the data structure before statistical analysis processes used within different research domains such as sociology, ecology, archaeology, geography, medicine, etc. The goal is to study the data in more detail and express the existing relations between groups, categories, or parameters of different systems or communities.

One of the widely known methods of such a statistical analysis is the **principal component analysis** PCA [41]. The PCA is a variable reduction procedure expressing the data in such a way as to highlight the data similarities and differences. The goal is to reduce the number of observed correlated variables into a smaller number of artificial variables called principal components, which are not correlated. This reduction process is closely related to singular value decomposition and typically used for the image compression. The PCA belongs to the family of linear transformation methods such as factor analysis, projection pursuit or independent component analysis. The goal of all these methods is to search special components of the representative data that are linear combinations of the original variables. The PCA involves the calculation of the eigenvalue decomposition of a data covariance matrix or singular value decomposition of a data matrix usually after mean centering the data for each attribute. The results of the PCA are usually discussed in terms of component scores and loadings. For our purpose, this method is interesting if two independent orthogonal principal components are identified as representing two sets of parameters within contradiction.

Quadratic assignment problem (QAP) is one a common combinatorial optimization problem [42]. It can be used for data analysis tasks characterized by the use of proximity matrices. The QAP is an extremely hard problem from both theoretical and practical points of view. In order to find a global optimal solution for a given QAP, one can either use methods of dynamic programming or branch and bound procedures. The problem is NP-hard, so there is no known algorithm for solving this problem in polynomial time. However, a number of algorithms have been proposed to deal with the complexity of the QAP. The most popular procedure used in QAP is

the branch and bound algorithm based on lower bounds, which are also used to evaluate the adequacy of the solutions produced by different heuristics.

The problem can also be classified as a **data mining problem** (methods and models used in data mining are reviewed in [43]). Data mining is a large interdisciplinary subfield of computer sciences involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The goal is to pre-process; analyze observational data sets to find relationships between parameters; and to represent the data in understandable and useful ways for the data owner. Cluster analysis and seriation methods are among the many data mining methods typically utilized.

The **seriation methods** [44] offer a visual means of examining the structure of the data by reorganizing the data to present as homogeneous a picture of the data structure as possible. This visual representation assists with problem interpretation. The idea of seriation is to bring similar rows and similar columns as close to each other as possible by their permutations. Then the seriated table or image reveals a structure that demonstrates the relationship between the variables. The concept of seriation is quite simple but in its general form is difficult to implement it on a computer. In our case we can use the quasi-seriation method of two classes where the residuals define Y_0 and the two classes define Y_1 and Y_2. This remains the block seriation methods [45], which searches simultaneous partition of objects (or experiments) and attributes (or variables). The principle is based on the permutation of the rows and the columns of a binary table in order to identify the diagonal disjoint blocks for maximal density. We can also cite also the matrix reordering problems, which seek to reorder objects into a sequence along a one-dimensional continuum [46] by exploratory combinatorial data analysis techniques.

The seriation methods are related to **cluster analysis** [47]. Cluster analysis is the organization of a collection of patterns – observations (usually represented as a vector of measurements, or a point in a multidimensional space) into clusters (groups) based on similarity. Typical clustering involves the pattern representation (feature extraction), definition of pattern proximity measure, the proper clustering (grouping) and eventually data abstraction and assessment of output. Clustering is a subjective process; the same set of data items often needs to be partitioned

differently for different applications. This subjectivity makes the process of clustering delicate. Different types of clustering methods exist such as hierarchical clustering; K means clustering; or two mode clustering [48],[49]. The k-means algorithm assigns each point to the cluster whose center is the nearest. For our purpose, two-mode clustering can be used where the objects and the features of the objects are clustered. This method provides a simultaneous clustering of the rows and columns of a rectangular data matrix but cannot restrict other parts of the matrix

Blockmodeling [50] deals with the clustering of a network. The goal is to reduce a large, potentially incoherent, network into a smaller comprehensible structure that can be more readily interpreted. Blockmodeling is based on the idea that units within a network can be grouped based on the extent to which they are equivalent, according to some meaningful definition of equivalence. One of the main procedural goals of blockmodeling is to identify clusters or classes of units that share similar structural characteristics in terms of equivalence relation. This method is based on the graph theory, and specifically on relations of dominance between different units. Two types of such relations are distinguished: structural equivalence and regular equivalence. In the first case of structural equivalence, the equivalent units have the same connection pattern to the same neighbors. For the case of regular equivalence, the equivalent units have the same or similar connection pattern to different neighbors. For our case, structural equivalence with complete blocks of ones should be searched.

All available methods associated with the above mentioned approaches can be used to find one or several unique solutions to our problem. However, not one method can be used to find all the solutions. In this paper, we use the binary integer programming model to analyze our problem. This results in a specific algorithm that provides all the solutions for our problem.

2-2 Formulation of the extraction problem as a binary integer program

In this section we will formulate our GTC extraction problem as a quadratic binary integer programming (BIP) problem. Then an exhaustive searching algorithm is

proposed to collect all the solutions. In addition, we propose to transform a quadratic binary integer program to a linear integer program, which will bridge the existing integer programming algorithm to allow for GTC searching. This will also allow for a supplementary approach when exhaustive searching algorithm is unavailable.

Identifying GTCs is generally considered a problem of defining how to group the columns and rows of matrix Z, which is a yes-or-no decision problem that can be transformed into a binary integer program and a linear programming problem in which all the variables are restricted to binary values [51]. Specifically for BIP, each variable can only take a value of 1 or 0, thereby denoting yes or no. Thus, a BIP can be utilized to model real-world situations by using logical expressions. For our problem, a BIP is used to express decision variables that indicate whether the columns or rows pertain to the columns and rows of a matrix block.

Suppose that matrix Z has M rows and N columns. Let us denote the parts of a 3-partition of the columns and rows as $\{Y_1, Y_2, Y_0\}$ and $\{E_1, E_2, E_0\}$, respectively. Our goal is to find all of the 3-partitions of the column set and 3-partitions of the row set that meet the requirements indicated in Table 2.

Table 5: GSC representation in experimentation output

X	Y_1	Y_2	Y_0
E_1	$E_1 \times Y_1$: $z_{ij}=1$	$\forall e_i \in E_1$ $e_i \times Y_2$: $\exists j / z_{ij}=0$	$E_1 \times Y_0$
E_2	$\forall e_i \in E_2$ $e_i \times Y_1$: $\exists j / z_{ij}=0$	$E_2 \times Y_2$: $z_{ij}=1$	$E_2 \times Y_0$
E_0	$E_0 \times Y_1$	$E_0 \times Y_2$	$E_0 \times Y_0$

First, we use two N-dimensional binary vectors $C_{i=1,2}$ to denote the two sets of columns (Y_1, Y_2). If one vector component of C_1 (resp. C_2) equals 1, the corresponding column belongs to the column set Y_1 (resp. Y_2); otherwise, it does not belong to Y_1

(resp. Y_2). Second, we use two M-dimensional binary vectors $R_{i=1,2}$ to denote the two sets of rows (E_1, E_2). If one vector component of R_1 (resp. R_2) equals 1, the corresponding row belongs to the row set E_1 (resp. E_2). The vectors are as follows:

$C_{i=1,2} = \left(c_1^i, c_2^i, c_3^i, ..., c_N^i\right)_N$, $R_{i=1,2} = \left(r_1^i, r_2^i, r_3^i, ..., r_M^i\right)_M$;

$C_{i=1,2}$ represents $Y_{i=1,2}$, $R_{i=1,2}$ represents $E_{i=1,2}$.

For convenience, we assume that $N_1 = \left(\sum_j c_j^1\right)$, $N_2 = \left(\sum_j c_j^2\right)$, $M_1 = \left(\sum_j r_j^1\right)$, and $M_2 = \left(\sum_j r_j^2\right)$.

Our problem can then be transformed into an integer programming problem. The restrictions regarding block $E_1 \times Y_1$ and $E_2 \times Y_2$ indicated in Table 2 can become an objective function, as $R_i Z C_i^T$ is equal to the sum of all elements in the block $E_i \times Y_i$. $N_i M_i$ is the product of the number of rows and columns of the block (E_i, Y_i).When the blocks (E_1, Y_1) and (E_2, Y_2) are full of "ones", the quantity $\sum_{i=1,2} R_i Z C_i^T - \sum_{i=1,2} N_i M_i$ is equal to zero and negative otherwise. It can be determined from analysis, the maximum value of this equation is zero.

The restriction for the block $E_2 \times Y_1$ shown on Table 4 indicates that the cardinality of Y_1 is greater than the sum of the elements in each row of the block $E_2 \times Y_1$. Thus, we have the inequalities for each row of $E_2 \times Y_1$:

$$N_1 - \left(\sum_{j: c_j^1 = 1, c_j^1 \in C_1} z_{ij}\right) \geq 1.$$

Using the same method to model the block $E_1 \times Y_2$, $E_0 \times Y_2$ and $E_0 \times Y_1$ provide three additional groups of inequalities. It should be noted that the membership of a row to E_0 is determined by the complementary set $E_1 \cup E_2$.

2-2.1 Problem formulation in form of integer programming problem

To summarize, the search problem subject to the restrictions shown in Table 4 becomes a problem of calculating all of the solutions of the following integer programming problem:

Decision variables:

$C_1 = \left(c_1^1, c_2^1, c_3^1, ...c_n^1\right), C_2 = \left(c_1^2, c_2^2, c_3^2, ...c_n^2\right), R_1 = \left(r_1^1, r_2^1, r_3^1, ...r_m^1\right), R_2 = \left(r_1^2, r_2^2, r_3^2, ...r_m^2\right).$

Input parameter: matrix Z

Objective function:

$$Maximise \sum_{i=1,2} R_i Z C_i^T - \sum_{i=1,2} N_i M_i \tag{4}$$

subject to

$$N_1 - \left(\sum_{j:c_j^1=1, c_j^1 \in C_1} z_{ij} \right) \geq 1, \forall i \in \left\{ i \mid r_i^2 = 1, r_i^2 \in R_2 \right\} \tag{5}$$

$$N_2 - \left(\sum_{j:c_j^2=1, c_j^2 \in C_2} z_{ij} \right) \geq 1, \forall i \in \left\{ i \mid r_i^1 = 1, r_i^1 \in R_1 \right\} \tag{6}$$

$$N_1 - \left(\sum_{j:c_j^1=1, c_j^1 \in C_1} z_{ij} \right) \geq 1, \forall i \in \left\{ i \mid r_i^2 = 0, r_i^2 \in R_2, r_i^1 = 0, r_i^1 \in R_1 \right\} \tag{7}$$

$$N_2 - \left(\sum_{j:c_j^2=1, c_j^2 \in C_2} z_{ij} \right) \geq 1, \forall i \in \left\{ i \mid r_i^2 = 0, r_i^2 \in R_2, r_i^1 = 0, r_i^1 \in R_1 \right\} \tag{8}$$

$$R_1 + R_2 \leq (1,1,...,1)_M \tag{9}$$

$$C_1 + C_2 \leq (1,1,...,1)_N \tag{10}$$

$$c_i^j = 0,1; r_i^j = 0,1 \tag{11}$$

$$C_1, C_2, R_1, R_2 \neq 0 \tag{12}$$

This BIP is nonlinear because the object function is a quadratic polynomial. However, because of the characteristics of our original problem, we can exhaustively search entire solution by dividing it into two sub-problems, thereby utilizing the concept of Divide and Conquer in [52].

2-2.2 Relaxation of binary programming problem into sub-problems

The binary programming problem described above cannot be solved by ordinary methods. The search space will grow exponentially based on the parameters N and M. Therefore, the problem cannot be solved by brute force searching. However, our goal is to obtain all solutions that meet the maximum number of conditions. Thus,

the basic approach to linear BIPs, such as branch and bound or branch and cut, are invalid. This gives rise to the design of a new algorithm for solving this problem.

According to the observations made above for the BIP, the restrictions of variables (C_1, R_1), and (C_2, R_2) are symmetrical. The result must then be the same by exchanging Y_1, Y_2 and E_1, E_2. Thus, we want to simplify the binary problem to a set of sub-problems where the merging provides the solution to the original problem. The generic search sub-problem is illustrated in Figure 10:

	Y_1	Y_0
E_1	$\forall(i,j)\in E_1\times Y_1; z_{ij}=1;$	
E_0	$\forall i\in E_0, \exists j\in Y_1; z_{ij}=0;$	

Figure 10: Sub-problem blocks in matrix Z

We use the vectors $R_1=(r_1,r_2,...,r_M)_M$, and $C_1=(c_1,c_2,c_3,...c_N)_N$ to denote Y_1 and E_1 in Figure 10. If the vector component is equal to 1, the corresponding column or row belongs to Y_1 or E_1.

For convenience, we assume that $M_1=\left(\sum_j r_j\right)$ and that $N_1=\left(\sum_j c_j\right)$. Our goal is to seek the block expressed by (E_1, Y_1) in which all of the elements in the block are equal to 1.

We can use the same approach used previously to define the object function: $R_1ZC_1^T-M_1N_1$, where the maximum value of which is zero. Moreover, its accompanying block $(\overline{E}, Y)$ underneath has at least one element equal to 0 for each row, which can be formulated by the inequalities: $N_1-\left(\sum_{j:c_j=1, c_j\in C_1} z_{ij}\right)\geq 1$, $i\in\{i\mid r_i=1, r_i\in R_1\}$.

Decision variables:

$R_1=(r_1,r_2,...,r_M)_M, C_1=(c_1,c_2,c_3,...c_N)_N$

Input parameter: matrix Z;

Maximize: $R_1 Z C_1^T - M_1 N_1$ (13)

Subject to:

$$N_1 - \left(\sum_{j:c_j=1, c_j \in C_1} z_{ij} \right) \geq 1 \, , i \in \{ i \mid r_i = 0, r_i \in R_1 \} \quad (14)$$

$$r_i = 0,1; c_i = 0,1 \quad (15)$$

$R_1 \neq \phi; \; C_1 \neq \phi$

After the results of the sub-problems are gathered, it is necessary to combine the blocks to solve the original problem. The combinatorial strategy is to examine all combinations of the two results of the sub-problems based on the following rule: If the intersection of their row sets and column sets are empty ($E_1 \cap E_2 = \phi$, $Y_1 \cap Y_2 = \phi$); then the blocks (E_1, Y_1) and (E_2, Y_2) constitute a solution to the original problem. The correctness of the strategy is based on the assumption that if the solution sets of two problems include each other, they are equal.

2-2.3 Problem resolution by exhaustive search algorithm

2-2.3.1 Search algorithm for the sub-problem

We now describe the method for addressing the sub-problem. The pattern can be searched along the columns and rows. As our problem has a restriction regarding $\overline{E}$, we search the blocks row by row.

The searching procedure consists of five steps:

1. Let i =1.
2. Select the columns' set, the elements of which equal 1 in row i.
3. Select one subset of the columns' set as defined in Step 2. If the column set was previously selected in another row, reselect the subset. If not, mark the column set as having been checked.
4. Store this column set as the block column set, and add row number i to the row set of the block. Look up the row k, k>i, if it has the same column set in which elements equal 1, add the row number in the block row set.

5. Output block as one solution: if the column set in row i does not finish traversing, return to Step 3. If i=M, the algorithm terminates; otherwise increment i by 1, and return to Step 2.

Another problem that must be addressed in Step 3 is traversing the subsets of the subsets. The problem is essentially a Generating All n-Tuples Problem. Equivalently, we want to visit all n-Tuples $(y_1, y_2, ...y_n)$ where each $y_i = 0 \text{ or } 1$. One can find more details in reference [53].

In the following section, an explanation is provided why the search algorithm can obtain all of the solutions of the sub-problem. First, we prove that the blocks are solutions of the sub-problem. It is explicit that the elements in a block are equal to 1; thus, they meet the requirements of the objective function. Equivalently, it is clear that each row in block $(\overline{E}, Y)$ has at least one element equal to 1. If not, the row whose elements are equal to 1 will be added to the block row set according to the algorithm. Second, we prove that the algorithm can gather all of the solutions. Suppose that there exists a block that is the solution of the sub-problem. Thus, all of its elements are equal to 1 to meet the requirements of the objective function, and each row in the block $(\overline{E}, Y)$ has at least one element equal 1 to satisfy the restriction. The block's row number set must have its smallest element ranked above the others; when the algorithm operates on this row, the block column set pertains to this row column set, as long as it has not been previously checked. Thus, this algorithm yields all of the solutions.

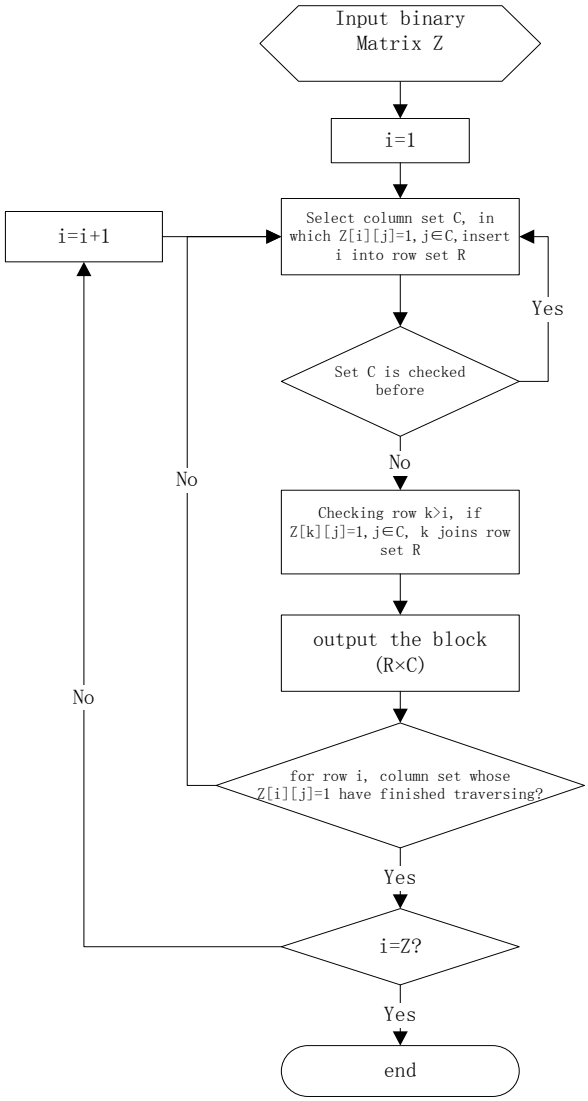


Figure 11: Flow chart of algorithm

2-2.3.2 Time and space complexity analysis

We presume that the maximum number of elements equal to 1 in each row is n and the row number is M; thus, the time consumed by visiting all of the possible combinations of elements equal to 1 is $O\left(M \times 2^n\right)$. Moreover, we must scan the remaining rows (fewer than M rows remain). Therefore, the time complexity in the searching phase is $O\left(M \times M \times 2^n\right)$. Suppose that the number of blocks in the searching phase is S; thus, time complexity in the comparing phase is $O\left(\frac{S(S-1)}{2}\right) = O\left(S^2\right)$. In conclusion, the time complexity is $O\left(M^2 \times 2^n + S^2\right)$.The

principal space cost is related to the storage of the blocks in the searching step. Because the number of blocks is lower than 2^N, the space complexity is $O(2^N)$.

2-2.3.3 Problem solution by Binary Programming algorithm

The objective function (4) is a quadratic function, it involves the items like $r_i^1 c_j^1$ or $r_i^2 c_j^2$. In fact, as [51] introduced, a group of new binary variables x_{ij}^1 are introduced to replace $r_i^1 c_j^1$ such that the binary values correspond the value of $r_i^1 c_j^1$.

Moreover, two linear constraints are added to constrain the value of x_{ij}^1 by $r_i^1 \text{and } c_j^1$:

$$2x_{ij}^1 \leq r_i^1 + c_j^1 \leq x_{ij}^1 + 1 \quad (16)$$

From Table 6, we are aware that with the constrain of the Equation (16), then $x_{ij}^1 = r_i^1 c_j^1$.

Table 6: Truth table for $x_{ij}^1, r_i^1 and\ c_j^1$

r_i^1	c_j^1	$x_{ij}^1 = r_i^1 c_j^1$	$2x_{ij}^1$	$r_i^1 + c_j^1$	$x_{ij}^1 + 1$
0	0	0	0	0	1
0	1	0	0	1	1
1	0	0	0	1	1
1	1	1	2	2	0

We can perform the same transformation for $r_i^2 c_j^2$. Subsequently, our problem has become a binary programming problem and can be solved by the use of a branch-and-cut algorithm [51]. However, at certain times there are requirements for the number of evaluation parameters involved within the GTC. As discussed in [33], a proposed GTC selection principle 'minimize the cardinality of Y_0'. Thus an equation can be added in the constraints to restrict the number of evaluation parameters:

$$\sum c_j^1 + \sum c_j^2 \geq K \quad (17)$$

K is the minimum number of evaluation parameter got involved with the GTC. Thereby, the transformed linear binary programming can be shown as:

Decision variables:

$$C_1 = \left(c_1^1, c_2^1, c_3^1, \ldots c_n^1\right), C_2 = \left(c_1^2, c_2^2, c_3^2, \ldots c_n^2\right), R_1 = \left(r_1^1, r_2^1, r_3^1, \ldots r_m^1\right), R_2 = \left(r_1^2, r_2^2, r_3^2, \ldots r_m^2\right).$$

Input parameter: matrix $Z(z_{ij})$,

Objective function:

$$Maximise \sum_{i,j}(z_{ij}-1)x_{ij}^{1}+(z_{ij}-1)x_{ij}^{2} \quad (18)$$

subject to

$$2x_{ij}^{1,2} \le r_{i}^{1,2}+c_{j}^{1,2} \le x_{ij}^{1,2} \text{ for all } i<M,\ j<N; \quad (19)$$

(5)-(12) and (17).

In the objective function, we can know that the number of new binary variables (x_{ij}^{1} ,x_{ij}^{2}) is the number of '1' in matrix Z. This solution is complementary when exhaustive searching is unavailable, as the branch-and-cut method has solved an instance containing as many as 6000 binary variables [51].

2-3 Illustration of the algorithm for the case of the electrical circuit breaker

Let us consider a simple technical system such as an electrical circuit breaker whose components are presented in Figure 12 as a means to illustrate the algorithm on the real case. When an overload occurs, the overload creates a force because of magnets and electrical field, which operates a piece called the firing pin. The firing pin opens the circuit by pressing the switch, located in the circuit breaker. In the event of high overload, the plastic stem firing pin breaks without opening the switch and cannot be reused.

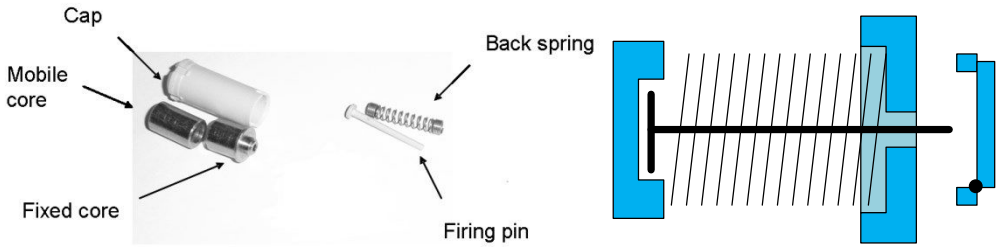


Figure 12: Components of electrical circuit breaker and the scheme

The problem has been studied and the main system parameters and their domains have been defined as action parameters:

x_1: firing pin material (plastic – 1, metal – 0) ;

x_2: core internal diameter (high – 1, low – 0) ;

x_3: core external diameter (high – 1, low – 0) ;

x_4: firing pin diameter (high – 1, low – 0) ;

x_5: spring straightness (high – 2, medium – 1, low – 0) ;

and the results are defined as the evaluation parameters as follows:

y_1: circuit breaker disrepair (satisfied – 1, unsatisfied – 0) ;

y_2: circuit breaker reusability (satisfied – 1, unsatisfied – 0) ;

y_3: spring core mounting (satisfied – 1, unsatisfied – 0) ;

y_4: firing pin bobbin mounting (satisfied – 1, unsatisfied – 0) ;

y_5: normal mode release (satisfied – 1, unsatisfied – 0) ;

y_6: firing pin initial position return (satisfied – 1, unsatisfied – 0).

The system behavior was modelled through the experimental table and it is shown in Table 7. The objectives that have been established to build this experimental table are as follows:

- The satisfaction of at least one evaluation parameter in each experiment;
- Each of the action parameters has at least one time each of its possible values;
- To minimize the number of experiments.

Even if the assumption is not totally consistent, the action parameters have been considered independent within the limits of their defined domains.

Table 7: Experimental table for the circuit breaker

	x1	x2	x3	x4	x5	y1	y2	y3	y4	y5	y6
e1	1	1	0	0	1	1	0	1	1	1	1
e2	0	1	1	1	1	0	1	0	0	1	1
e3	1	0	1	0	0	1	0	1	0	0	0
e4	1	1	0	0	0	1	1	1	1	0	0
e5	1	0	1	0	1	1	0	1	0	1	1
e6	0	1	0	1	2	0	1	0	1	1	1
e7	1	0	1	1	0	1	0	1	0	0	0
e8	1	0	0	0	1	1	0	0	1	1	1
e9	0	1	0	0	2	0	1	0	1	1	1

First evidence is that no solution can be found in the defined Table 3, as no experiment enables the satisfaction of all the evaluation parameters. Additionally, looking for generalized system of contradictions in such a table could lead to several ones, at least one per evaluation parameter, as soon as each evaluation parameter is at least satisfied once.

Assume it is that the choice of action parameters is performed in such a way that each evaluation parameter will be satisfied in one experiment. It is also assumed that no solution is found in the table; therefore, each evaluation parameter will have at

least one experiment in which it will be satisfied and one experiment in which it will not be satisfied. Thus, a contradiction could be formulated for each of the evaluation parameters. However, the generalized system of contradictions also enables the formulation of more complex generalized system of contradictions, thereby implying two combinations of evaluation parameters. Thus, a set of generalized system of contradictions can be formulated for a design of experiment without a solution. An example of non-classical contradiction and a GTC is illustrated in Figure 13: When the first concept of evaluation parameters “y1 and y3 fit requirements” is satisfied, the second concept “y2 and y5 fit requirements” is not satisfied and vice versa.

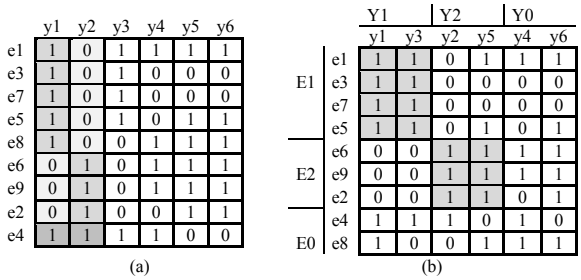


Figure 13: Example of false classical TC (a) and GTC (b)

The first set of experiments concerned the search of generalized technical contradictions in the case of an electrical circuit breaker (Figure 14). We identified 117 generalized technical contradictions within the 341 possible pairs of concepts. As an example, we discuss four different GTCs, which can be interpreted as follows.

GTC #1 represents the contradiction between the circuit breaker reusability y2 and both the spring core mounting y3 and normal mode release y5. GTC #2 represents the contradiction between the following pairs of parameters: the circuit breaker reusability y2 and normal mode release y5 and the couple circuit breaker disrepair y1 and spring core mounting y3. GTC #3 represents the contradiction between the couple circuit breaker reusability y2 and firing pin initial position return y6 and the couple spring core mounting y3 and normal mode release y5. Finally, GTC #4 represents the contradiction between the couple circuit breaker reusability y2 and firing pin initial position return y6 and the rest of the evaluation parameters.

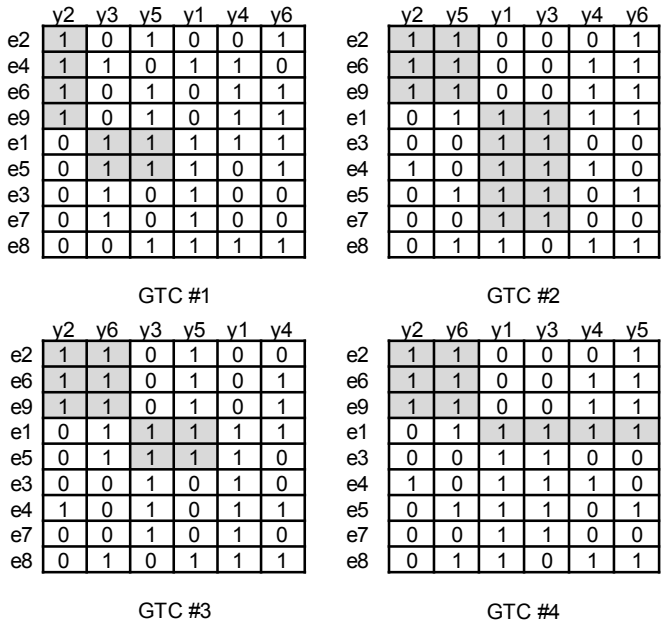


Figure 14: Examples of GTCs in the case of electrical circuit breaker

2-4 Exploitation of the algorithm in order to answer the questions

In this paragraph, we examine a sample of matrices with the same number of rows and columns as the matrix considered in the electrical circuit breaker example. However, different numbers of evaluation parameters equal to 1 are used to answer the questions posed in our problematic section.

The first question in our research methodology is when no classical technical or physical contradiction is available. Do the generalized technical and generalized physical contradictions exist? Are they numerous? Can we always extract generalized contradictions from the behavioral representation and the objectives of the system intrinsically? What are the consequences?

In the previous paragraphs we used a circuit breaker example to illustrate the various concepts of technical contradictions. The algorithm used in this example assists in measuring the number of technical contradictions as well as generalized technical contradictions for the real case. As there are no classical TCs in the case of the electrical circuit breaker that prompt us to further study the question when there are no TCs; the question then becomes how many GTCs are there? There can be several

factors influencing the number of TCs and GTCs. We will perform experiments to discuss factors such as number of columns, number of rows; and density of ones (the number of 1 in evaluation parameter matrix), which are supposed to affect the number of TCs and GTCs. An obvious factor is the density of ones in the matrix. We randomly generated a sample of 377 matrices with the same number of rows and columns as the matrix considered in the electrical circuit breaker example. However, with different numbers of evaluation parameters equal to 1–9 rows and 6 columns with no column full of zeros and no row full of ones. This study was performed to confirm the real case results and to measure and compare different possible but quite similar cases. Our study determined the complexity of such a search based on a relatively small number of technical contradictions versus a large quantity of discovered generalized technical contradictions that exists. Figure 15 illustrates the evolution of the number of TCs/GTCs with the density of ones. Some trends can be observed: the number of TCs decreases with the density and tends to zero at a density of up to 50%. The average number of GTCs increases up to a density of 50% and then decreases. The mean values of each population are correlated to the general trend. The points indicated by a star point in Figure 15 represent the real characteristics of the electrical circuit breaker matrix. The results confirm that this example is not an exception.

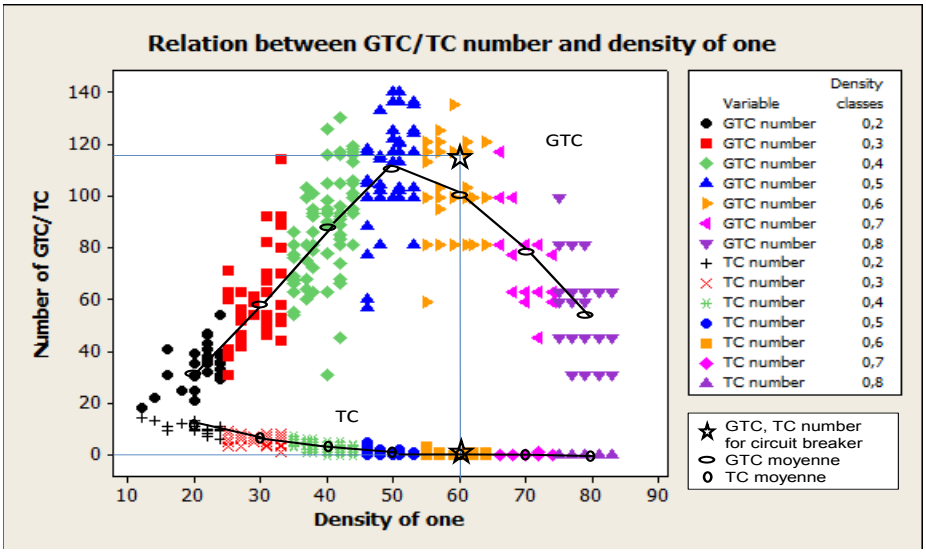


Figure 15: TC/ GTC number versus density for general matrices

To study the dispersion of the GTC and TC number in greater detail, the two box plots shown in Figure 16 were created. The plots show the interval extent, the interquartile box, the linked mean values, and the abnormal points for each population.

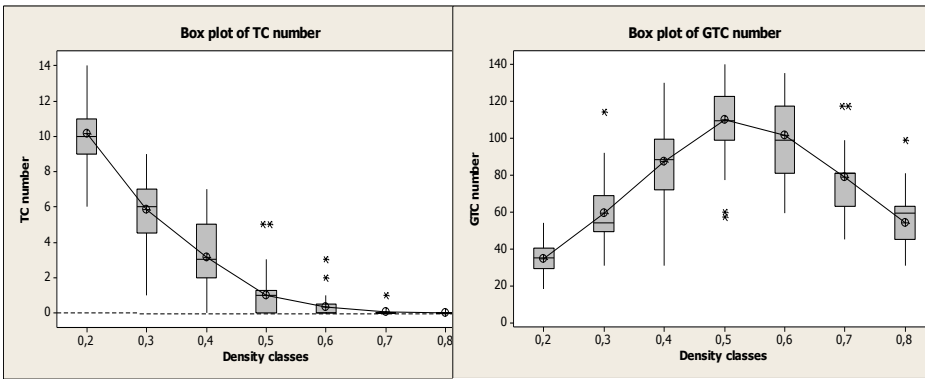


Figure 16: Relation between GTC/TC number and density of ones

The results prove the importance of the density of ones in the number of TC and GTC. However, the other two factors of matrix; the number of the columns and the number of the rows could allow for generalization of the conclusions to the other case.

2-4.1 Factors affecting GTC and TC number: generalization

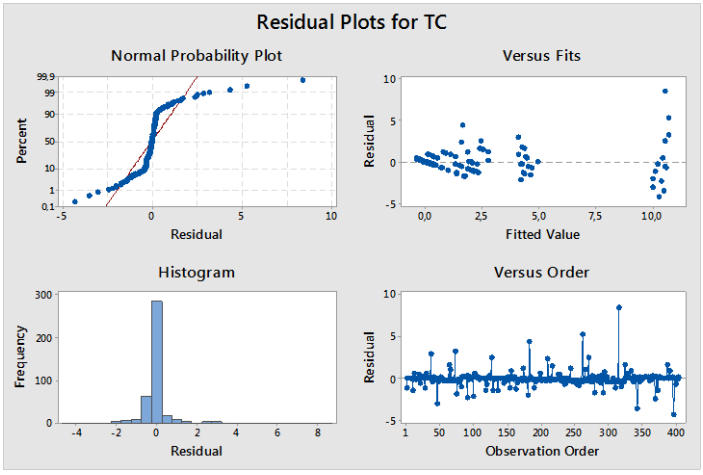
As mentioned before, the results of our algorithm indicate the complexity required during the search for technical and generalized technical contradictions. In some conditions, there may not be a technical contradiction, thereby preventing the solution of the inventive problem or overcoming the fact of no existing technical contradictions. On the contrary, there may be a significant number of generalized technical contradictions preventing the human expert from dealing with numerous possibilities. From this the choice of our future work is the best set of generalized technical contradictions to solve the inventive problem. The important factors influencing the number of technical and generalized technical contradictions in order to reduce this complexity are then identified.

This section reveals how the number of columns (parameters) and rows (experiments) and the density of ones (number of satisfied parameters) affect the number of found TCs and GTCs. We generated several populations of random

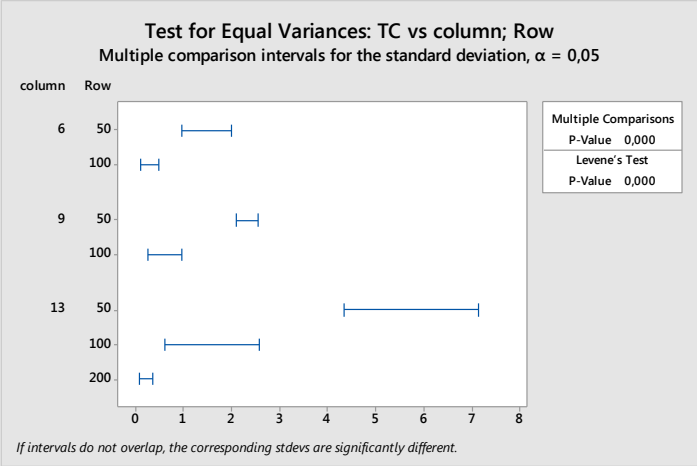
matrices according to the three previously mentioned influencing factors. Each factor has three levels (i.e., 6, 9, and 13 columns—50, 100, and 200 rows—20%, 50%, and 80% density). Finally, for each combination, 15 matrices with no line full of ones and no column full of zeros were created randomly. Overall, the sample contained a total of 405 different matrices.

2-4.1.1 Factors affecting the number of TCs in the general population

To identify the main factors affecting the number of TCs, we used the general complete factorial design of experiments and variance analysis [54]. Three factors were studied, i.e., the column number, the row number, and the density of ones in the matrix. The results obtained for the design of experiments revealed three different populations and thus three Henry's curves (Figure 17a). Each population should be treated independently because the populations do not show similar behavior as their variances are not the same as shown in Figure 17b. There is a large variance in the number of TCs for the matrices with a small number of rows. The variance in the number of TCs also increased with the number of columns. The variance in the density was not studied because the technical contradictions were identified only for the case of 20% density and the number of TCs varied from 0 to 19. The matrices with 200 rows and the matrices with densities greater than 50% generally did not feature any TCs



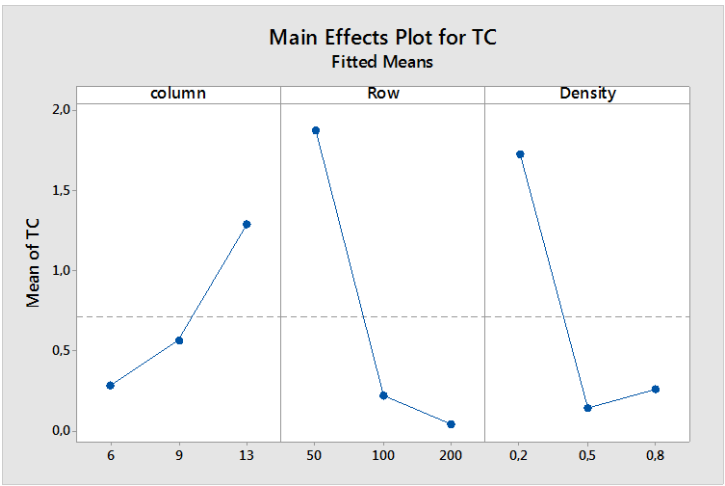
(a)



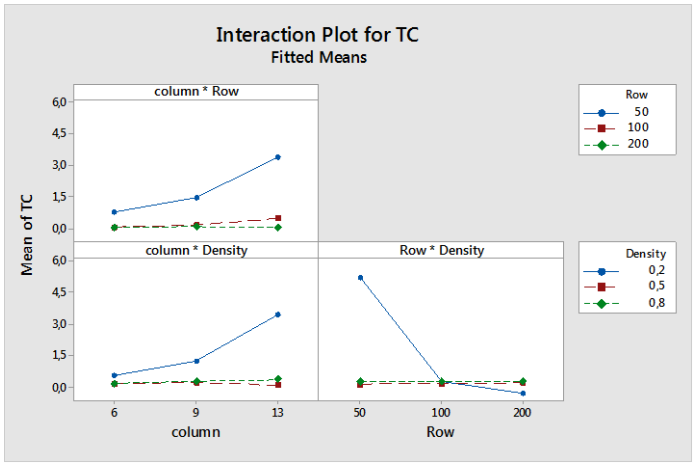
(b)

Figure 17: Residual values (a) and test of variance (b) for TC number

With the aid of an interaction diagram, a graphical representation of the main effects, and the ANOVA table, different hypotheses were tested for each factor. As shown in Figure 18a, the column number has a positive factual effect on the number of possible technical contradictions. However, the column number is correlated with the negative effect of the number of rows and the density. This is especially noticeable for the matrices with fewer than 100 rows and a density less than 50%. At densities greater than 50%, the number of technical contradictions is always zero; thus, other effects are counteracted against. These results confirm intuition and generalize the average limiting zone for the existence of TCs.



(a)

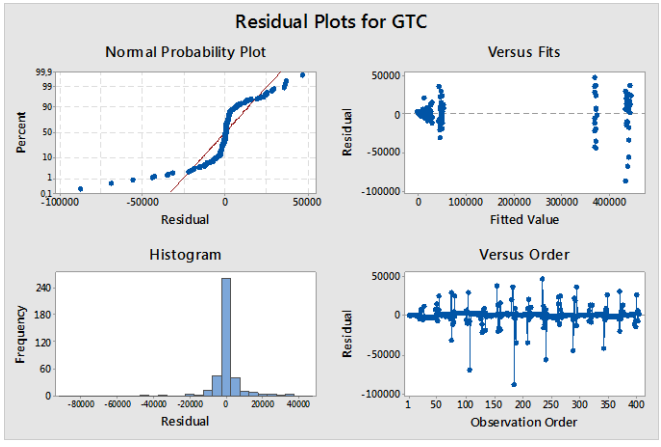


(b)

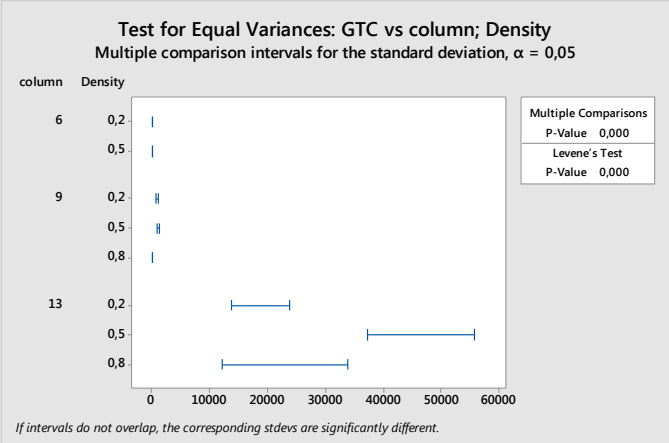
Figure 18: Principal effects of factors (a) and their interactions (b) for TC

2-4.1.2 Factors affecting the number of GTCs in the general population

We identified the main influencing factors for the number of GTCs by using the same procedure followed for the TC case. The general complete factorial design of experiments and variance analysis were used to study three factors: the column number, the row number, and the density of ones in the matrices. As in the case of the TCs, three different populations and thus three Henry's curves (Figure 19a) were detected. Their variances were compared by the test of variance equality, as shown in Figure 19b. The variance in the number of GTCs increases with the number of columns, whereas the variance does not change with row number. The variance is smaller at densities of 20% and 80% but large for a density of 50%.



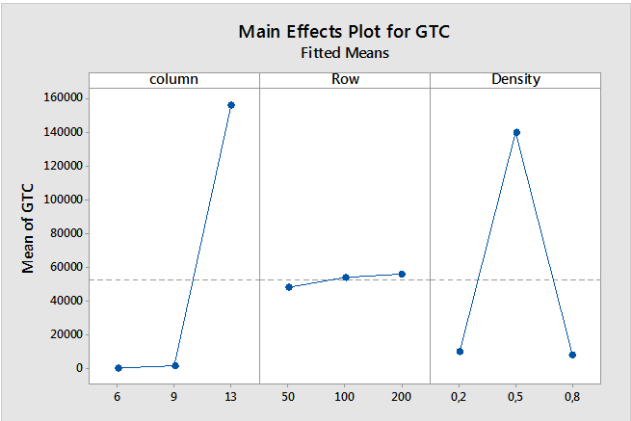
(a)



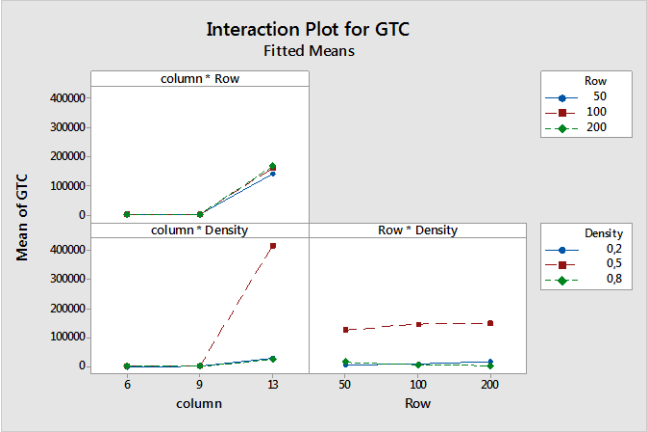
(b)

Figure 19: Residual values (a) and test of variance (b) for GTC

The variance tests confirm the results of the main effect analysis and interaction diagram shown in Figure 20. As shown, the number of rows does not have a significant effect on the number of GTCs. The most positive significant effect is the number of columns. The next level of significant effects is the density and the interaction between the density and the number of columns. It appears that the shape of the number of the GTCs versus the density curve does not change with the number of columns and rows. Instead it increases up to a density of 50% and then decreases thereafter. This behavior will have an impact on our future strategy for filtering for GTCs.



(a)

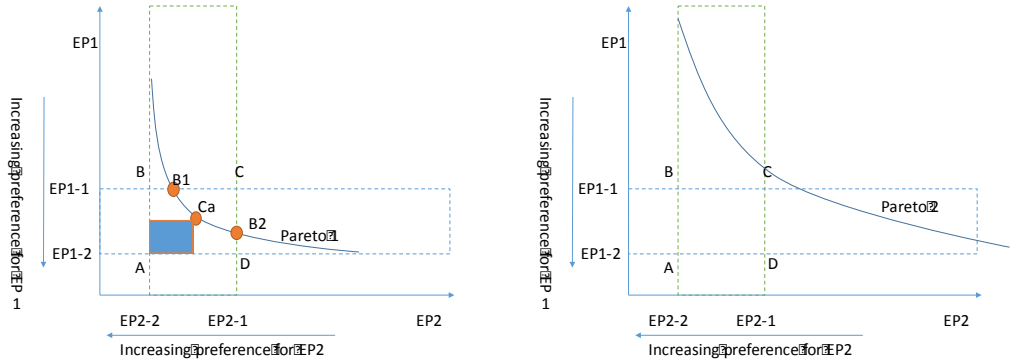


(b)

Figure 20: Principal effects of factors (a) and their interactions (b) for GTC

2-4.2 Comparison of optimization and inventive design results

In the previous section, we have been aware that the column and density of ones plays an important role in the number of GTCs. In this paragraph, our goal is to compare different models of contradictions according to the optimization principles, the Pareto front and dominance relation, to show their limits and possible solving issues. Two situations of optimization vs innovation are illustrated in [55]. In Figure 21, there are two evaluation parameters EP1 and EP2. The most desirable values for evaluation parameters are EP1-2 and EP2-2. The acceptable individual alternatives for EP1 and EP2 are within range [EP1-1, EP1-2] and [EP2-1, EP2-2], respectively. The first situation is shown in Figure 21(a), where the Pareto line has partially dominated the region 'ABCD', and there is a consensual agreement at point Ca. The zone in the hatched zone of 'ABCD' presents the solution space of innovation activities. In this situation, the binary value in the experimental table can be [1, 1], [1, 0], [0, 1] and [0, 0] in the compromise point of view. However, in Figure 21(b) the Pareto does not dominate the zone 'ABCD', which is the solution obtained via innovation activities for situation 2. The binary value of EP1 and EP2 in the experimental table can only be [0, 0].



(a) (b)

Figure 21: Optimization vs innovation

From the above analysis, we can see that the acceptable zone and compromise solution could be a factor of how many ones exist in the experimental table. The acceptable zone and compromise solution also affect the density of ones in the table. Hence, two things affect the GTC from an optimization point of view. The first is an acceptable range, which reflects the negotiation degree of the most desirable value. The second is the Pareto line itself as the point of the most desirable value is located on the Pareto line; therefore, the mutual satisfied evaluation parameters are from the Pareto line dominating the compromise zone.

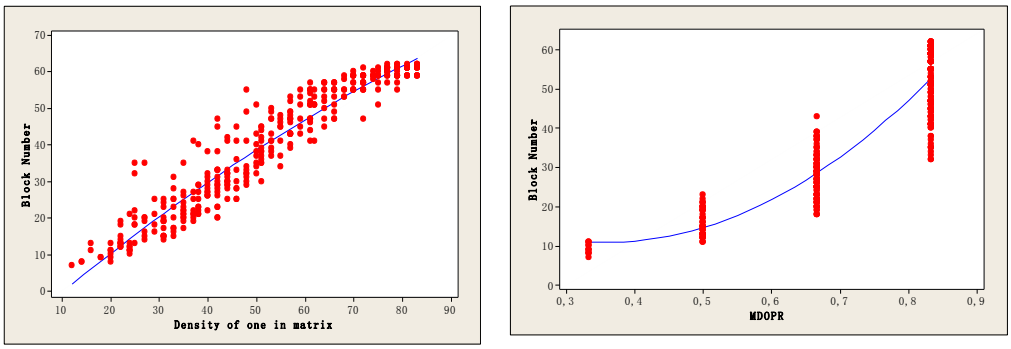
For the binary level, the binarized Pareto is the decisive factor. Additionally, the GTC set from binary Pareto is the same from the whole experiment set based on the study of the process of exhaustive GTC searching algorithm. When we enumerate the column set; if the non Pareto binary result cannot provide the new column set than the Pareto is dominating the column set. This means that we only need the binary Pareto to collect the GTC set. In 2-2.3.2, decreasing the row number cannot reduce computational complexity from a theoretical point of view. However, for the binary programming solution in 2-2.3.3, the binary Pareto decreases the number of variables. Thus, in practical terms, the binary Pareto is able to effectively save computational time.

2-4.3 Time consumption of the algorithm

To estimate the algorithm's time consumption for a given matrix, we consider that the block number mainly affects the searching time. In the following, we analyze the interaction between the block number, the density of ones, and the maximum density of ones per row for the general population of matrices similar to the matrix for the electrical circuit breaker.

2-4.3.1 Relationship between block number and density of ones in the case of the electrical circuit breaker

For the case of the electrical circuit breaker, we studied the relationship between the number of blocks, which indicates the time consumption of the algorithm and the density of ones in the matrix. Figure 22 a) indicates that the number of blocks, and thus, the time consumption increases nearly linearly with the density of ones by monitoring the mean values. In Figure 22 b), the number of blocks is plotted as a function of the maximum density of ones per row in the matrix. For convenience, we use MDOPR to denote the maximum density of ones per row. In summary, for a given matrix similar to that of the electrical circuit breaker, the most influential factor affecting the algorithm's time consumption is the block number. Moreover, the results of the experiments indicate that the time consumption is approximately proportional to the density of ones in the matrix.



(a)

Block number versus density of ones

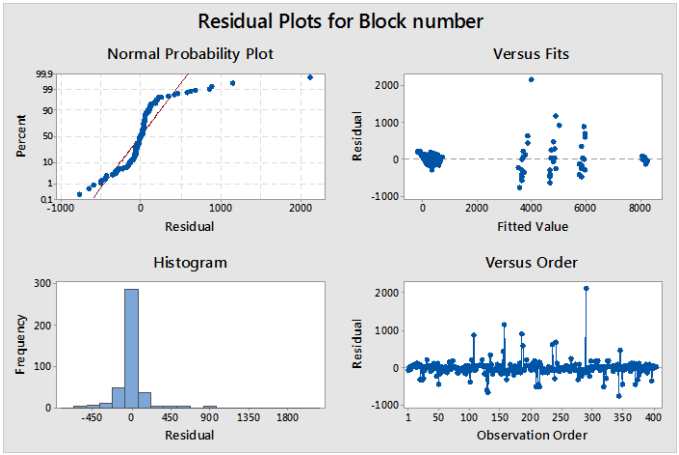
(b)

Block number versus maximum density of ones per row

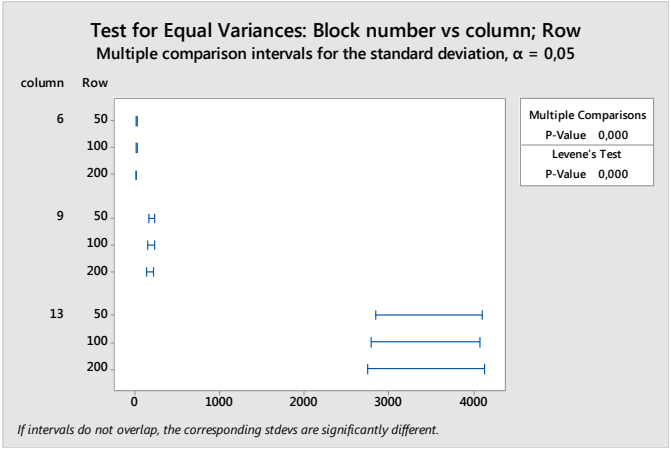
Figure 22: Scatter plots of block numbers

2-4.3.2 Relationship between block number and NC/NR/density in the general population

We identified the main influencing factors for the number of blocks in the same manner as that for TCs and GTCs. The general complete factorial design of experiments and variance analysis were used to study three factors: the column number, the row number, and the density of ones. Three different populations and thus three Henry's curves (Figure 23a) were detected; the variances were compared by the test of variance equality, as shown in Figure 23b. The variance in the number of the block increases with the number of columns. The variance does not change with different row numbers and only slightly increases based on the density of the ones.



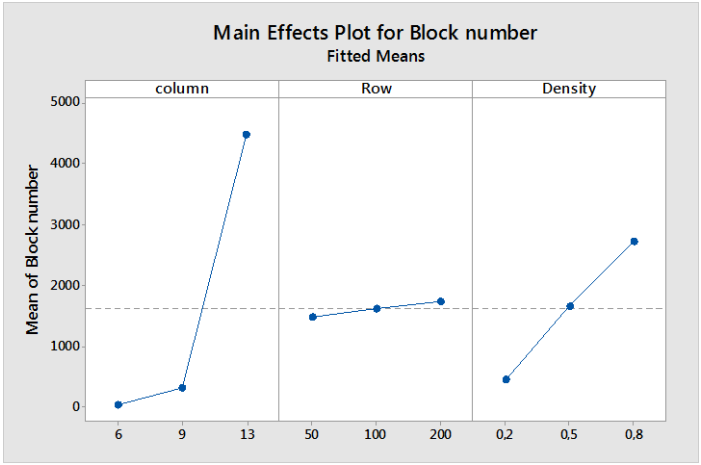
(a)



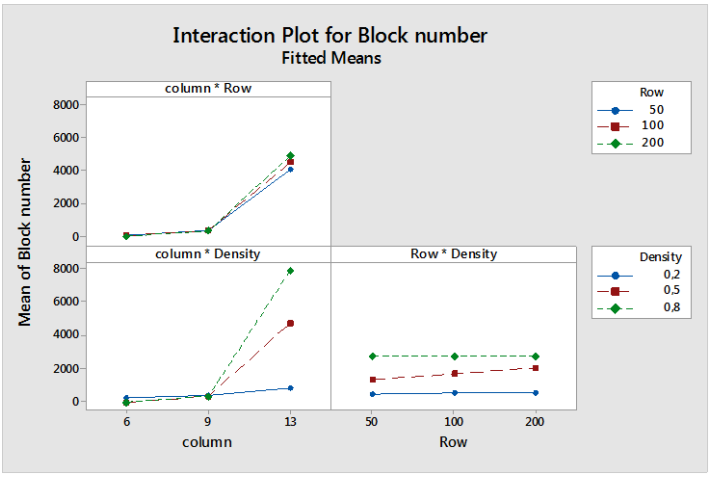
(b)

Figure 23: Residual values (a) and test of variance (b) for block number

We cannot conclude that the number of rows has a significant effect on the block number. The significant positive effects are the number of columns and the density. The interaction between the density and the number of columns also has a significant effect.



(a)



(b)

Figure 24: Principal effects of factors (a) and their interactions (b) for block number

2-4.3.3 Large-scale matrices and time consumption of our algorithm

This part of the study concerns large-scale matrices and the relation between the number of satisfied evaluation parameters (equal to 1) and the time cost of the algorithm. We used matrices in which each element is equal to 1 owing to the dominating effect of the number of evaluation parameters whose values equal 1 on

the time complexity. We performed the test for five large-scale matrices and present the time consumption —the matching time and searching time— in Table 8. The test was performed on five matrices containing only ones as the elements and with an increasing number of evaluation parameters (columns). The number of matrix blocks is proportional to the maximum of satisfied parameters in each row.

Table 8: Experimental results

	Num of Columns	Matching time(s)	Num of Blocks	Searching time(s)
Data1	5	0.007	31	0.190
Data2	8	0.240	255	0.197
Data3	10	3.632	1023	0.234
Data4	12	50.159	4095	0.401
Data5	15	3666.167	32767	2.283

Data 1 has 5 evaluation parameters (columns) and 1500 experiments (rows). Data 2 has 8 evaluation parameters and 1500 experiments. Data 3 has 10 evaluation parameters and 1500 experiments. Data 4 has 12 evaluation parameters and 1500 experiments. Data 5 has 15 evaluation parameters and 1500 experiments. The computer used to run the tests was 64-bit machine with 8G of memory and a dual-core CPU operating at 3.4 GHz.

From Table 8, we draw the following conclusion: the time cost grows exponentially with the number of evaluation parameters equal to 1; the time consumed in the searching block is less than that consumed in the matching block. Thus, in these experiments, the time cost is determined by the matching phase. Because we need to store all of the blocks in memory during the matching phase, the space consumption is beyond the capability of the experimental computer (8G) when the number of evaluation parameters is greater than 15 and the number of experiments is equal to 1500.

2-5 Discussion

The results of the experiments from this chapter indicate a greater the number of possible combinations are possible when the number of technical parameters and experiments describing the technical system increases. Thus, a human expert is not able to identify and consider all of the combinations, which is why we attempted to automate the search for generalized technical contradictions. The method is reliable

when the maximum number of evaluation parameters is less than 15. The matching time is the main factor affecting the algorithm's speed. Thus, to improve the algorithm, we worked to find some restriction for discarding blocks before the matching phase. This problem can also be classified as a data mining problem and is similar to seriation and matrix reordering problems; such as the two-mode clustering problem. However, these methods do not propose to find all possible solutions. The exhaustive algorithm answers Q1 and Q2 about the number of TC and GTC and how to extract them, and it provides a powerful tool for extracting GTCs when an evaluation parameter is less than 15. Moreover, a linear binary programming model is proposed, which provides an opportunity to search a characterized GTC based on the searching strategy in constraint. This model positively answers the Q7 in terms of GTC identification using optimization methods, and provides a foundation for answering Q5 (searching relevant contradictions without exhaustive search) if we know and can transform the principle of GTC selection into BIP constraints. In addition, section 2-4.2 discusses the GTC in the binary Pareto is the same as the original experimental table. This property can help us effectively reduce the computation complexity in solving linear binary programming problem.

Because of the algorithm, it becomes possible to quantitatively evaluate the volume of contradictions and check the hypothesis about rarity of TC existence. In the experimental section of this paper, the results of our algorithm for the case of an electrical circuit breaker were provided. These results confirm the large number of GTCs. Secondly an analysis of the number of GTCs and TCs versus the density of ones for the population of binary matrices was performed with the same number of rows and columns than the circuit breaker. The results show that the average number of GTCs increases with density until 50% and then decreases, while the number of TCs decreases with density and is generally equal to zero after 60%. The values of TCs and GTCs of the circuit breaker example are consistent with the results of this general population of matrices. In order to generalize previous results, we evaluated the impact of the number of columns, density and rows on the number of GTCs and TCs. The analysis of the results shows that, for a given number of rows and columns, the average value of GTCs and TCs evolve in the same way as for the circuit breaker. The average number of GTCs increases with density until 50% and then decreases, while

the number of TCs decreases with density and is generally equal to zero after 60%. Nevertheless, the number of columns acts as a shape factor. For instance, the maximum value of GTCs, which is obtained for density equal to 50%, increases exponentially with the number of columns. The main factors affecting the number of technical contradictions and generalized technical contradictions was identified with the aid of the principles of the design of experiments and statistical analysis tests. These results will have an impact on our strategy for solving inventive design problems and for selecting the most appropriate technical or generalized technical contradictions to solve such problems. Additionally, the relation between the number of contradictions and the experimental table could provide the answer to Q1, such as when the density of ones is more than 60%; there is no TC; and number of GTC is significantly greater than number of TCs.

Once all generalized technical contradictions are identified, several elimination strategies are subsequently needed to choose only the GTCs leading to resolution of the design problem. Two different criteria were proposed in [33] to select the most meaningful generalized technical contradictions. Nevertheless, generalized technical contradictions provide only a partial solution in TRIZ problem solving. To resolve the contradictions, more details are needed regarding the system parameters and the relationships thereof. The next chapter discusses the identification of physical contradictions according to the identified GTCs.

Chapter 3 Extraction of generalized physical contradiction

In this chapter, we introduce a theoretical problem for the extraction of generalized physical contradictions and practical solving principles to obtain the exhaustive search algorithm. In the second part of this chapter, the GPC extraction problem is formulated into a binary integer programming model. The generalized technical contradiction divides the whole set of experiments E into three subsets: a subset satisfying the first set of evaluation parameters E1; the second subset satisfying the set of evaluation parameters E2; and the third subset complete the previous two evaluation parameters, E0. The search of each concept of generalized physical contradiction is first defined as the search for discriminating concepts for the sets of experiments as defined by generalized technical contradiction. To search for CONCEPT 1 means to search a function that recognizes the elements from E1 and eliminates the elements from E2 U E0. To search for concept 2 refers to searching for a function that recognizes the elements from E2 and eliminates the elements from E1 U E0. The research binary integer program is defined and the proprieties are used to solve the function. These contradictions can be numerous and impossible to list. Hence, we indicate that one specific generalized physical contradiction could continue all possible generalized physical contradictions. The search of this specific contradiction could be performed by adding the objective function to the set of equations in the binary program; therefore, we will solve for the binary integer optimization problem. The exhaustive search algorithm is presented in the third part and solved by the enumeration and removal of different states. Then we illustrate the problem formulation and the search algorithm for the case of a simple inventory Kanban situation. The experiments are performed and a discussion is provided regarding how the number of experiments involved in the generalized technical contradictions affects the number of classical physical contradictions. In the

discussion, we conclude this chapter and discuss what questions this chapter enabled to answer.

3-1 Introduction and state of the art

In Chapter 2, the inventive problem was limited only to the objective or evaluation problem, and so was presented through the generalized technical contradiction as a generalization of the classical technical contradiction. As the technical contradictions are concerned with the evaluation parameters, the inventive problem was formulated only at the evaluation parameters level, which is equal to the network of contradictions [25] in OTSM-TRIZ. To obtain a deeper comprehension and interpretation of the inventive problem, the physical contradiction should be formulated and identified as an initial point for the inventive solving process.

In terms of OTSM-TRIZ, a network of parameters is under the network of contradictions and describes the conflicts relation between numerous action parameters. Similarly, for the generalized contradiction model [30], a generalized physical contradiction model was proposed in [31], which managed the conflict system states instead of the network of parameters. It was a generalization of classical TRIZ physical contradictions, which used a conflict between two combinations of valued action parameters to replace the conflict in one action parameter. Integrating with the GTCs, the generalized system of contradiction is presented in Figure 25.

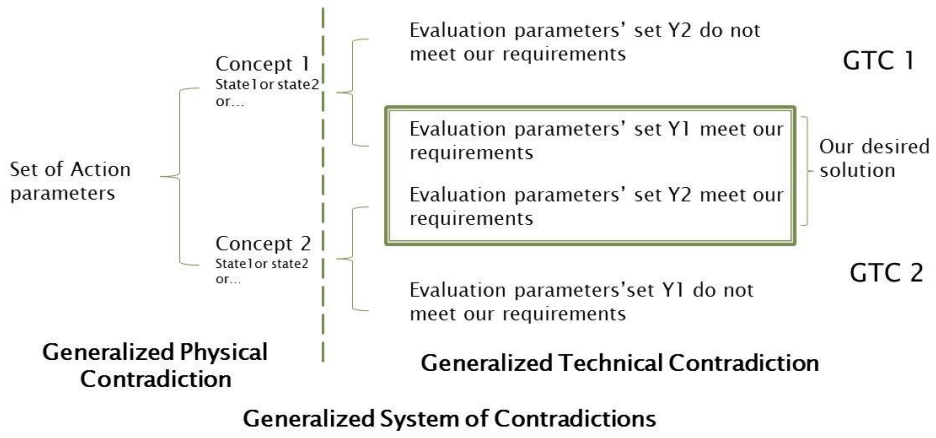


Figure 25: Generalized system of contradiction

The generalized system of contradiction is a generalization of the system of contradictions within OTSM-TRIZ. The element is generalized to the system. The single action parameter is replaced by set of action parameters, as is the evaluation parameter. However, the two values of conflict are the two states of one action parameter, while for a set of action parameters we use Concept to describe a union of system states. The system state is an instantiation of a system by fixing the value of the action parameters. Thereby, Concept = State 1, State 2… or State n.

3-1.1 Generalized physical contradiction within an experimental table

In the background of chapter 1, we proposed the representation of the generalized system of contradiction within the experimental table. The generalized technical contradiction has been formulated also through the set of equations issued from this experimental table as well as three sets of experiments E1, E2, E0 that were defined in order to model the single components of the GTC. E1 satisfies the Y1 and dissatisfies Y2; E2 satisfies Y2 and dissatisfies Y1. Based on these three sets of E1, E2 and E0; a generalized physical contradiction can be identified according to the corresponding values of action parameters.

The representation of the experiments by a set of action parameters A = (A_1, A_2,…, A_n); a set of evaluation parameters Y = (y_1, y_2,…, y_r); and a set of experiments E = (e_1, e_2,…, e_9) is presented in Table 9. Inputs of experiments e_i are characterized by a set of values of action parameters, (V_{i1}, V_{i2},…, V_{in}). The outputs of experiments e_i are characterized by a set of binary value of evaluation parameters, (Z_{i1},Z_{i2},…, Z_{ir}), while y_i is satisfied by equaling 1 or equaling 0.

Table 9: An experimental table

	A_1	…	A_n	y_1	…	y_i	…	y_r
e_1	V_{11}		V_{1n}	Z_{11}		Z_{1i}		Z_{1r}
e_2	V_{21}		V_{2n}	Z_{21}		Z_{1i}		Z_{2r}
…								
e_8	V_{81}		V_{8n}	Z_{81}		Z_{1i}		Z_{8r}
e_9	V_{91}		V_{9n}	Z_{91}		Z_{1i}		Z_{9r}

If there is a given generalized technical contradiction characterized by the experiment set E1, E2 and E0, a generalized physical contradiction can be formulated. Identifying a generalized physical contradiction in Table 9, as the following:

- Concept 1: admit the states in E1, reject states in E2 and E0.
- Concept 2: admit the states in E2, reject states in E1 and E0.

Table 10: Representation of a Generalized Physical Contradiction

	A_1	⋯	A_n	Y1	Y2	y_r
E1	Concept 1			1	0	
E2	Concept 2			0	1	
E0				0	0	

Table 10 is obtained by performing permutations of the rows and columns of Table 9 in order to form (E1,E2,E0) and GTC (Y1,Y2). In Table 10, the value of the evaluation parameter set is normalized as being 1 if all the evaluation parameters in set Yi and Ej are satisfied, and as being 0 if not.

When the number of action parameters and experiments are large, the generalized physical contradiction extraction process is impossible to perform by manual approach. The automatic extraction process of Concept 1 and Concept 2 is the key problem for generalized physical contradiction extraction. An example is presented in next part to better understand the definition of Concept and generalized physical contradiction (GPC).

Table 11: Experiment results of the electrical circuit breaker

							Y1		Y2			Y0
		A1	A2	A3	A4	A5	y1	y3	y2	y5	y6	y4
E1	e1	1	1	0	0	1	1	1	0	1	1	1
	e3	1	0	1	0	0	1	1	0	0	0	0
	e4	1	1	0	0	0	1	1	1	0	0	1
	e5	1	0	1	0	1	1	1	0	1	1	0
	e7	1	0	1	1	0	1	1	0	0	0	0
E2	e2	0	1	1	1	1	0	0	1	1	1	0
	e6	0	1	0	1	2	0	0	1	1	1	1
	e9	0	1	0	0	2	0	0	1	1	1	1
E0	e8	1	0	0	0	1	1	0	0	1	1	1

The generalized physical contradiction (GPC) model is a generalization of the classical TRIZ physical contradiction where one action parameter should be in two different states in order to satisfy design objectives for two different evaluation parameters. In GPC, we consider a set of action parameters, which should take two different combinations or concepts of values to satisfy two different sets of evaluation parameters. In a table, such as Table 10, the results for a rectangular matrix, where the rows correspond to the experiments and the columns respectively, the action and evaluation parameters describe the inventive problem. For example, the re-ordering presented in Table 11 is a representation of a generalized system of contradictions, as illustrated in Figure 26.

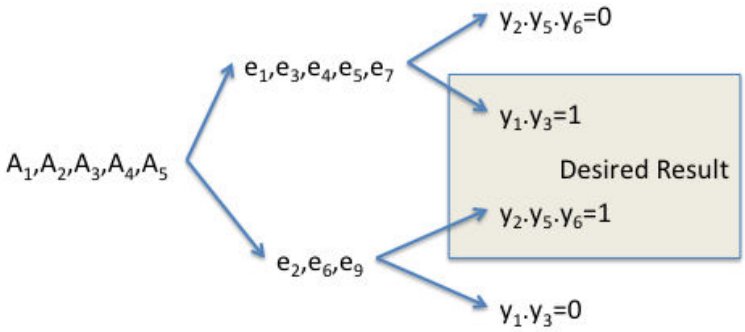


Figure 26: Generalized system of contradictions for the example

Looking for a generalized physical contradiction for the GSC, as represented in Figure 26, means searching for two Concepts such as:

$Concept_1(A_1,A_2,A_3,A_4,A_5) \Rightarrow (y_1=1).(y_3=1).[(y_2=0)+(y_5=0)+(y_6=0)]=1$

$Concept_2(A_1,A_2,A_3,A_4,A_5) \Rightarrow [(y_1=0)+(y_3=0)].(y_2=1).(y_5=1).(y_6=1) =1$

When trying to formulate the contradiction, one can easily recognize that no classical TRIZ physical contradiction exists. Even if A_1 always equals 1 in experiments of E_1, and equals 0 in these of E_2, it does not fit the condition. We have the condition $(A_1=0) \Rightarrow [(y_1=0)+(y_3=0)].(y_2=1).(y_5=1).(y_6=1) =1$; however, it does not fit the condition $(A_1=1) \Rightarrow (y_1=1).(y_3=1).[(y_2=0)+(y_5=0)+(y_6=0)]=1$ as for experiment e_8, $A_1=1$ whereas $(y_1=1).(y_3=1) = 0$.

A simple way to formulate the two concepts is to consider the models formed by the consideration of all the available knowledge of experiments of E_1 or E_2. Then, when considering all the experiments of E_2, the following applies:

$Concept_2(A_1,A_2,A_3,A_4,A_5)=(A_1=0)(A_2=1)(A_3=1)(A_4=1)(A_5=1)+(A_1=0)(A_2=1)(A_3=0)(A_4=1)(A_5=2)+(A_1=0)(A_2=1)(A_3=0)(A_4=0)(A_5=2)$,

which is the same for $Concept_1$, i.e., a concept built with the five experiments e_1, e_3, e_4, e_5, and e_7. However, such a representation, even if it is exact, is not necessarily the more appropriate to path leading to resolution.

An even more simplified expression could considered for the two concepts: $Concept_1(A_1,A_2,A_3,A_4,A_5)=(A_1=1).(A_3=1)$ and $Concept_2(A_1,A_2,A_3,A_4,A_5)=(A_1=0).(A_3=0)$. This contradiction is maybe more meaningful for the experts and can provide guidance towards a concept solution considering the difference for $Concept_2$, where only $(A_1=0)$. With regard to Table 9, it is relevant as the previous formulation; however, with regard of the considered decision research space, the first formulation was more constrained. Therefore, this pointed towards a more narrowed research space.

A number of GPC can be formulated for one considered GTC; therefore, the question rises of which one is more relevant to consider for resolution of the problems. To be able to answer this question, the first problem to solve, which is presented in this chapter, is to list all the possible GPCs or to have a synthetic overview of them.

3-2 Formulation of the BIP problem and the exhaustive search algorithm

As seen previously, there are many concept candidates for belonging to a GPC. To seek any possible GPC, we need to be able to represent a general shape of the concept that allows searching for a concept belonging to a GPC. In this section, we propose a parametric form of concept parameters, which are required to verify a set of binary equations to define an admissible concept of the GPCs. This set of binary equations will define the set of admissible concepts. However, as this set may be very large, it may not be simple to list all concept solutions or even find all of them. To address this problem, specific properties are used for the search problem. It can be shown that one specific GPC can summarize all possible GPCs. The search problem for this specific GPC can then be obtained by adding an objective function to the

previous set of equations; thus, this search results in solving a tractable binary integer optimization problem.

3-2.1 Parametric form of a concept

The purpose of this section is to propose a generic shape of concept based on a finite number of parameters. Therefore, defining the concept is equivalent to defining the value of its parameters.

Definition of concept

Denote A_k with $k \in \{1,\ldots,n\}$ as an Action Parameter and x_k the value of the parameter A_k. Therefore, the definition of concept is:

Concept is a logical function of $(A_k = x_k)$ that accepts the corresponding E_{1or2} and rejects the rest of experiments, i.e., for $\forall e_i \in E_1, \forall e_j \in E_2 \cup E_0$,

we then have $Concept_1(e_i) = 1 \; and \; Concept_1(e_j) = 0$.

For convenience, we use '+' to represent OR, and use $'\times'$ to represent AND. For this logical function, we suppose the logical AND has operator priority to the logical OR.

Thus, concept has the form of $Concept = \sum(\prod(A_k = x_k))$,

where Π is the synthetic way of expressing the logical operator AND; and $\sum$ is the synthetic way for expressing the logical operator OR.

For $\prod(A_k = x_k)$, the logical operator AND has the property of the commutativity; we can permutate and combine the same item $(A_k = x_k)^n$. The idempotence is then $(A_k = x_k)^n = (A_k = x_k)$.

Furthermore, because of the logical property of complementation, which means that $x_1 \neq x_2, (A_k = x_1)(A_k = x_2) = 0$, i.e., the action parameter cannot be equal to different values within one experiment.

These properties promote the definition of the state of system as:

$I_i = \prod_{k \in J_i}(A_k = x_k)$ where $J_i \subset \{1,\ldots,n\}$ (J_i is a subset of {1,…,n}).

Thereby, the concept has the form of: $Concept = \sum\left(\prod_{k \in J_i}(A_k = x_k)\right)$ or $Concept = \sum state$.

The binary variable representation for Concept space

The actual question is how many possible states exist for the concept? Denote *s* as the number of possible states that can be obtained because of the action parameters and their values. Then expressing the value of s as a function of n and of the number of values of each action parameter A_k. The variable a_k is defined as the number of possible values of A_k. Then, as an action parameter can either be present or not in a state but where at least one parameter is in a state, the value of s is $s = [\prod_{k=1}^{n}(a_k + 1)] - 1$. The number of non-empty subset of $J_i \subset \{1,\ldots,n\}$ then equals *s*.

As the whole set of states is known and the definition of the whole set of concept by combining any of the states is also known, the concept can be written as:

$$Concept(A_1, A_2, A_3, \ldots, A_n) = \sum_{i=1}^{s} d_i . I_i \quad (1),$$

where d_i equals either the logical expression 1 (true) or 0 (False). Thus, d_i allows for keeping or removing a state from the concept and $\sum$ is the logical 'OR' function.

Once, we know the concept space can be described as (1) by the binary vector $(d_1, d_2, \ldots, d_s)$, the space $\{0,1\}^s$ can be the representation of the concept space.

Defining an order for a state

At this stage defining a concept is equivalent to defining the d_i functions. However, the definition of di requires knowing the definition of I_i. For convenient processing by computer, a bijection between the states and the set $S = \{1,\ldots,s\}$ is constructed. Using this bijection, a binary vector with the dimension of s can be used for representing a generic concept. Thus, the vector $(d_1, d_i,\ldots,d_S)$, which represents a concept, is interpreted as follows: if a concept has the item I_i, its corresponding function value d_i equals '1'; otherwise, its value equals '0'.

The bijection we proposed is the lexicographical order. To be specific, for any state $I_i = \prod_{k \in J}(A_k = x_{kj})$, x_{kj} is the j^{th} value of A_k. Its order can be calculated by:

$$\text{Order}(\prod_{k \in J \subseteq \{1,\ldots,n\}}(A_k = x_{kj})) = j_1 + j_2(a_1 + 1) + j_3(a_1 + 1)(a_2 + 1) \ldots + j_k \prod_{l=1}^{k-1}(a_l + 1) : k \in J$$

By this function, we put all states into [1, s], and the corresponding integer number is their order. The vector $(d_1, , d_i,\ldots,d_S)$ represents the set of the states and d_{order} is a component in vector representing one state. For computer program, we can use a

binary array, with length of s, to present a concept. If the state ordered i belongs to the concept, then d_i equals 1, otherwise d_i equals 0.

3-2.2 Choice of the relevant concepts for the GPC

In previous section we showed how to define a concept with a vector (d1, ,di,…,ds). In this section, we show the constraints on the di values that allow the concept to be a concept of a GPC. Concept1 and Concept2 are related by the set of experiments E1 and E2 respectively. This indicates that each concept has to satisfy the requirements: for any experiment of Ei=1,2, where the configuration of the values of any evaluation parameter has to enable the satisfaction of Yi and to assure that Yi,j=0,1,2;, and j≠i is not satisfied.

It indicates that $Concept_i$ should admit experiments of E_i and reject those of $E_{j,j\neq i}$. In other words, the options of experiments in the E_i ensure the $Concept_i$ is equal to 1, while other experiments ensure it is equal to 0. According to the aforementioned properties, we can restrict the research decision space by several equations.

Defining e_{jk} as the value for experiment e_j for the action parameter A_k in the experimental matrix and c_{ji} as the value of the state I_i for experiment e_j then

$c_{ji}=\prod_{k\in Ji}(A_k=e_{jk})$ (2)

With these notations, the following are constraints for Concept1:

For each $e_j\in E_1$: $\sum_{i=1}^{s} c_{ji}d_i = 1$ (3)

For each $e_j\in E_2\cup E_0$: $\sum_{i=1}^{s} c_{ji}d_i = 0$ (4)

Dual constraints hold for Concept2:

For each $e_j\in E_2$: $\sum_{i=1}^{s} c_{ji}d_i = 1$ (5)

For each $e_j\in E_1\cup E_0$: $\sum_{i=1}^{s} c_{ji}d_i = 0$ (6)

The set of equations (3) and (4) define the constraints on d_i for being Concept1 of a GPC. Additionally, the set of equations (5) and (6) define the constraints on d_i for being the Concept2 of a GPC. In other words, these equations provide the states that can compose a concept of a GPC. It can be shown that any item verifying these equations is a relevant concept of GPC. As a consequence any combination of the states describing Concept1 (resp. Concept2) are also admissible concepts for

Concept1 (resp. Concept2). Moreover, the way these concepts where defined implies that any admissible concept verifying equations (3) and (4) can be coupled with any solution of equations (5) and (6) to build a GPC. Thus, the number of possible GPCs can be very large and it may not be simple to list all the concept solutions of these equations. In order to address this problem, we will use specific properties for our search problem that are given in the next section.

The most general concept

As we noted previously, any solution of $Concept_i$ is a combination of a limited number of states as defined by previous mentioned constraints. So, the most general $Concept_i$ is the concept containing all the admissible states for $Concept_i$. The knowledge of this concept would allow us to know all the states that allow generation of all the admissible concepts. The size of a concept can be seen as the sum of the *di* values. As a consequence, we can state our search of the longest concept as the maximization of an objective function subject to constraints previously mentioned:

For $Concept_1$:

Objective function: $Max(\sum_{i=1}^{s} d_i)$

Restrict to:

$\sum_{i=1}^{s} c_{ji} d_i = 1$; For each $e_j \in E_1$

$\sum_{i=1}^{s} c_{ji} d_i = 0$; For each $e_j \in E_2 \cup E_0$

$d_i is\ binary\ variable$;

For $Concept_2$:

Objective function: $Max(\sum_{i=1}^{s} d_i)$

Restrict to

$\sum_{i=1}^{s} c_{ji} d_i = 1$; For each $e_j \in E_2$

$\sum_{i=1}^{s} c_{ji} d_i = 0$; For each $e_j \in E_1 \cup E_0$

$d_i is\ binary\ variable$;

The d_i values of the previously mentioned optimization problems provide the states that generate all the possible concepts involved in the GPC of a given set of experiments. Up to this stage we did define a Binary Integer Programming problem (BIP) for obtaining our concepts. Next step is to propose an algorithm that solves it.

3-2.3 The exhaustive GPC search algorithm

Methods to solve our BIP problem

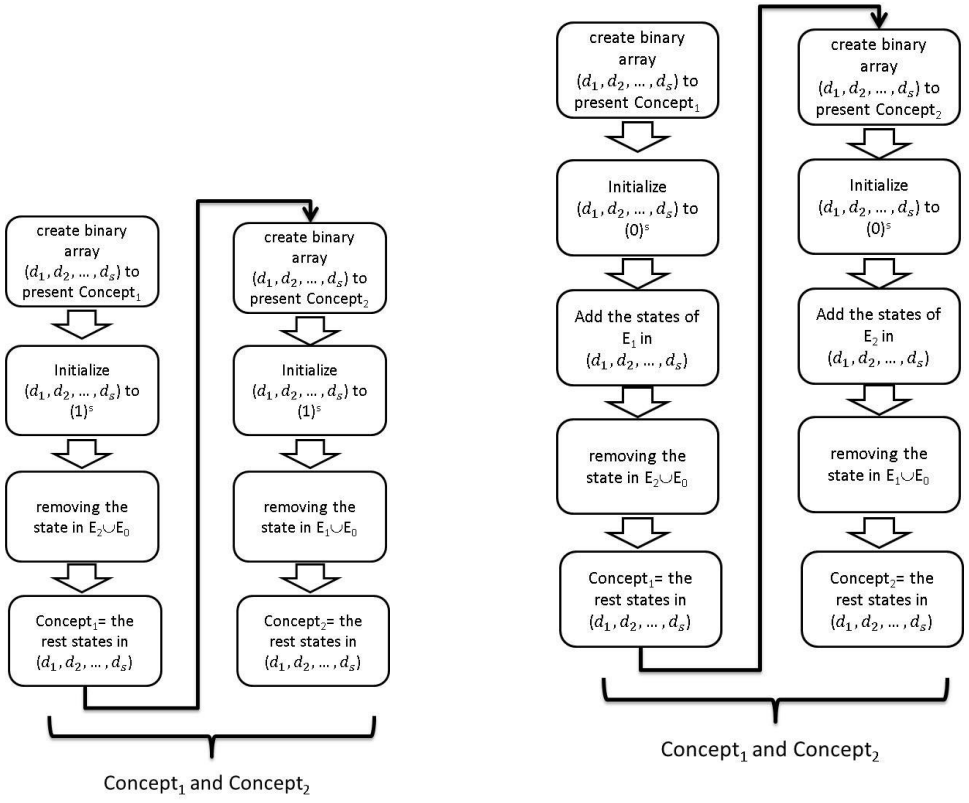
There are many general methods for solving BIP problems, such as Branch and Cut; Branch and Bound, and cutting plane [51]. However, our BIP has properties that simplify the solving process. In the constraint equations described above, the notation '+' indicates the logical operation 'or'. In terms of logical operations, if the right side of the equations is 0 (i.e., equations (4) and (6)), all associated items on the left side of the equation should be equal to zero. Hence, if coefficient c_{ji} is not equal to 0 on the left side of equation, the associated d_i certainly equals 0.

The exhaustive search algorithm

Based on the statement above, we can calculate the c_{ji} values first; then make all the associated d_i of non-zero coefficient c_{ji} equal to 0. Ensuring the remainder of the decision variables d_i is equal to 1 will lead to our objective function reaching a maximum value. However, two situations appear. If our experimental table corresponds to full factorial table in the design of experiments, the proposed method is then reasonable. Alternately, if our experimental table does not correspond to the full factorial table in the design of experiments this method will not work, such as in the case of the electrical circuit breaker. To illustrate, if we set all variables of d_i, that are not constrained by equations (4) and (6) equal to 1, then some states not constrained by equations (3) and (5) will become factors for the two concepts. For this situation, we are required to initialize the set of states by equations (3) and (5), then remove the states by utilizing equations (4) and (6).

In practice, to calculate $Concept_1$, we define the binary array $(d_1, d_2, \dots, d_s)$ to initialize the array to $(1)^s$. For full factorial experiments, we enumerate the combinations of action parameters with values in $E_2 \cup E_0$, which is the state involved in the experiment. Each AP has only one value because of the single experiment; thus, the number of combination states is 2^n-1. After enumerating the states and denoting the corresponding value to 0 in $(d_1, d_2, \dots, d_s)$, the rest of states in $(d_1, d_2, \dots, d_s)$ is the state for $Concept_1$. The same process is followed for $Concept_2$. When searching $Concept_1$ in the incomplete experiments, we initialized $(d_1, d_2, \dots, d_s)$ to $(0)^s$. First we enumerate all the states in E_1, and denote the

corresponding value to 1 in $(d_1, d_2, \dots, d_s)$. Then we enumerate all the states in $E_2 \cup E_0$, and remove it from $(d_1, d_2, \dots, d_s)$ by denoting the corresponding value to 0. The rest state is $Concept_1$. It is also the same for $Concept_2$. The whole searching process is described in Figure 27.



(a) For full factorial experiments (b) For incomplete experiments

Figure 27: Searching process of two concepts

The computation complexity of concept searching

The time consumption for a full factorial experiment corresponds to two parts. The first part is to remove the states in E_1+E_0 and E_2+E_0. Thus, the computation time is $s \times (|E_1 + E_0| + |E_2 + E_0|)$. The second part involves comparing the states from two concepts. It cannot be known the exact number of states for the two concepts. However, there is an upper bound s and lower bound 1. Thus, the comparing time is ranges from 1 to s^2. Based on two part time consumption, s has a decisive role in the time complexity as follows:

s in fact is the number of states: $s = [\prod_{k=1}^{n}(a_k + 1)] - 1$,

a_k is the number of possible values of A_k, and it at least equal to 2.

Thereby, with the increasing number of columns, the time consumption experiences an exponential increase.

3-2.4 Full factorial experiment vs incomplete experiment results

When the experimental table is not complete, some states in concept are uncertain because some general states are generalized from many experiments.

Table 12: Full factorial experiments and its partial experiments

(a) Full factorial experiments

		A1	A2	A3
E1	e1	0	0	1
	e2	1	0	0
	e3	1	0	1
E2	e4	0	0	0
	e5	0	1	0
	e6	0	1	1
E0	e7	1	1	0
	e8	1	1	1

(b) Incomplete experiments

		A1	A2	A3
E1	e3	1	0	1
E2	e4	0	0	0
	e6	0	1	1
E0	e7	1	1	0
	e8	1	1	1

For example in Table 12 (a), there are three binary value APs: AP1, AP2, and AP3. The GTC is given for E1 vs E2. For Concept1, the state (A1=1)(A2=0) is from experiments e2 and e3. However, for the case of partial experiments shown in Table 12(b), Concept1 still has the state (A1=1)(A2=0) from the proposed concept extraction method. The other state (A1=0)(A3=0) of E2 in Table 12(a) is from e4 and e5. It is also a state of Concept2 in Table 12(a) from e4. If we compare the two states (A1=1)(A2=0) and (A1=0)(A3=0); a physical contradiction is obtained where A1=1 vs A1=0 with the context of (A2=0)(A3=0). This physical contradiction is available for the case in Table 12(a). Whereas, for the case in Table 12 (b), the physical contradiction A1=1 VS A1=0 with context of (A2=0) is uncertain unless we are aware the result of e2 and e5 in Table 12(a). Furthermore, if we carefully observe Table 12(b), (A1=0) is a state for $Concept_2$, and (A2=0)(A3=1) is a state for $Concept_1$. There is no conflict when comparing the two states and it indicates that (A1=0)(A2=0)(A3=1) could

provide a solution for the system. This no conflicting pairs of states could provide a potential solution based on the hypothesis of mutual independency of two states. Thereby, an incomplete experiment could potentially provide physical contradiction with context and even some potential solutions based on non-conflicting pairs as needed by the engineers. Alternately, the full factorial experiments can provide certain physical contradictions and present an entire decision search space. That is why the focus on the properties of GPC extraction from full factorial experiments, although for some situations full factorial experiment is not available. In the next chapter we will propose a method to bypass this problem.

3-2.5 From the generalized to classical physical contradiction

As mentioned above, conflict between two states of Concept can be formulated as a physical contradiction with context. This presents a straight line link between the system of contradictions and the generalized system of contradiction.

Table 13: Two concepts for the case above

$Concept_1$=(A1=1)(A2=0)(A3=0)+(A1=0)(A2=0)(A3=1)+(A1=1)(A2=0)(A3=1)+(A2=0)(A3=1)+(A1=1)(A2=0)
$Concept_2$=(A1=0)(A2=0)(A3=0)+(A1=0)(A2=1)(A3=0)+(A1=0)(A3=0)+(A1=0)(A2=1)(A3=1)+(A1=0)(A2=1)

The two Concepts of Table 12(a) are shown in
Table 13. Any pair of states, one from $Concept_1$ and the other one from $Concept_2$, can form a classical physical contradiction with context. In the Figure 28, (A1=1)(A2=0)(A3=0) from $Concept_1$ and (A1=0)(A2=0)(A3=0) from $Concept_2$ form a classical physical contradiction with context.

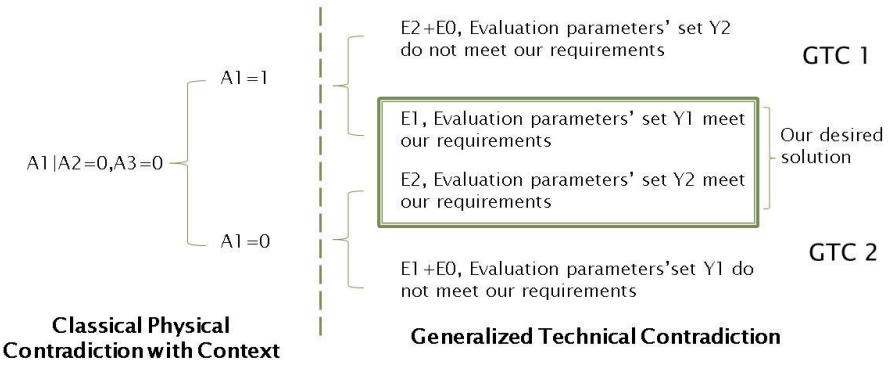


Figure 28: Illustration of the classical physical contradiction

The context of contradiction describes a background of the contradiction. It is more like super system of the conflict parameter, which is the environment of conflict element. As the physical contradiction in Figure 28, the environment is (A2=0)(A3=0), the conflict is (A1=0) vs (A1=1). If we change the environment, the contradiction commonly changes.

3-3 Illustration of the algorithm on the practical cases

3-3.1 Case of electrical circuit breaker

In this section, we aim at illustrating the definition of the GPC based on the example used for the extraction of GTCs in Chapter 2. The description of the circuit breaker was presented in Chapter 2 and a summary is provided here based on the reorganized results to illustrate the GTC and GPC as shown in Table 14.

Table 14: Experimental results of the electrical circuit breaker

							Y1		Y2			Y0
		A1	A2	A3	A4	A5	y1	y3	y2	y5	y6	y4
E1	e1	1	1	0	0	1	1	1	0	1	1	1
	e3	1	0	1	0	0	1	1	0	0	0	0
	e4	1	1	0	0	0	1	1	1	0	0	1
	e5	1	0	1	0	1	1	1	0	1	1	0
	e7	1	0	1	1	0	1	1	0	0	0	0
E2	e2	0	1	1	1	1	0	0	1	1	1	0
	e6	0	1	0	1	2	0	0	1	1	1	1
	e9	0	1	0	0	2	0	0	1	1	1	1
E0	e8	1	0	0	0	1	1	0	0	1	1	1

For the GTC in Table 14, the experiments are not full factorial; thus, the GPC extraction described in Figure 27(b) was performed. There are 67 states involved in Concept1, and 58 states in Concept2. They are presented in Table 15.

Table 15: Two concepts of GTC for electrical circuit breaker

Concept 1	Concept2
(A1=1)(A2=1)(A3=0)(A4=0)(A5=0)	(A1=0)(A2=1)(A3=1)(A4=1)(A5=1)
(A2=1)(A3=0)(A4=0)(A5=0)	(A2=1)(A3=1)(A4=1)(A5=1)
(A1=1)(A3=0)(A4=0)(A5=0)	(A1=0)(A3=1)(A4=1)(A5=1)
(A3=0)(A4=0)(A5=0)	(A3=1)(A4=1)(A5=1)
(A1=1)(A2=0)(A3=1)(A4=0)(A5=0)	(A1=0)(A2=1)(A4=1)(A5=1)
(A2=0)(A3=1)(A4=0)(A5=0)	(A2=1)(A4=1)(A5=1)
(A1=1)(A3=1)(A4=0)(A5=0)	(A1=0)(A4=1)(A5=1)
(A3=1)(A4=0)(A5=0)	(A4=1)(A5=1)
(A1=1)(A2=0)(A4=0)(A5=0)	(A1=0)(A2=1)(A3=1)(A5=1)
(A2=0)(A4=0)(A5=0)	(A2=1)(A3=1)(A5=1)
(A1=1)(A2=1)(A4=0)(A5=0)	(A1=0)(A3=1)(A5=1)
(A2=1)(A4=0)(A5=0)	(A1=0)(A2=1)(A5=1)
(A1=1)(A4=0)(A5=0)	(A1=0)(A5=1)
(A4=0)(A5=0)	(A1=0)(A2=1)(A3=0)(A4=0)(A5=2)
(A1=1)(A2=0)(A3=1)(A4=1)(A5=0)	(A2=1)(A3=0)(A4=0)(A5=2)
(A2=0)(A3=1)(A4=1)(A5=0)	(A1=0)(A3=0)(A4=0)(A5=2)
(A1=1)(A3=1)(A4=1)(A5=0)	(A3=0)(A4=0)(A5=2)
(A3=1)(A4=1)(A5=0)	(A1=0)(A2=1)(A4=0)(A5=2)
(A1=1)(A2=0)(A4=1)(A5=0)	(A2=1)(A4=0)(A5=2)
(A2=0)(A4=1)(A5=0)	(A1=0)(A4=0)(A5=2)
(A1=1)(A4=1)(A5=0)	(A4=0)(A5=2)
(A4=1)(A5=0)	(A1=0)(A2=1)(A3=0)(A4=1)(A5=2)
(A1=1)(A2=1)(A3=0)(A5=0)	(A2=1)(A3=0)(A4=1)(A5=2)
(A2=1)(A3=0)(A5=0)	(A1=0)(A3=0)(A4=1)(A5=2)
(A1=1)(A3=0)(A5=0)	(A3=0)(A4=1)(A5=2)
(A3=0)(A5=0)	(A1=0)(A2=1)(A4=1)(A5=2)
(A1=1)(A2=1)(A3=0)(A5=0)	(A2=1)(A4=1)(A5=2)
(A2=0)(A3=1)(A5=0)	(A1=0)(A4=1)(A5=2)
(A1=1)(A3=1)(A5=0)	(A4=1)(A5=2)
(A3=1)(A5=0)	(A1=0)(A2=1)(A3=0)(A5=2)
(A1=1)(A2=0)(A5=0)	(A2=1)(A3=0)(A5=2)
(A2=0)(A5=0)	(A1=0)(A3=0)(A5=2)
(A1=1)(A2=1)(A5=0)	(A3=0)(A5=2)
(A2=1)(A5=0)	(A1=0)(A2=1)(A5=2)
(A1=1)(A5=0)	(A2=1)(A5=2)
(A5=0)	(A1=0)(A5=2)
(A1=1)(A2=1)(A3=0)(A4=0)(A5=1)	(A5=2)
(A2=1)(A3=0)(A4=0)(A5=1)	(A1=0)(A2=1)(A3=0)(A4=0)
(A1=1)(A2=0)(A3=1)(A4=0)(A5=1)	(A1=0)(A3=0)(A4=0)
(A2=0)(A3=1)(A4=0)(A5=1)	(A1=0)(A2=1)(A4=0)

(A1=1)(A3=1)(A4=0)(A5=1)	(A1=0)(A4=0)
(A3=1)(A4=0)(A5=1)	(A1=0)(A2=1)(A3=0)(A4=1)
(A1=1)(A2=0)(A4=0)(A5=1)	(A2=1)(A3=0)(A4=1)
(A2=1)(A4=0)(A5=1)	(A1=0)(A3=0)(A4=1)
(A1=1)(A2=1)(A3=0)(A5=1)	(A3=0)(A4=1)
(A2=1)(A3=0)(A5=1)	(A1=0)(A2=1)(A3=1)(A4=1)
(A1=1)(A2=0)(A3=1)	(A2=1)(A3=1)(A4=1)
(A2=0)(A3=1)(A5=1)	(A1=0)(A3=1)(A4=1)
(A1=1)(A3=1)(A5=1)	(A1=0)(A2=1)(A4=1)
(A1=1)(A2=1)(A5=1)	(A2=1)(A4=1)
(A1=1)(A2=1)(A3=0)(A4=0)	(A1=0)(A4=1)
(A1=1)(A2=0)(A3=1)(A4=0)	(A1=0)(A2=1)(A3=0)
(A2=0)(A3=1)(A4=0)	(A1=0)(A3=0)
(A1=1)(A3=1)(A4=0)	(A1=0)(A2=1)(A3=1)
(A3=1)(A4=0)	(A2=1)(A3=1)
(A1=1)(A2=1)(A4=0)	(A1=0)(A3=1)
(A1=1)(A2=0)(A3=1)(A4=1)	(A1=0)(A2=1)
(A2=0)(A3=1)(A4=1)	(A1=0)
(A1=1)(A3=1)(A4=1)	
(A1=1)(A2=0)(A4=1)	
(A2=0)(A4=1)	
(A1=1)(A4=1)	
(A1=1)(A2=1)(A3=0)	
(A1=1)(A2=0)(A3=1)	
(A2=0)(A3=1)	
(A1=1)(A3=1)	
(A1=1)(A2=1)	

If we randomly choose one of the states from Concept1 such as (A2=0)(A3=1)(A4=0)(A5=0), we can see that it is supported by e3 in Table 14, and rejected by experiments in (E2∪E0). It is the same for states in Concept2 such as (A1=0)(A4=0)(A5=2), which is supported by e9, and rejected by experiments in (E1∪E0). The classical contradiction is formed by (A2=0)(A3=1); (A4=0)(A5=0); and (A1=0)(A4=0)(A5=2). The context is (A1=0)(A2=0)(A3=1)(A4=0), and the conflict is (A5=0)vs(A5=1). As the experiment set is not full factorial, this contradiction is not supported by the experiments in Table 14, and it has to be proved by experiments or human experts. There are also some non-conflict pairs such as (A1=1)(A2=1) that are supported by e1 and e4 in Concept1, and (A2=1)(A5=2) as supported by e6 and e9 in Concept2. The combination of the two states indicate a solution at (A1=1)(A2=1)(A5=2), which also needs to be proven by new experiments and experts.

For the incomplete experiments where many experiments are unknown, the deduction from experiments is uncertain. The statistical method for providing reliable deduction is not the object of this thesis. We tend to propose the GPC extraction on the full factorial experiments. The next case study illustrates this kind of situation.

3-3.2 Case of simple inventory Kanban system

The case of the simple inventory Kanban is presented in order to illustrate the BIP formulation and its resolution through the full factorial experiment table. The proposed case has been built out of an industrial supply chain problem, but only an excerpt of the problem is presented here. The problem is the following one: consider an inventory of products where the supplier has interaction with a delay of 8 time units for delivering the product and a customer who has exclusive demands. The customer needs are as follows: one unit; the average period between two demands being one; but no regular period between two demands that can be simulated by an erlang-2 distribution with a mean value of 1. The process has been modeled on Witness[1], as illustrated in Figure 29.

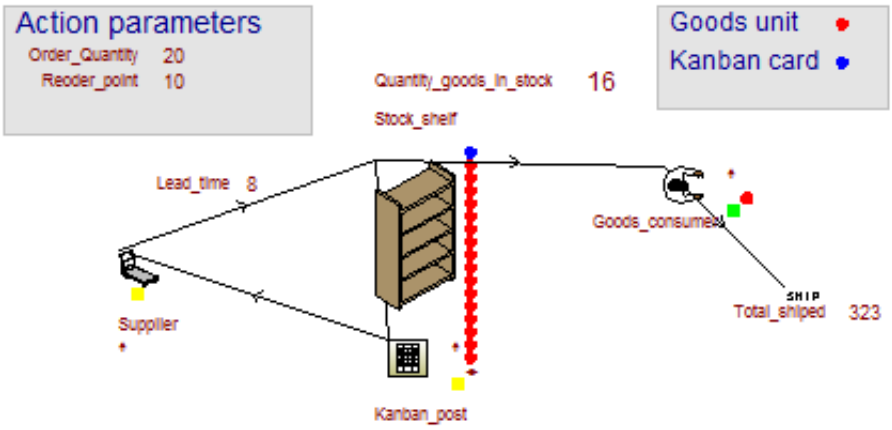


Figure 29: Witness model of the process

The inventory is controlled by a Kanban system, based on a type of reorder point system. The value of the reorder point is written on the Kanban card. The order quantity is also written on the Kanban card. When the inventory level is lower than the reorder point, the Kanban card is sent to the supplier. Goods are delivered with

[1] http://www.lanner.com/fr/witness/

the Kanban card and left together on the inventory shelf until the inventory level crosses under the reorder point[2].

For this problem, two evaluation parameters have been defined:

- The service breakdown (1-customer level service), which has to be as low as possible, ideally equal to 0.
- The average stock, which also has to be as low as possible, and also ideally equal to 0.

Influences on the system because of the perimeter of the system are defined as follows: there is no possible action by the supplier and no possible action by the customer. Then, the following points are noted:

- The level of the reorder point, which is the level of inventory defining when to order the product, can vary between 1 and 20.
- The lot size (reorder quantity) can vary between 1 and 20.
- The supplier scrap rate is a percentage of the non-certified raw material. For the experiments, three levels have been considered: 0, 10, 20%.

With these data, 1200 experiments were simulated in Witness (full factorial experiments), and they defined the reachable solutions (presented on Figure 30.

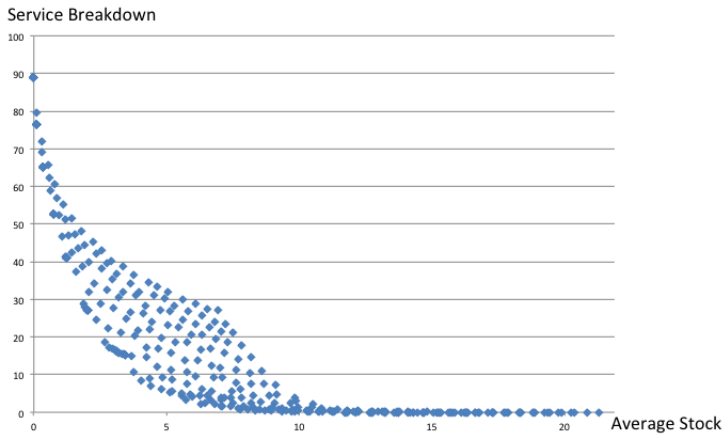


Figure 30: Pareto frontier of the initial simulation

[2] A system working on this principle is described in https://www.youtube.com/watch?v=tum1lLwy6gE

GSC extraction for the case study

The first step of inventive problem solving for this case study is the GTC extraction, as there are only two evaluation parameters: average stock (AS) and service breakdown (SB); the only GTC is a TC of AS and SB. We define the ideal value 0 as satisfying the value of AS and SB. Thus, there are 60 experiments involved in E_1 for satisfying AS and 120 experiments involved in E_2 for satisfying SB.

The GPC extraction was performed during the full factorial experiments. Eighty-four states were extracted for $Concept_1$, 152 states for $Concept_2$. Thus, there are 12,768 GPCs with context, 736 of them only involve one action parameter conflict. In order to formulate the system of contradictions, among all these contradictions, only two states have been considered for forming the physical contradiction. These two states correspond to the two extreme points of the Pareto frontier, i.e., the ones taking separately the best values of each evaluation parameters. They are the state equations of (Reorder Point =15)(Lot size = 13)(scrap rate =0) vs (Reorder Point =15)(Lot size = 1)(scrap rate =0). The system of contradictions that has been extracted out of the experiments is presented in Figure 31.

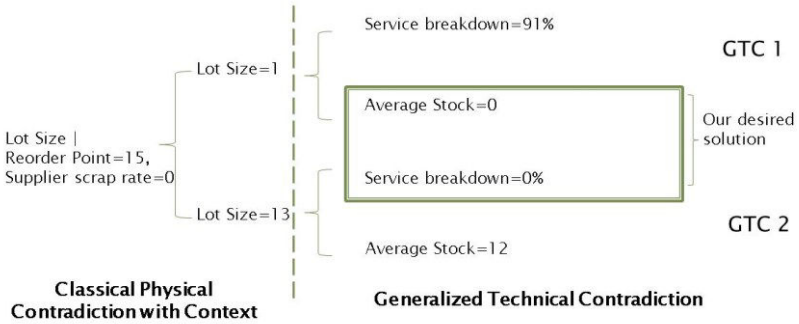


Figure 31: GPC and its context related to the initial Kanban system

This contradiction stated that a classical TRIZ physical contradiction exists if the action parameters reorder point and the supplier scrap rate are respectively fixed at 15 units and 0%. As a result, the lot size has to be 1 to achieve an average stock of 0 units, but it has to be 13 in order to satisfy a service breakdown of 0%.

To interpret this GPC for this simple Kanban model, there are two APs describing the capacity of production: a lot size and a reorder Point. The Reorder Point describes the frequency of supplying, and the Lot size describes the quantity to supply. There

are two states: the supplier scrap rate is equal to 0% and the Reorder Point is equal to 15 while lot size is 13. When there are less than 15 products in the inventory, the supplier sends 13 products to the stock because of the lead time. This point lies on Pareto frontier where Lot size = 13 achieves the most ideal (lowest) average stock while Service Breakdown equals to 0 (service rate is 100 %). On the other side, if lot size = 1, it means when there is less than 15 products in the inventory. The supplier will then send only 1 product to the stock, which arrives after eight time units of lead time. However, during this time, more than 1 product should be delivered, which leads to the average stock = 0 but a service breakdown of 91 % (service rate of 19 % only). This is another point on the Pareto frontier minimizing the average stock but maximize the service breakdown.

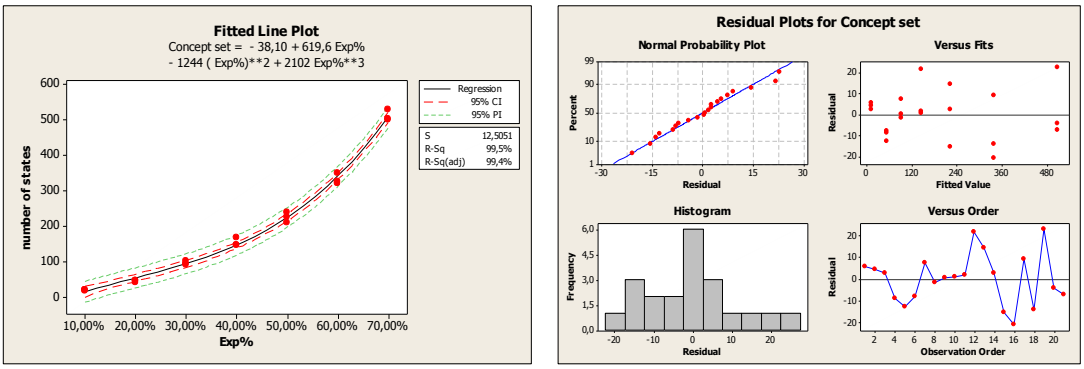
3-4 Exploitation of algorithm

The more states available for concept, the more physical contradictions that can be provided, which would allow for more partial solutions. Additionally, the system has great potential to be improved. The GTC extraction provides the information about the number of experiments in the sets E1 and E2. The question is if this has any effect on the GPC extraction? In order to answer this question and investigate the relation between the number of the physical contradictions and corresponding GTC, we proposed two groups of full factorial experiments. The first group is to study the relation between the number of states and the number of experiments in E1 and E2. The second group is to study the relationship between the number of states and the number of physical contradictions. The second group is also reorganized in the last section to present the relation between number experiments and the number of classical physical contradiction with context.

3-4.1 Relationship between number of experiments and states

The number of experiments involved in E1 and E2 can be described by the percentage of whole number of experiments l. The first group of experiments is in regards to the relationship between the number of states in a concept and the percentage of l number of experiments. We set the action parameter equal to 7 where AP has two options. So there are 128 experiments in full factorial

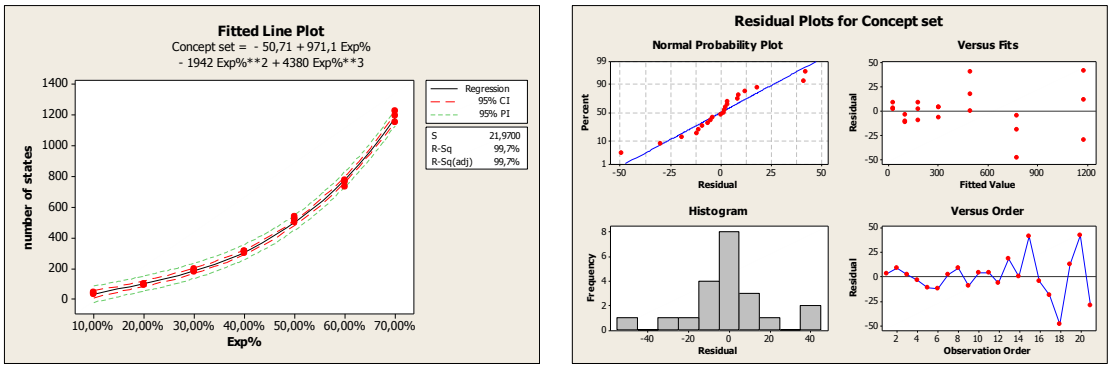
experimental table and we randomly choose 13(10%), 26(20%), 39(30%), 52(40%), 65(50%), 78(60%), 91(70%) rows and then repeated three times for each. In Figure 32 (a) and (b), Exp% denotes the percentage of experiments. We can then see that the number of states experiences cubic growth with the percentage of experiments.



(a) Fitted line plot (b) Residual plots

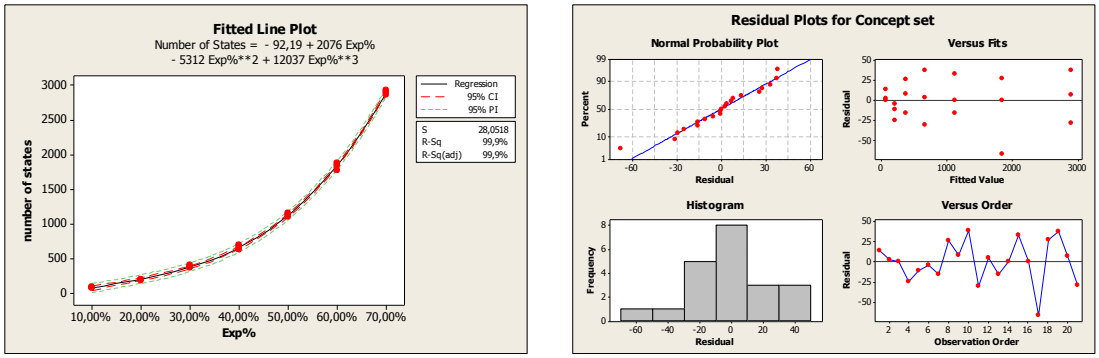
Figure 32: Relation between number of states and experiments (7 APs)

The following experiments are similar to previous experiments. Additionally 8 and 9 Binary action parameters were pulled. Then we randomly selected the same percentages of number of experiments in order to study the variation of the number of states. From Figure 33 and Figure 34, the fitted line revealed that the number of states and percentage of experiments involved in the concept has a cubic polynomial relation.



(a) Fitted line plot (b) Residual plots

Figure 33: Relation between number of states and experiments (8 APs)



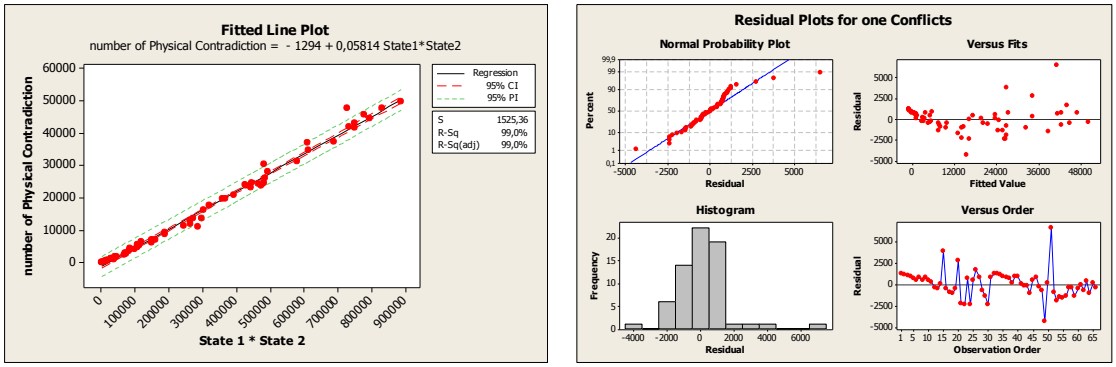
(a) Fitted line plot (b) Residual plots

Figure 34: Relation between number of states and experiments (9 APs)

Overall, we can conclude that with an increase of the percentage of experiments, which is comprised in the concept, the number of states experiences a cubic growth. It suggests that when we choose GTC, if we want to choose a GTC in which concepts has a more states, we need to choose the correct combination of E1 and E2 based on a large number of experiments.

3-4.2 Relationship between number of states and number of GPC

The GSC is composed of two GTCs (GTC1 and GTC2) and the classical physical contradiction, which is then a match for two states corresponding respectively to GTC 1 and GTC 2. Intuitively we can conclude that more pairs of states will lead to more classical physical contradictions with context. The compare time is equal to the product of the Concept1 states and the Concept2 states. In the following experiments, we will study how the number of classical physical contradictions varies with the product of the number of states in Concept1 and Concept2. We defined nine Binary action parameters. In order to make products uniform, we randomly performed 68 experiments with a different number of states and their products were located in the intervals from 952 to 888,349. From Figure 35, we can draw the conclusion that a number of classical physical contradictions experience linear growth with the product of two set of states.



(a) Fitted line plot (b) Residual plots

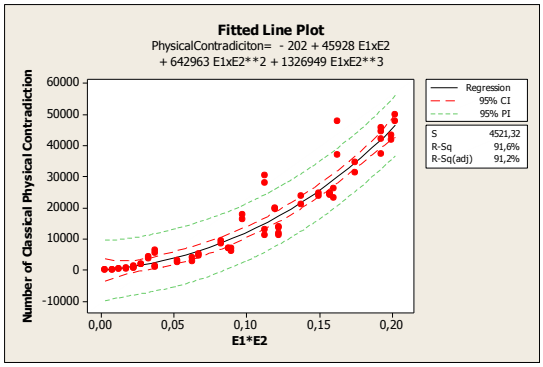
Figure 35: Link between number of classical PC and product of States 1 and 2

The relation of the number of classical physical contradiction is a linear proportion of State 1 * State 2. Additionally, each State is a cubic proportion to the number of experiments or percentage associated with the concept. Hence, we can provide an estimation that the number of classical physical contradictions is based on a cubic proportion of the number of experiments in E1 and E2.

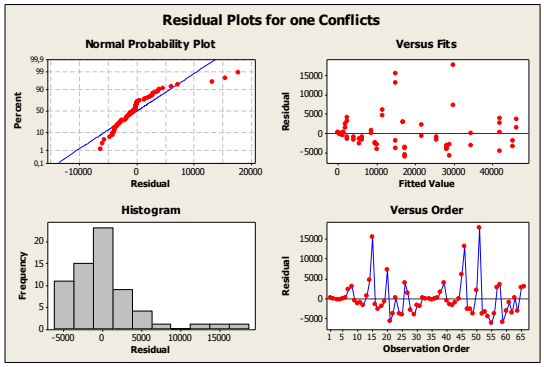
3-4.3 Relationship between number of experiments and number of PC

The previous experiments implied that the number of physical contradictions experiences cubic growth based on the product of the percentage of the number of experiments in E1 and E2. For this reason, we will work to verify the cubic growth by using the experiment results of the last section. In Figure 36, although we did not obtain a better cubic regression, it is revealed that with the growth of production of the percentages the number of physical contradiction is cubical and increasing despite less than an optimal cubic regression.

In summary, the experiments provide information regarding the number of experiments in E1/E2, which has a large effect on the number of classical physical contradictions with context. When we choose the GTCs, the experiments in E1 and E2 should be a considered element.



(a) Fitted line plot



(b) Residual plots

Figure 36: Relation between PC and number of experiments in E1 and E2

3-5 Discussion

The object of this chapter is to show how to extract GPCs out of an experimental table. The results are the clarification of the significant number of generalized physical contradictions that can be extracted for any generalized technical contradiction. In Chapter 2, we presented the number of GTCs that can be extracted from the electrical circuit breaker (117). In this chapter it was illustrated for one of these GTCs that the number of GPCs that can be formulated. It is then a simplified process to imagine the significant number of systems of contradictions with context that can be built for such an example. Thus, if this model of GSC overcomes the limits of classical TRIZ model, several questions will arise including the following:

- What are the limitations, in terms of number of action parameters and evaluation parameters that can be processed in order to extract the GSC in a reasonable amount of time? Several elements necessary to this question were presented in [56].

- If a reasonable number of parameters have to be taken into account, the building of the experimental table is very important. The choice of the considered action parameters is also crucial. An approach, based on the principal component, factor analysis, and fractional design of experiments was presented in [57]. However, their selection and extraction of action parameters is only a practical engineer method without rigor, and therefore, may lack rigor for every situation.

- Finally, one of the most important remaining questions is: if so many contradictions can be formulated, how to choose the one to consider prior to the others for the problem resolution that remains the Q5.

It is interesting to note that out of the 3886 GPC that can be formulated, not all of these GPCs are equivalent. Thus, GPCs can be categorized by the number of conflicting action parameters included in its formulation. For the example, 1743 GPC were found with only one conflicting parameter. This indicates equivalency to the classical TRIZ contradiction under certain conditions as defined by the fixed values for other action parameters. For example, one GPC can be formulated as:

if $(A_2=1)(A_3=0)(A_4=0)$, then $(A_1=1) \Rightarrow ((y_1.y_3)=1)(y_2.y_5.y_6=0))$ and $(A_1=0) \Rightarrow ((y_1.y_3)=0)(y_2.y_5.y_6=1))$.

This practical point is interesting, because in practice these conditions are not provided explicitly by experts when a contradiction holds. Thus, it provides a foundation for Q4 (using the generalized contradictions in the inventive problem solving), and ensures progress to answering Q6 (using the generalized contradictions to express implicit expert's knowledge).

However, the more interesting result, out of this example, is that 533 pairs of items can be considered as non-conflicting pairs as mentioned in 3-2.4. For example, the following pair of items can be extracted out of the two concepts:

$(A_2=1)(A_3=1)(A_4=1)(A_5=1)$ and $(A_1=1)(A_3=1)(A_5=1)$.

This can be recognized as a solution concept defined by $(A_1=1)(A_2=1)(A_3=1)(A_4=1)(A_5=1)$. This concept has to be tested and approved though by the experts. Even if it is not recognized as a solution, at least it will point out important information that was not present in the initial problem formulation. These non-conflicting pairs have a potential to become a method for optimization design of experiments based on each group of action parameters that are involved in non-conflicting pairs vs one action parameter.

Although the BIP model proposed in this chapter is not a common BIP that is its constraints are in the binary field, the model strengthens the understanding of the defined problems, and use an optimization method to identify generalized physical contradiction (answer Q7). In addition, if we know the selection principles for GPC and can transform them into a constraint of the BIP, Q5 how to extract among the

numerous contradictions those which are relevant could be solved. The proposed solution is an exhaustive searching solution that answers Q2.

In the experimental part, four groups of experiments have been performed in which all the experiments are full factorial. They were used to primarily study the relationship between the number of physical contradictions (we consider the contradiction with one conflict AP as a classical physical contradiction) and the number of rows involved in the two concepts. The results answer Q1 and the number of physical contradictions is numerous if there are many experiments involved in the two concepts. The experiments tend toward to a cubic increase with the product of the number of experiments in the two concepts. In the random experiments, there were nine binary APs, when the number of experiments in the two concepts are (25, 25), there are at least 20 classical physical contradictions with context. This result also indicated that when choosing GTCs, we need to select the GTC that is involved with a larger number of experiments.

Chapter 4 Identification of parameters involved in physical contradictions

Given a GTC, an exhaustive GPCs' search algorithm was proposed in a previous chapter. However, when the number of action parameters of the system model is more than twelve, the previous algorithm cannot perform the search problem because of its computing time. The computing time grows exponentially with the number of parameters. Nevertheless, the use of the existing algorithm provides practical evidence that, most often, only few of the model action parameters are involved in the description of the physical contradictions. The purpose of this chapter is to define reduced sets of action parameters that are relevant candidates for physical contradiction (or eliminate those that are not) beforehand. This may allow using the exhaustive physical contradiction search algorithms for systems described by more than twelve action parameters and two values or facilitate the human search as it may also facilitate human search of the physical contradictions. In order to do so, the search of the parameters is stated as a set of classification problems and an adaptation of a Support Vector Machine (SVM) feature selection algorithm is proposed to address them. Limitations of this algorithm are discussed. Finally, strategies for using the proposed SVM algorithm within GPC extraction context are suggested.

4-1 Introduction

As mentioned above the use of the exhaustive contradiction search algorithm presents some practical limitations. Contrary to the evaluation parameters the system model could contain a lot of action parameters and values level used in the experiments. The algorithm computational complexity is $o(M^N)$, where N is the number of action parameters M the number of levels plus 1. That is the reason why quite often the exhaustive GPC search algorithm cannot be used with actual standard computer means. We fixed $M^N < 5{,}3.10^5$; which allows for instance twelve action

parameters with two value levels. But all the action parameters and levels are not necessary for defining the GPCs. This chapter presents a method to bypass this limitation by selecting, when it is relevant, few enough parameters and values so that the previous algorithm can be used; otherwise it will let us know which action parameters and levels are explaining the performance of the actual system regarding each evaluation parameter.

The GPC extraction algorithm [58] presented in Chapter 3 provides all the possible states that can define the concept. The use of this algorithm in real cases leads us to the hypothesis that in many cases only few varying action parameters are involved in a GPC. We develop the hypothesis that there are enough cases where the number of parameters and values involved in the GTC is lower than the limitation of the actual extraction algorithm. Thus, in the case when there is numerous action parameters in the model but only few are necessary for the GPC description, we would like to define them quickly before computing the GPC with the algorithm of Chapter 3. Moreover, we have seen in chapter 3, that in order to define classical or generalized contradictions within a given context it is advised to be able to perform a full design of experiment. This leads us to seek knowing the relevant values of the variables that are involved in the GPCs description such that the number of experiment is reduced and allows performing a good input for the algorithm presented in Chapter 3.

Let us now provide some highlights about the red line that leads us to use feature selection model for addressing the problem stated above. Let us consider an example where there are three action parameters A1, A2, A3 and two evaluation parameters denoted by EP1 and EP2. A set of eight experiments denoted e1, …,e8, the results of which are provided in the binary table in Figure 37, is performed on the system. In this table, “positive one” values of the EP1 and EP2 columns mean that the values of EP1 or EP2 satisfy the requirement for the given experiment. Reversely “zero” values of the EP1 and EP2 columns mean that the values of EP1 or EP2 do not satisfy the requirement for the given experiment. In our case, experiments from E1 = {e1, e2, e3} satisfy EP1 but not EP2; those from E2 = {e4, e5 e6} satisfy EP2 but not EP1; and those from E0 = {e7, e8} do not satisfy any requirement. The sets E1 and E2 define the domain of experiments where the technical contradiction TC1 and TC2 reside. The set E1 can also be interpreted as the set of experiments where the EP1 value is

satisfying against the set E2+E0 and where the EP1 value is not satisfying. Similarly E2 can be understood as the set of experiments where the EP2 value is satisfying against the set E1+E0 where EP2 value is not satisfying.

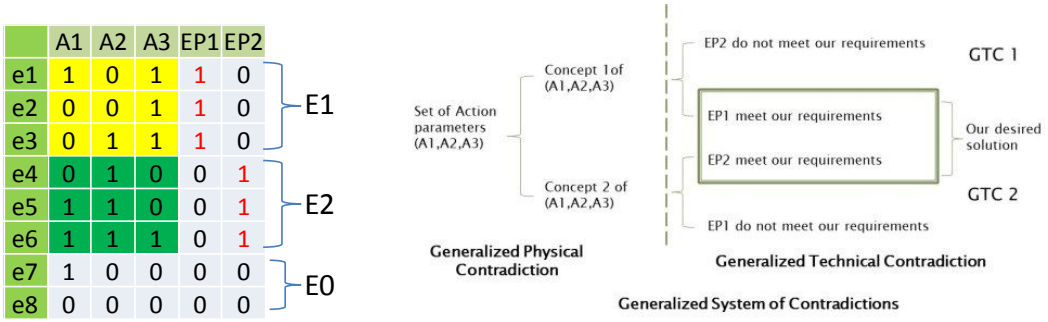


Figure 37 : Binary table and contradiction model

Let us now consider the action parameters. The ones and zero values in the table of Figure 37 refer to two different levels of the actions parameters during the experiments. One can remark that Concept 1 is a function that discriminates experiments from E1 from the remaining set E2+E0. Likewise, Concept 2 is a function that discriminates experiments from E2 from the remaining set of experiments E1+E0. The identification of the parameters that allows classifying elements into two disjoint sets is known as the feature selection problem in several research areas. The proposed search principle of the parameters involved in technical contradictions is based on feature selection for the classifier. The existing feature selection algorithms are good starting points for addressing our action parameter selection problem. The next section provides a quick overview of the feature selection algorithms.

4-2 Algorithms of feature selection: overview and choice

4-2.1 Overview

Feature selection techniques receive a great deal of attention in the areas of pattern recognition and bioinformatics. Specifically with the development of text categorization in [60],[61] and gene expression in [62],[63] whose data size and dimension is dramatically increasing followed by explosive growth of computational complexity, feature selection has become an inevitable pre-processing step in text

analysis and gene analysis. As the name suggests, the feature selection task consists of selecting an optimal subset of features according to a certain criterion in order to improve the accuracy of the classification problem. The tasks also are ensuring the data mining algorithms work faster on lager size data, thereby providing a better understanding of the mined results [64],[65]. The whole process of feature selection depicted in Figure 38 mainly includes three steps: generation procedure, evaluation function and stopping criterion. Generation procedure is to generate a subset of features for evaluation. Evaluation function is a function to evaluate the adequacy of a feature subset produced by generation procedure. The stop criterion decides when to stop by the result from the evaluation function and avoids exhaustively searching the entire subset space. The last step validation in Figure 38 is not a part of the feature selection process itself, its objective is to test the validity of the subset selected by the feature selection algorithm.

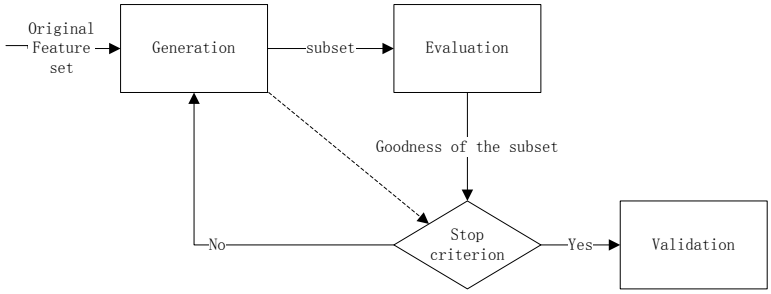


Figure 38: Feature selection process with validation [66]

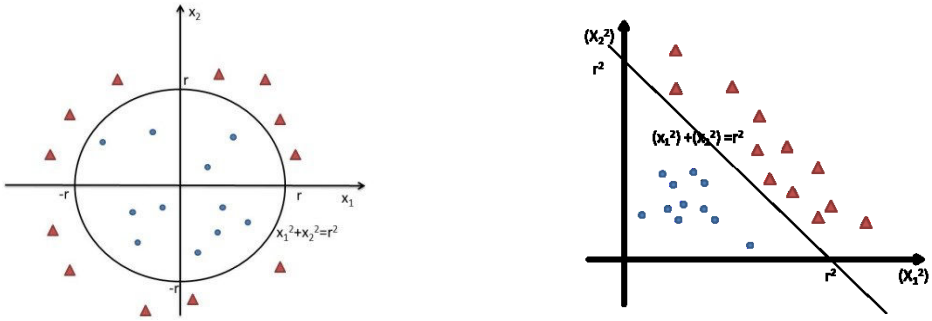
Obviously, the choice of the evaluation measures for determining the best feature or best feature subset plays an important role in the whole process of feature selection. Recently, a current focus in the evaluation function concerns three types of measures. The first one is the information theory-based evaluation. This evaluation is to decrease the uncertainty between features and classes. For example, information gain was used to evaluate and rank the feature in the text categorization [60]. Additionally, mutual information was used in the algorithm MRMR for gene selection [67]. The second scenario is a distance-based evaluation. This evaluation measures the separability of two classes in terms of distance. For example, RELIEF was a feature-weighting algorithm proposed in [68], which evaluates features through investigating the distant difference between two class samples. Furthermore, SVM-RFE was proposed as a gene selection algorithm in the case of cancer gene selection.

The weight vector of SVM can be seen as a description of distance between samples and a hyperplane [62]. The third one is correlation-based measure. For example, the Pearson correlation coefficient is a measure of linear correlation between two variables [65]; thus, it is also a good way to evaluate the contribution from features to classes. The other example is correlation based a feature selection algorithm (CFS) [69], CFS aims at selecting a subset of features, which are highly correlated with the class, yet uncorrelated with each other.

However, the feature selection algorithms proposed previously are not suitable to our selection problem. Through investigating the objective of physical contradiction extraction, feature selection for evaluation parameter aims at selecting the action parameters that are able to make evaluation parameters at satisfied intervals as denoted by ones in Figure 38. Thereby, our purpose is essentially to search a positive-effect subset of the action parameters that is simple enough for GPC extraction. However, most feature selection algorithms maintain both positive and negative features. For instance when searching for concept 1, most feature selection algorithms would provide a set of parameters that allows for classifying the experiments in E1 (i.e., positive features defining concept 1) and in E0 ∪ E2 (i.e., negative features defining an empty concept in our problem); however, we only need the parameters defining concept 1. Subsequently, finding a feature selection algorithm that can be used in our simplification process becomes the key problem of solution. Assigned feature weighting methods can be our solution. The most typical and crucial method is the Pearson correlation coefficient. Its nature is the cosine value of two vectors of samples, which is fitting from the two random variables. Thereby, the Pearson correlation coefficient presents the measurement of angle between two random vectors. Apart from that, SVM weight is another feature weighting algorithm proposed by Guyon in [62]. The weighting is on the basis of the SVM classifier, in which the linear weight vector of the features presents the distance contribution between the sample and the classification hyperplane. Thus, the weight of the features in the SVM classifier can directly be adequate for evaluation. The sign of weight displays the positive or negative effect. A weighting algorithm based on SVM was implemented, customized and evaluated in the following section.

4-2.2 SVM Feature Selection Principle

The Support Vector Machine (SVM) was proposed in [70]; SVM essentially performs linear classification by a non-linear mapping of the original space to a high-dimension space, which is called kernel trick. This non-linear mapping technique helps us look for a classification curve in mapped feature space in which the classification curve is a linear classification plane. For example in Figure 39(a), there are two groups of samples presented by circles and triangles. The classification curve is obviously a circle, $x_1^2+x_2^2=r^2$. However, linear technique is a mature technology with many advantages, such as good computation performance. The problem is how to use linear technique to deal with a non-linear classification. If we develop a mapping $\phi(X)$: (x_1, x_2) ->(x_1^2, x_2^2) in Figure 39(b), in the new coordinate system, the classification line is easy to obtain: (x_1^2) +(x_2^2) =r^2. Returning it to original space, we can find it is the same curve. For SVM, the kernel function is a special non-linear mapping function, which possesses the ability of performing linear classification.



(a) samples in original space

(b) samples in mapping space

Figure 39: Comparison of original and mapping feature space

In [62], SVM as classifier first time contributed to feature selection problem in cancer classification problem, its achievements emphasize on the estimation of each gene, and adopted recursive elimination to rank features (SVM Recursive Feature Elimination, SVM-RFE). To be specific, the estimation of SVM is based on the weight coefficient of linear classifier in transformed space. For interpreting the weight coefficient, we assume that the linear classifier has the general form of $\sum w_i x_i + b$, where w_i is the weight coefficient and x_i is the feature value. Apparently, if take the

sample point in a linear classifier, we can find that the sign of the linear classifier decided the class of the sample. Moreover, for each unit of the feature value, if the weight coefficient is large, it can determine the sign of the classifier. For example in Figure 39(b), the weight vector is (1,1), which means that x_1^2 and x_2^2 have the same positive influence on the sign of decision function($x_1^2 + x_2^2 – r^2$).

For a simple explanation, the application of SVM model in non-linear separable problem is given by a two-class instance set of $(x_i, y_i), i = 1,...,l$ where $x_i \in R^N$ and $y_i \in \{1, 0\}$, y_i is class label. SVM searches for a linear classifier $D(x) = W^T \phi(x) + b$ in higher dimensional Hilbert space by utilizing a higher dimensional mapping function $\phi(x)$, where the two-class samples are linearly separable. As long as $D(x)$ is obtained, for a new instance X, the class label y is determined by the sign of $D(X)$:

$$D(X) \Rightarrow \begin{matrix} > 0 \rightarrow y = 1 \\ < 0 \rightarrow y = 0 \end{matrix}$$

Technically, the higher dimensional mapping function $\phi(x)$ used in SVM to induce the inner product function called Kernel function and it is defined as $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$. Four basic and most common used kernel functions are the followings[71]:

- Linear kernel function: $K(x_i, x_j) = x_i^T x_j$.
- Polynomial kernel function: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0.$
- RBF kernel function: $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|)^2, \gamma > 0.$
- Sigmoid: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r), \gamma > 0.$

After defining a higher dimensional function $\phi(x)$ or a kernel function $K(x_i, x_j)$, SVM training problem is transformed into the following optimization problem to search for the maximum-margin hyperplane $D(X)$:

$$\underset{W,b,\xi}{Min} \ \frac{1}{2} W^T W + C \sum_{i=1}^{l} \xi_i$$

$$\text{Subject to } y_i\left(W^T\phi(x_i)+b\right)\geq 1-\xi_i; i=1,2,...,l,$$

$$\xi_i \geq 0; i=1,2,...,l$$

In this optimization problem, W is the weight vector of final decision function $D(x)=W^T\phi(x)+b$, b is the constant term. $\xi=(\xi_1,\xi_2,...,\xi_l)$ are non-negative slack variables for non-separable situation. C is the penalty parameter for presenting the weight of an outlier in non-separable situation. Each constraint is built from the given instance. The weight vector W is the evaluated value of the features. Regarding the optimization problem solving, it can be transformed to a convex quadratic programming problem and be solved [72].

4-2.3 Feature selection method adopted in our problem

In practice, we applied the principle proposed in [71] for building up a SVM model for feature selection of the evaluation parameters based on the following steps:

1. Transform the table of experiments into to the format of a SVM problem model.
2. Empirically select kernel function. As the data are obviously nonlinear separable, we consider utilization of RBF kernel.
3. Use cross-validation to find the best state of SVM according to the accuracy.
4. Training the SVM by whole training set (x_i, y_i), $i=1,...,l$.
5. Output the weight W as the evaluation of each feature.

In regard to the SVM recursive features and elimination algorithm in [62], the weight value of W is used for a measure of ranking. It removes the smallest terms of absolute value after each training SVM, then the inverted order would show on the order of removing. Interpretation of W as an evaluation of feature: throughout linear decision function $D(x)=W^T\phi(x)+b$, the weight of each component of x plays a significant role in decision making, if absolute value of weight is large, the correspond feature x_i have a large contribution in the sign of $D(X)$, which denotes the final decision. In the next section, we will present how to interpret the weight for

ranking the action parameter and understand the meaning of positive and negative weight based on the Decision function $D(x)=W^T\phi(x)+b$.

Based on this theoretical model we did develop tools for ranking the APs explaining the performance of each EP by separating the performance in two classes: good and others. This allows building a strategy of selecting APs based on the EPs. This model also allows using the two sets of two classes defined by a given GTC, and building a second strategy of selection of the APs. Next it allows for providing a core algorithm that is presented for ranking the AP related to each EP. The practical rules for using the outputs of this algorithm are then discussed. The same algorithm will be used further in this chapter for ranking the APs related to a given GTC. Moreover, this algorithm will be applied for selecting and ranking the values of each AP. Finally we will show two strategies for using this algorithm within the GTC extraction context.

4-3 Proposed core SVM algorithm for ranking the APs related to each EP

This section shows how the SVM feature selection methodology was applied to provide a clustering and a ranking of the action parameters. This will allow the user to select the APs that can reach the desired value for each EP separately. The clustering separates the AP in two groups of AP: the APs that positively influence EP towards the desired result, and second, the APs that negatively influence EP towards the desired result. If we are only concerned regarding the influence without positive and negative, the set of APs can be ranked using the more influencing parameter to the less influencing parameter because of weights provided by the algorithm. The way to use the output for our problem is discussed in section 4-4.

We proposed to select RBF kernel and use an existing C++ library for this kernel [71], we have to shape the data for using it. The general SVM function can be depicted as seen in Figure 40.

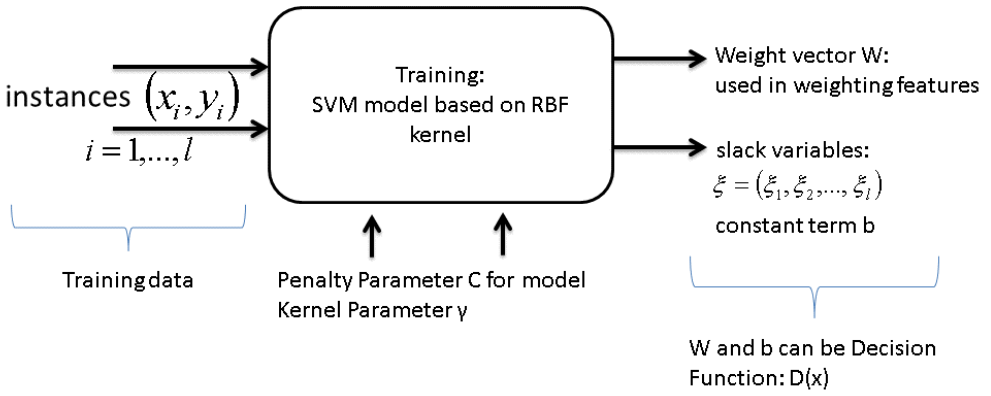
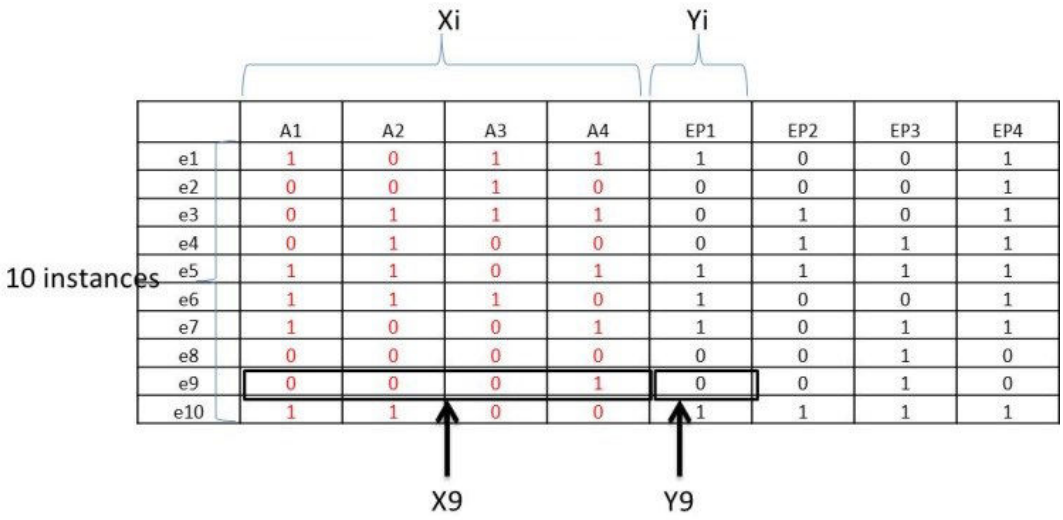


Figure 40: General SVM model

To summarize, the values of Xi and Yi are provided as inputs and then C and γ are defined, The SVM will provide the weights that are useful for our purpose. It also provides other outputs (b and ξ) that are impractical for our purpose.

The remaining section explains and illustrates how to apply the generic process for our case in five steps as depicted in section 4-2.2 for our case. We shall illustrate the general procedure with the example of Figure 41. In this case there are four action parameters (A1,A2,A3,A4), four evaluation parameters (EP1, EP2, EP3, EP4), and ten experiments (e1,e2,…,e10).



	A1	A2	A3	A4	EP1	EP2	EP3	EP4
e1	1	0	1	1	1	0	0	1
e2	0	0	1	0	0	0	0	1
e3	0	1	1	1	0	1	0	1
e4	0	1	0	0	0	1	1	1
e5	1	1	0	1	1	1	1	1
e6	1	1	1	0	1	0	0	1
e7	1	0	0	1	1	0	1	1
e8	0	0	0	0	0	0	1	0
e9	0	0	0	1	0	0	1	0
e10	1	1	0	0	1	1	1	1

Figure 41: An example for AP selection

In this section we focus on a procedure providing the APs related to one EP. Thus, we use an example EP1. We would like to determine the APs required to be maintained for explaining the "1" values of EP1 and rank them. Both clustering and ranking will be provided by the weights provided by the SVM feature selection procedure. Indeed

the algorithm will provide a weight for each AP. In the general case, the weights will vary in a range of negative and positive values. The AP with large absolute value of weights will explain the "ones" values of EP1 and are a good candidate to be retained; others can be removed. The ranking of the AP based on from the highest absolute value to the lowest absolute value provides a rank of importance of the APs.

The first step as described in 4-2.2 is to transform the data into the shape to be analyzed with a SVM model in general. In our case the AP are already in a good shape for being processed, so is EP1 as it has only two values. These two values allow separating the experiments in two groups. As depicted in Figure 41 we have provided the Xi and Yi vectors related to the input data. In our example the Xi and Yi are given by the all the rows and the columns A1 to EP1 of the Figure 41. For instance X9, Y9 is highlighted in the Figure 41. Thus vector x_i represents the values of the action parameters for a given experiment e_i and y_i is the value of the studied EP for experiment e_i. The number of action parameters is denoted as N; the number of experiments is denoted as *l*. Thus, in our example l=10 and N=4.

The second step is to adjust model parameter C and γ such that our SVM model has a high cross-validation accuracy. In the case of the RBF kernel, the parameter C and γ are unknown beforehand for a given problem. In order to search the best value for (C, γ), the common strategy is to randomly separate the l instance into k groups: the first k-1 group was called the training groups where we know whether the evaluation parameter is satisfied or not. The last one group called the testing group where the Evaluation Parameter value is unknown. For a given evaluation parameter, the training groups provides the decision function $D(X)$ that allows separating its experiments into two sets: the subset of experiments in which the value of evaluation parameter is 1 (satisfied) on the one hand, and the set of experiments in which the value of the evaluation parameter is -1 (dissatisfied). Then, $D(X)$ is used for classifying the experiments of the testing group. The quality of $D(X)$ is measured by the prediction accuracy i.e. the percentage of correctly classified experiments of the testing group. When prediction accuracy is not satisfied, we can change parameters (C, γ). The 'grid-search' recommended for a better (C, γ) as discussed in [71], for

example C=2^{-5}, 2^{-3},2^{-1} ,…2^{5}, γ=2^{-5}, 2^{-3},2^{-1} ,…2^{5}. In that step, at least an acceptable accuracy for cross-validation should be more than 80%. The whole process is depicted in Figure 42.

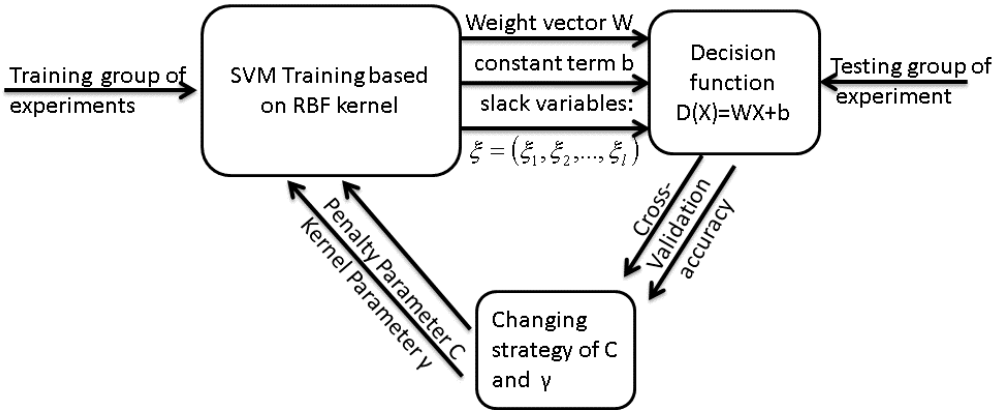


Figure 42: Cross-validation process for RBF kernel SVM

Once the system parameter (C and γ) is determined, the SVM is trained by an entire set of experiments, and the output the weight vector W, constant term b, and non-negative slack variables $\xi = (\xi_1, \xi_2, \ldots, \xi_l)$. The weight vector W is the evaluation of AP for EP1.

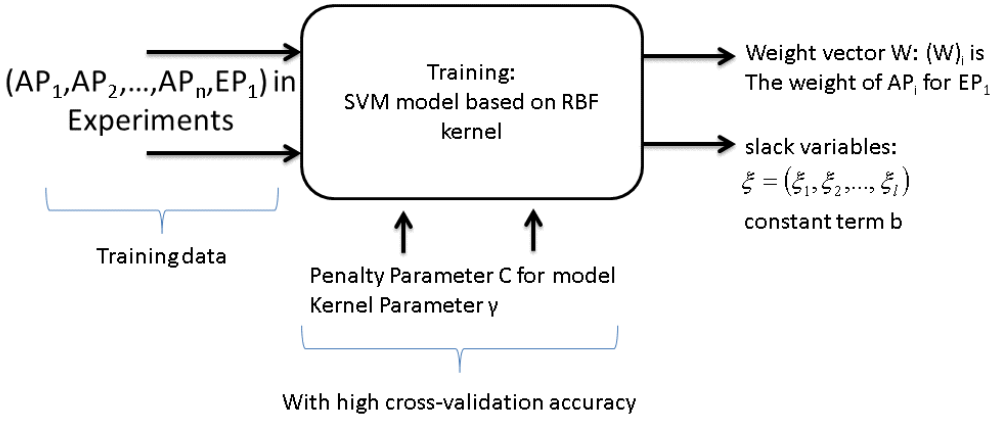


Figure 43: SVM training process

When performing the above process for EP1 on the example of Figure 41, the instances for EP1 are (A1,A2,A3,A4,EP1) of experiments (e1,e2,e3,e4,e5,e6,e7,e8,e9,e10). Then three-fold cross-validation was performed, and when C=8 and $\gamma = 0.1$, the cross-validation accuracy is 100%. The weight vector evaluating the AP for EP1 is provided in the EP1 row of Table 16. When repeating the

same process for EP2, EP3 and EP4 the weights for each EP are determined as shown in Table 16.

Table 16: Weights of Action Parameter for Evaluation Parameter

Weight		Action Parameters				3-fold cross-validation
		A1	A2	A3	A4	
Evaluation Parameters	EP1	12.05	0.1	0.08	0.1	100%
	EP2	-0.48	11.6	-0.56	0.48	90%
	EP3	0.03	-0.03	-12.12	-0.03	100%
	EP4	3.15	2.42	2.42	-0.85	80%

The practical question is how to interpret the results in order to select a few APs. This question will be analyzed in detail in section 4-4. Next paragraphs provide what can be illustrated straightforward from the result of Table 16 the absolute value of weight measures the importance of an action parameter on the evaluation parameter. The higher the absolute value, the more important the action parameter is for the EP.

First consider the weights of EP1. All of them are positive values; thus, it seems that theoretically all action parameters have a positive effect on satisfying EP1. However, when looking at the values of the weight it can be seen that A1 is more than 100 times the weights of A2, A3 or A4. The question then arises, should we keep A2, A3 and A4? For EP2 the weights related to A2 and A4 have positive values. However, for the previous case, the weights are very different. The weight of A2 is 11.6, which is more than 20 times the weight of A4 (0.48). Should we keep A4? In order to address this type of question we used an experimental process with the results provided in section 4-4.

Another situation for EP2 is that A1 and A3 have negative weights; thus, they try to keep the EP2 from the “one” values (satisfied). The question is whether we should remove them because they stop the EP from being satisfied. Especially for EP3, A3 has the largest negative influence. The understanding of negative weight can be understood from the output of SVM machine-decision function D(X). As mentioned in 4-2.3, $D(x)=W^T\phi(x)+b$ is a linear separation plane for two classes, one is above the plane and the other is below the plane. In practice, if we take the instance to

D(x), the sign will show if it is above or below the plane. Thus, the weight $(W)_i$ describes the unit contribution of the feature $(x)_i$ to the sign of D(x). For our problem, we want to know how much unit contribution to the class signed positive. How negative weight influences satisfying EP is how the negative weight influences D(X) to be positive. From a mathematical point of view, if the weight is negative we develop the feature value to be the smallest one. It has the positive contribution to the sign of D(X), i.e., minimizing the negative is maximizing the positive. For example, A3 = {0, 1} has largest negative influence on EP3 and we find that A3 = 0 play a decisive role in satisfying EP3. To sum up, the absolute value of weight should be the evaluation criterion for deciding whether the action parameter is acting positively (towards our objective for the evaluation parameter).

4-4 Evaluation of the core SVM algorithm for choosing APs

In order to discuss the limitation of the SVM based method we proposed, we need to answer the following questions. How can we recognize adequate parameters based on the weights of the SVM? In a theoretical point of view, the parameters that have influence should have positive weight and be proportional. However, in the reality, the sample has large effects on the results of weights. Hence, we must consider the impact from a sample.

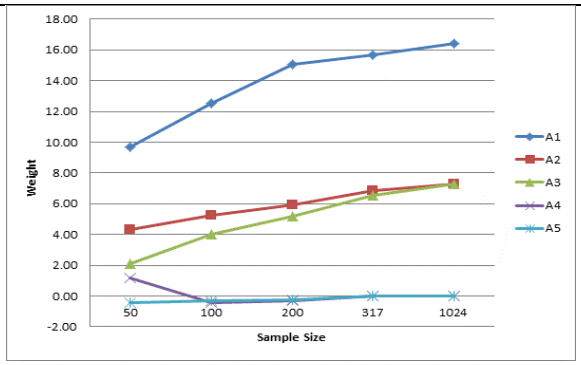
Before answering questions, we have to ensure the reliability of weights, i.e., under these weights the SVM has a low probability of misclassification. In the aspect of theory, the reliability is measured under the theoretical framework of probable approximately correct learning [72]; however, in reality, the upper-bound is very loose and cross-validation is a practical method. Hence, for the first question we only discuss the recognition capability of reliable weights, whose three-folder cross-validation accuracy is more than 85%.

First, let us imagine what kind of parameter should have a large positive influence. There are two situations when the parameter has a strong discriminating ability of a positive class. One is the when the parameter is associated to a more general states of a system such as (A1=1) and (A2=1)(A3=1), which are two states in concept. They are the unique two states associated with parameters A1, A2, and A3. Hence A1 should be a more discriminative parameter than A2 and A3. The other situation is if

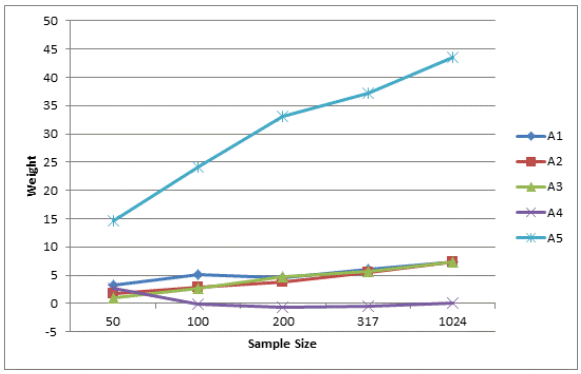
the parameter that associates with more states should be a more discriminative parameter, i.e., (A1=1)(A2=1) and (A2=1)(A3=1) are the only two states that are associated to parameter A1, A2, and A3. For this case, A2 should be more discriminative than A1 and A3. In general, the more discriminative parameter should have a larger weight.

Thereby, the first groups of experiments were performed with ten action parameters with two values (1 or 0). Thus, the full experiment had 1024 instances. For the test evaluation parameters, eight concepts were selected, and the weights of ten APs for each concept would be calculated. On the aspect of sampling of experiments, we adopted a random selection based on uniformed distribution. In order to make a comparison of weights because of differences between different levels of action parameters, the weights from four concepts are presented in Figure 44. All of the weights were considered reliable as the three-fold cross-validation accuracy was more than 85% for each group of sample, with many higher than 95%. In order to observe the impact from a sample, we varied the sample size for each concept.

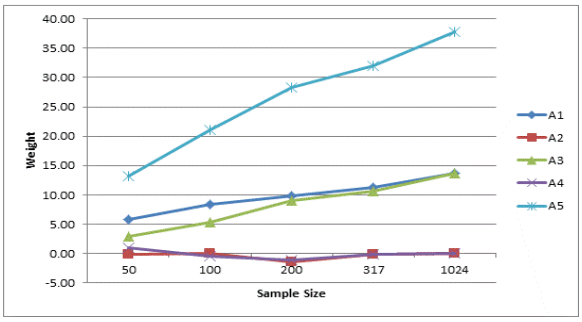
Through observing the figure, we can draw a conclusion that the more discriminative the parameter is, the larger weight it should be assigned. For example, in Figure 44 (a), A1 apparently is the most discriminative parameter; thus, it has largest weight. A4 and A5 are undiscriminating parameters, so their weights tend to be around 0. This conclusion indicates that the weights allow us to recognize the significant positive influence of the APs, and also reflects how much positive influence the parameter has based on its value.



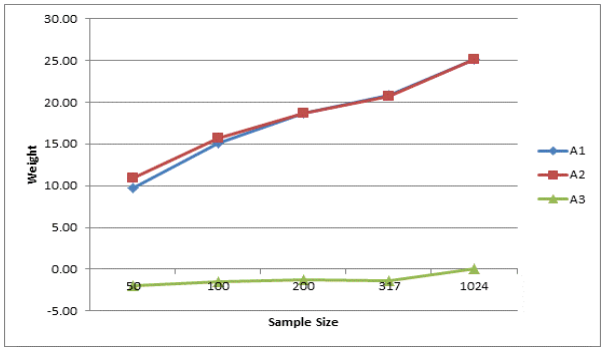
(a) (A1=1)(A2=1)+(A1=1)(A3=1)



(b) (A1=1)(A2=1)(A3=1)+(A5=1)



(c) (A1=1)(A3=1)+(A5=1)



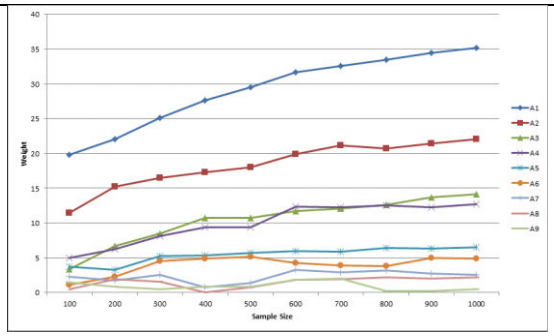
(d) (A1=1)+(A2=1)

Figure 44: Weight values for different concepts with different sample size

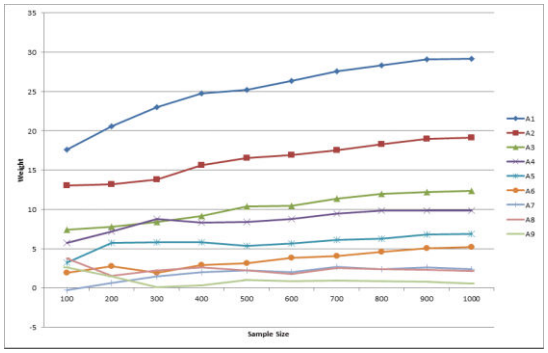
From Figure 44, we could also obtain another result with the increasing of the sample size where the weight difference between different discriminative-level of parameters was rising. For instance, in Figure 44(c), we can see that the gap of weights between A5 and A1 for the same sample size were expending with an increasing of sample size. This result suggests to us the sample size has a positive effect on intensifying the discriminative ability of the positive influent parameters. However, another problem arose about how many experiments we needed to perform to ensure that the weight difference is large enough to be recognized. This problem resulted in a second group of experiments where the 20 APs, 17 APs, 15 APs and 13 APs with binary value were defined. That meant the experimental space was

2^{20}, 2^{17}, 2^{15}, 2^{13}. Selecting a representative concept = (A1=1) + (A2=1)(A3=1) + (A2=1)(A4=1) + (A3=1)(A5=1)(A6=1) + (A4=1)(A5=1)(A7=1)(A8=1). From the theoretical point of view, the discriminative ability ranking is A1 > A2 > A3 > A4 > A5 > A6 > A7 = A8.

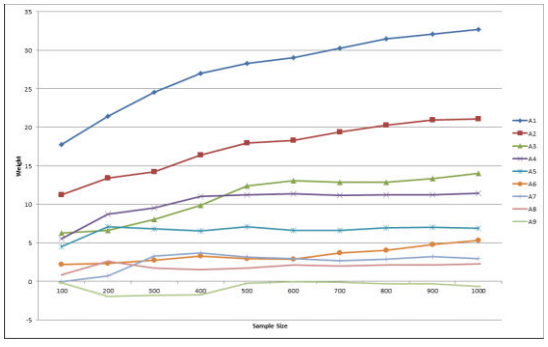
However, the practical result suffered from the effect of the sample size. In Figure 45(a), when the sample size was small, the weight of some AP was not evident. However, following the increase in the sample size, their positive discriminative ability was clear. When the sample size was large enough, their ranking followed the theoretical analysis (A1 > A2 > A3 > A4 > A5 > A6 > A7 = A8 and A9 had no effect).



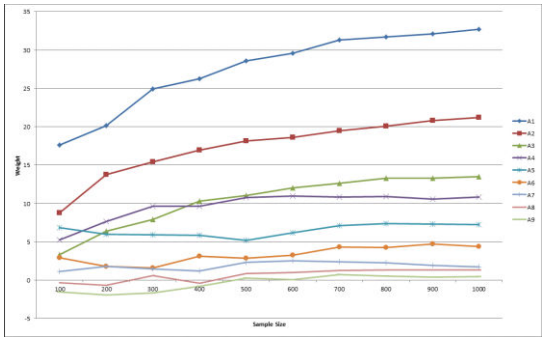
(a) Weight with different sample size (20 APs)



(b) Weight with different sample size(13 APs)



(c) Weight with different sample size (15 APs)



(d) Weight with different sample size (17 APs)

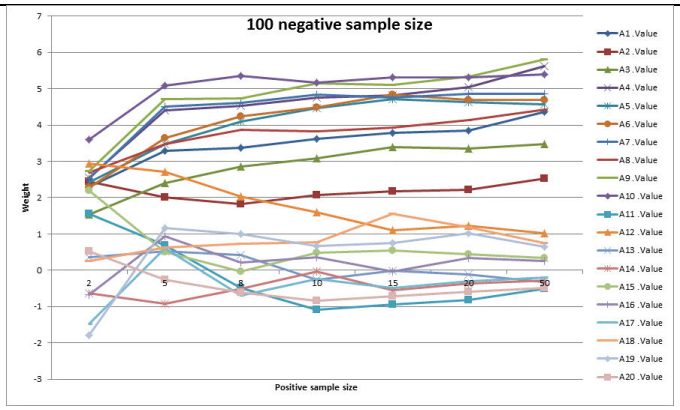
Figure 45: Weight values for different numbers of AP

(Concept=(A1=1)+(A2=1)(A3=1)+(A2=1)(A4=1)+(A3=1)(A5=1)(A6=1)+(A4=1)(A5=1) (A7=1)(A8=1))

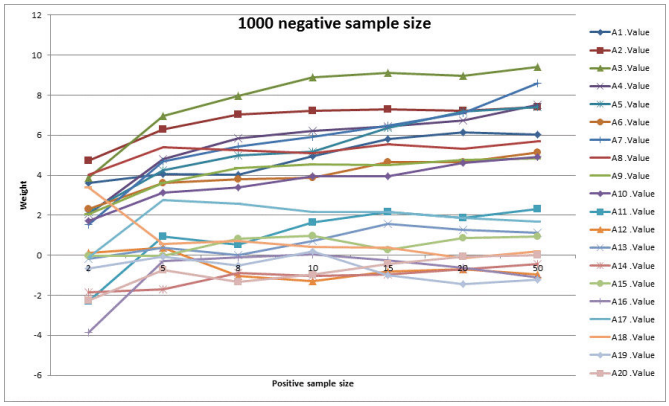
From the analysis of Figure 45, we can summarize three points. First, with the increase of the sample size, the differences between the weights of the positive influent APs are stable and clear. Secondly, even when the sample size is small such as 100 or 200, the weights of relatively high positive influent level of AP are large. Thirdly, the weights of the positive influent AP tend to increase with the rise of sample size.

The above experiments only reveal the impact from sample size. In some situations, the positive instances are very few; thus, it is instructive for sampling when we are aware how many positive instances are required to clearly discriminate positive action parameters. The following experiments should test the impact from a number of instances. Twenty APs were defined and 12 groups of experiments were performed for the same concept (A1=1)(A2=1)(A3=1)…(A10=1). Associated APs have a small influence while each group has different number of negative instances. We varied the number of positive instances in each group. Four groups of results are shown in Figure 46.

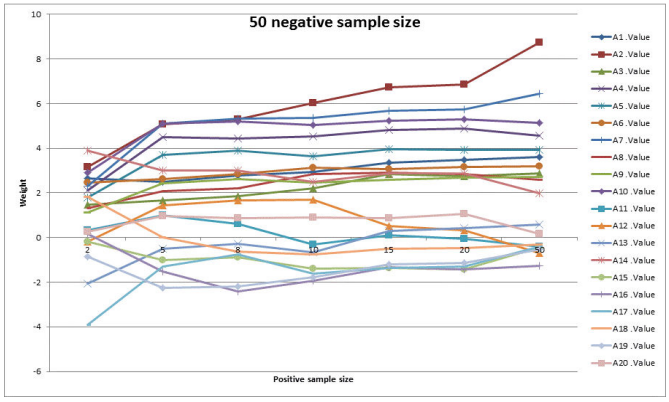
If we carefully observe when negative instance is over 100, with the increase of the number of positive instance, there is a difference between the two groups of AP (A1, A2,…, A10 and A11, A12,…, A20). The two groups of AP tend to separate from each other. However, in Figure 46(c), A14 goes to the influent group, and in (d) A9 goes to the no influent group. That means the number of negative instances should be more than 100 to promise the correct separation. In addition, twenty positive instances were necessary to separate the two groups.



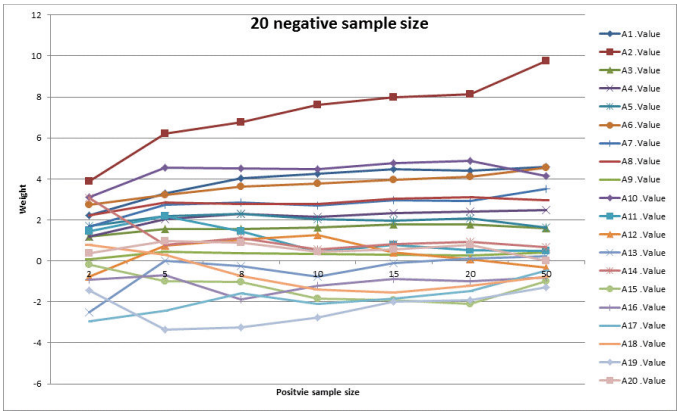
(a)100 negative



(b)1000 negative



(c)50 negative



(d)20 negative sample

Figure 46: Weight variations for different number of negative and positive instances

In summary, the experiments do not only reveal how weight discriminate positively influences APs, but it also indicated that the sample size does not have an impact on the highly discriminative APs, whereas, a large sample size is necessary for the weakly discriminative APs. Finally, a minimum number of positive and negative instances are necessary for weakly influent AP.

The analysis of this section will help define the strategies proposed in section 4-6. However, before the next section, we describe how to use the core SVM algorithm for defining the value of the action parameters that are involved in concepts of the general physical contradiction.

4-5 Applying the core SVM algorithm for ranking the values of APs

Ranking and understanding the value of an action parameter generally comes after ranking and selecting the action parameters of a GPC concept. The ranking is also based on the weight of SVM model. In order to weight the value of an action parameter, we need to make each single value of action parameter become single action parameter. The best way is to code each value of action parameter as a binary value. For instance, A1 has two values, A1 = 1 and A1 = 0, in the experiments. We code it into two mutual exclusive action parameters, ‘A1 = 1’ and ‘A1 = 0’. If A1 = 1, the parameter ‘A1 = 1’ is equal to 1, and ‘A1 = 0’ is equal to 0. If A1 = 0, the parameter ‘A1 = 1’ equal to 0, and ‘A1 = 0’ is equal to 1. Using this method, the experimental results in Figure 41 became the result in Figure 47. The weight vector was then calculated by the same approach in 4-3, the weights result is displayed in Table 17.

	A1=1	A1=0	A2=1	A2=0	A3=1	A3=0	A4=1	A4=0	EP1	EP2	EP3	EP4
e1	1	0	0	1	1	0	1	0	1	0	0	1
e2	0	1	0	1	1	0	0	1	0	0	0	1
e3	0	1	1	0	1	0	1	0	0	1	0	1
e4	0	1	1	0	0	1	0	1	0	1	1	1
e5	1	0	1	0	0	1	1	0	1	1	1	1
e6	1	0	1	0	1	0	0	1	1	0	0	1
e7	1	0	0	1	0	1	1	0	1	0	1	1
e8	0	1	0	1	0	1	0	1	0	0	1	0
e9	0	1	0	1	0	1	1	0	0	0	1	0
e10	1	0	1	0	0	1	0	1	1	1	0	1

Figure 47: Coding value of action parameters into binary

Table 17: Weights of value of action parameter for evaluation parameter

Weight		Action Parameter								Cross-validation
		A1=1	A2=1	A3=1	A4=1	A1=0	A2=0	A3=0	A4=0	
Evaluation Parameter	EP1	7,2	0,1	0,1	0,1	-7,2	-0,1	-0,1	-0,1	100%
	EP2	-1,3	6,5	-2,4	1,3	1,3	-6,5	2,4	-1,3	90%
	EP3	-1,9	-0,2	-7,3	1,9	1,9	0,2	7,3	-1,9	80%
	EP4	1,3	1,5	1,5	-0,2	-1,3	-1,5	-1,5	0,2	80%

The interpretation of the weight for the value of the action parameter is the same as that for the weights of the action parameter. Two groups of values are formed by the weights: positives (i.e., for EP1: A1 = 1; A2 = 1; A3 = 1; A4 = 1) and negatives (i.e., for EP1: A1 = 0; A2 = 0; A3 = 0; A4 = 0). In our example, the first group is the value with the positive weights, which should make the APs denoted “A1 = 1”;”A2 = 1”;”A3 = 1”;”A4 = 1” equal 1 to satisfy the evaluation parameter EP1. In our example, the second group is the value with negative weight, which should make the APs denoted “A1 = 0”;”A2 = 0”;”A3 = 0”;”A4 = 0” equal to 0 for satisfying evaluation parameter EP1. For example, in Table 17 ‘A1=1’ has a positive weight 7.2 for EP1; thus, as proposed in 4-3 ‘A1=1’ should be 1, i.e., the value 1 of A1 should be kept to satisfying the EP1. Reversely, ‘A1=0’ has a negative weight -7.2 i.e. ‘A1=0’ must be 0 to satisfy EP1. This means A1 should not equal 0 and the value 0 of A1 should be removed. To sum up, for AP values ranking, negative weight values are unusable and should not be considered. The selection of the value(s) of an action parameter should be performed among the positive weights. Again, the higher the positive weight, the more important the value is in order to reach the objective assigned to the EP.

4-6 Using the core SVM algorithm in context of GPC extraction

To facilitate the GPC extraction algorithm proposed in Chapter 3, the key problem is how to reduce the number of action parameters so as to decrease the time complexity of GPC extraction algorithm. Thanks to SVM algorithm, it provides us a method to weight the action parameters, and select those that make a significant contribution to satisfy the evaluation parameters. In this section, we will propose two strategies to reduce the number of action parameters and their number of different values based on SVM algorithm for GPC extraction. The first strategy is based on the information provided by the GTC, where we weight the action parameters for the two sets of evaluation parameter Y1 and Y2 (not Y0) as defined by the GTC. The second is based on the information provided by each evaluation parameter, where we weight the action parameters for each evaluation parameter.

4-6.1 GTC based extraction

As Chapter 3 described, GTC has two set of evaluation parameters, Y1 and Y2, such as in Table 18, Y1 = {y1, y3}, Y2 = {y2, y5, y6}.

As described in Table 18, GTC divides the experiments into E1, E2 and E0. Y1 is satisfied by $\mathrm{E1}$, and rejected by $\mathrm{E2} \cup E0$. Y2 is satisfied by $\mathrm{E2}$, and rejected by $\mathrm{E1} \cup E0$. Our purpose is to select the most positive weighted action parameters and their values. These will have the most positive effect on satisfying the evaluation parameters of Y1 or Y2, and then extract the GPC from the selected action parameters and its values.

Table 18: Experiment result of electrical circuit breaker

							Y1		Y2			Y0
		A1	A2	A3	A4	A5	y1	y3	y2	y5	y6	y4
E1	e1	1	1	0	0	1	1	1	0	1	1	1
	e3	1	0	1	0	0	1	1	0	0	0	0
	e4	1	1	0	0	0	1	1	1	0	0	1
	e5	1	0	1	0	1	1	1	0	1	1	0
	e7	1	0	1	1	0	1	1	0	0	0	0
E2	e2	0	1	1	1	1	0	0	1	1	1	0
	e6	0	1	0	1	2	0	0	1	1	1	1
	e9	0	1	0	0	2	0	0	1	1	1	1
E0	e8	1	0	0	0	1	1	0	0	1	1	1

The class label is not one evaluation parameter but the evaluation parameter set Y1 and Y2, as Y1 is satisfied by the experiments of $\mathrm{E}1$, and rejected by experiments of $\mathrm{E}2 \cup E0$. Thereby, for class label $y_i \in \{1, 0\}$, we label the experiments in $\mathrm{E}1$ as 1, and experiments in $\mathrm{E}2 \cup E0$ as 0. Then SVM should provide the action parameters involved in concept 1. The same thing is for Y2 where for the class label is $y_i \in \{1, 0\}$, and we label the experiments in $\mathrm{E}2$ as 1 with the experiments in $\mathrm{E}1 \cup E0$ as 0. Then, SVM should provide us the action parameters involved in Concept 2 of the system of contradictions. After defining the experiments and class label as well as, instances of (x_i, y_i), we could start our process, which is presented in Figure 48.

In a brief summary of the process in Figure 48, the first four steps are in regards to the selection of the action parameters and their value by SVM weights. The output is a reduced set of action parameters and values, which facilitate GPC extraction. As GPC extraction needs full factorial experiments for good context definition, the inputs experiments of process commonly cannot cover full factorial experiments on a reduced set of action parameters for the system. This leads towards performing a new group of experiments as shown in Step 5. For the new experiments, we do not know whether the targeted GTC still can be extracted. Thus, we extract GTC for the new experiments in order to evaluate our simulation model where many parameters are fixed. Our purpose is to verify that at least the GTC we wanted to study remains in the experiments. Alternately, we can choose a target GTC for GPC extraction. We can also make a comparison with the original GTC set to estimate our parameter selection. If our target GTC is not involved in the new GTC set, which means we make a bad selection regarding parameters and or values, we return to select additional parameters and values out of the ranking provided by the SVM algorithm. Otherwise, when the targeted GTC is present, we extract the GPC to form a generalized system of contradiction. In the following, we illustrate each step by an example.

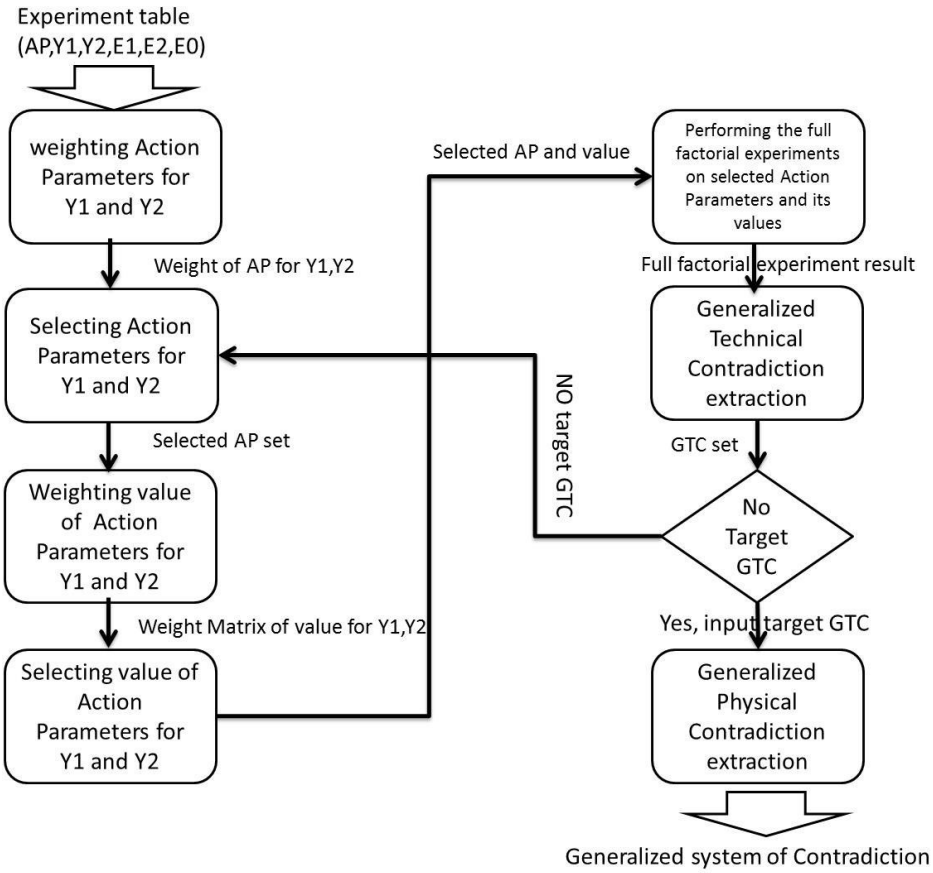


Figure 48: Process of GTC-based extraction

The first step is the weighting action parameter for Y1 and Y2. We develop an example in Appendix 1 of 316 experiments that were randomly generated; ten binary action parameters and four evaluation parameters. In order to explain the process, we took an example where the action parameters and values are known. The concepts for evaluation parameter are presented below:

EP1=(A1=0)(A2=1)(A3=1) EP2=(A2=1)(A3=1)(A4=0)

EP3=(A2=1)(A3=0)(A5=0) EP4=(A3=0)(A5=0)(A6=1)

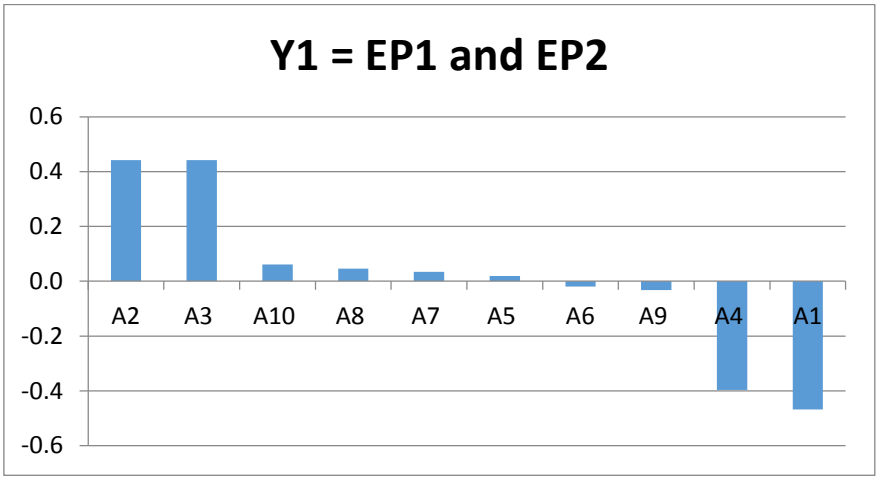
EP1=(A1=0)(A2=1)(A3=1) means that when (A1=1) and (A2=1) and (A3=1), the evaluation parameter 1 is satisfied, and denoted by 1.

The target GTC is presented by the shaded cells where (Y1 is {EP1,EP2}) vs (Y2 is {EP3, EP4}) in Appendix 1. We perform the first step of weighting the action parameter to Y1 and Y2, with the weights presented in Table 19.

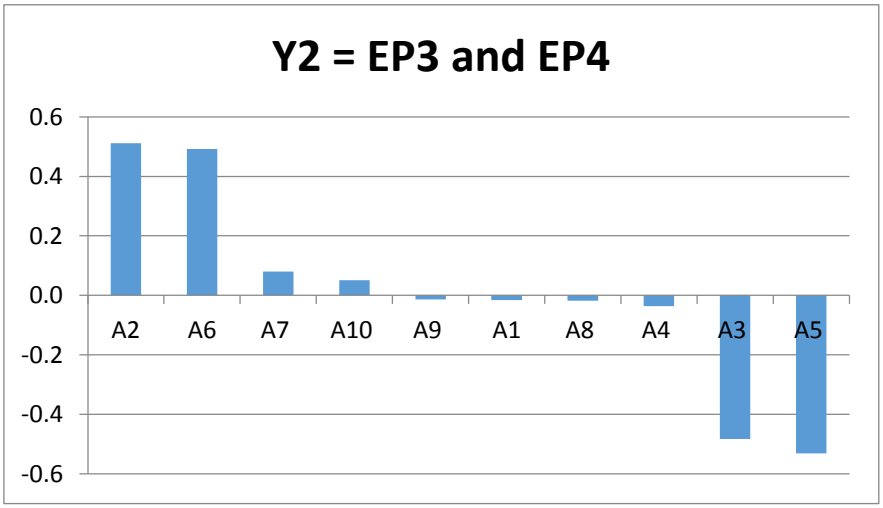
Table 19: Weights for 10 action parameters

Weight		A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	3 fold Cross-validation
Y1	EP1 and EP2	-4,7	4,4	4,4	-4,0	0,2	-0,2	0,3	0,5	-0,3	0,6	100%
Y2	EP3 and EP4	-0,2	5,1	-4,8	-0,4	-5,3	4,9	0,8	-0,2	-0,1	0,5	100%

In order to facilitate selection and observation of the impact of action parameters, we present the weights in bar charts in Figure 49.



(a) Weights of action parameter for Y1



(b) Weights of action parameter for Y2

Figure 49: Bar chart of weights for action parameters for Y1 and Y2

The second step is the selection of the most affected action parameter set for Y1 and Y2 by weight value. There are several selection principles:

- As explained in 4-3, the evaluation of the action parameter is from the absolute value of weight. For example, for Y1 the most effective action parameter is A1 according to the absolute value of its weight.

- The most effective action parameter must be selected, for example in Table 19, A1 and A5 must be selected to Y1 and Y2.
- Select an action parameter depending on the purpose of trying to cover the 2nd and 3rd effective action parameter for each evaluation parameter. For example, A2 is the 2nd effective action parameter for Y1 and Y2.
- The number of selected action parameter should be small to suit the capacity of the GPC extraction algorithm.

In the example, the selection set are S = {A1, A2, A3, A4, A5, A6} = {A1, A2, A3, A4} U {A2, A3, A5, A6}, where {A1, A2, A3, A4} is from Y1 (for Concept 1 of contradiction), and {A2, A3, A5, A6} is from Y2 (for Concept 2 of the contradiction).

This example could be performed directly with the GPC algorithm of Chapter 3. However, we only performed the value selection for action parameter set S.

The third step is the value selection, as explained in 4-5. We removed {A7, A8, A9, A10} from the experiment in Appendix 1, by coding the value of {A1, A2, A3, A4, A5, A6} to be a new experiments table with twelve action parameters {A1=1,A1=0,….,A6=1,A6=0}, four evaluation parameters {EP1, EP2, EP3, EP4}. The weights are shown in Table 20.

Table 20: Weight matrix of values of action parameters for the example

weight		Action Parameter												3- folderCross-validation
		A1=1	A2=1	A3=1	A4=1	A5=1	A6=1	A1=0	A2=0	A3=0	A4=0	A5=0	A6=0	
Y1	EP1EP2	-1,7	1,7	1,7	-1,7	0,0	0,0	1,7	-1,7	-1,7	1,7	0,0	0,0	100%
Y2	EP3EP4	0,0	1,7	-1,7	0,0	-1,7	1,7	0,0	-1,7	1,7	0,0	1,7	-1,7	100%

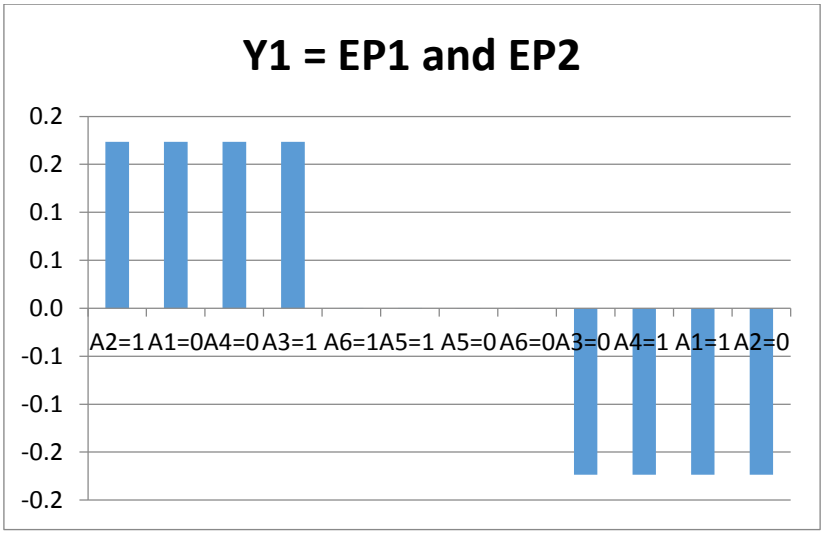
As weights of action parameters, the bar chart for Y1 and Y2 are shown in Figure 50.

Regarding the value selection, we also followed several principles:

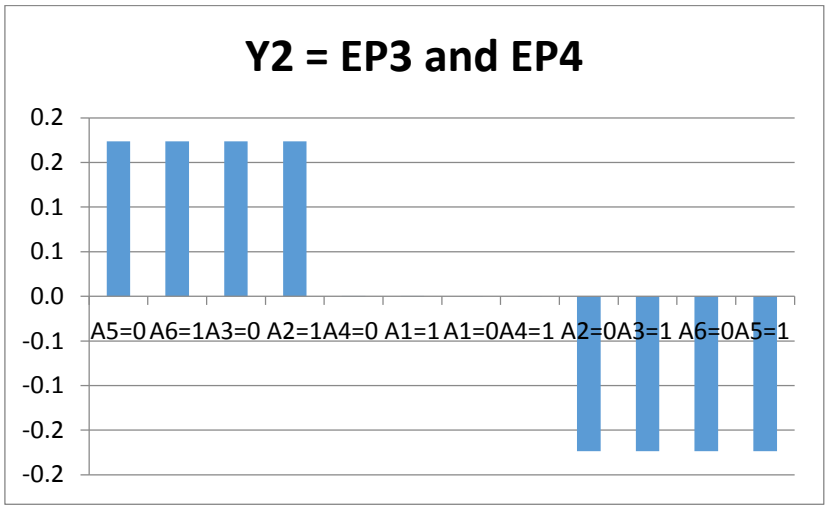
- Only consider the value with positive weight for Y1 and Y2. For example in Table 20, the value set S1 for Y1 is {A2=1, A3=1, A1=0, A4=0}. The value set S2 for Y2 is {A2=1, A6=1, A3=0, A5=0}.
- Similar to the action parameter selection, value selection is also required to selection a value set which try to cover all the most effective value of each action parameter for Y1 and Y2, as the positive value from different action parameters can be combined.

- Compromise the capacity of GPC extraction and the number of the selected value. Too much value will increase the time complexity of the GPC extraction algorithm.

After selection, the remaining action parameters and values are A1={0}, A2={1},A3={0,1}, A4={0}, A5={0}, A6={1}.



(a) Weight of values for Y1



(b) Weight of values for Y2

Figure 50: Bar chart of weights for values of AP

The forth step is to perform full factorial experiments for the reduced system. The parameters were extracted from one set of experiments. We made sure that the "simplified" real or simulated systems still provide the GTC and the GPC for which we need. In order to perform the experiments, it is required to provide a fixed value to {A7, A8, A9, A10}. The question is then what value should we give to them.

Regarding the fixing, we also have three situations:

- If the weights of one action parameter are only positive, we fix it at the largest value. As the decision function $D(x)=W^T\phi(x)+b$, if $(W)_I$ is positive, this means the action parameter Ai has a positive effect on satisfying the evaluation parameter. Thus, we should fix it at the largest value to maximize the positive effect. For example, the weight of A7 is 0.3 for Y1 and 0.8 for Y2; therefore, we fix the input level of A7 to 1.

- If the weights of one action parameter are only negative, we fix it at the smallest value. As the decision function $D(x)=W^T\phi(x)+b$, if $(W)_I$ is negative, it means the action parameter Ai has a negative effect on satisfying the evaluation parameter. Thus, we should fix it smallest value to minimize the negative effect. For example, the weight of A9 is -0.3 for Y1 and -0.1 for Y2; therefore, we fix the input level of A9 to 0.

- When one action parameter has both negative and positive weights, the fixed value choice is based on the following principle: positively satisfy evaluation parameters, and attempt to not seriously dissatisfy any evaluation parameters. For example, A8 has a weight of 0.5 for Y1, which requires A8 = 1. On the other hand, A8 has a weight –of 0.2 for Y2, which require A8 = 0. After balancing the satisfactory and dissatisfactory to Y1 and Y2, as 0.5 is more than 0.2, we fix A8=1.

The fixing result is {A7 = 1, A8 = 1, A9 = 0, A10 = 1}. The value selection results in Step 4 make A1 = 0, A2 = 1, A4 = 0, A5 = 0 and A6 = 1, which means that we retain only one value for these variables. Thus, in this case, the value of these variables is also fixed. So only variable A3 is not fixed and has more than one value (i.e., two values {1,0}). To summarize, applying the SVM algorithm to the parameters leads to fixing the value of A7, A8, A9, and A10. Applying the SVM algorithm to the remaining set of APs leads us to fix the values of A1, A2, A4 and A5. Thereby, the two remaining experiments in step 5 are illustrated in Table 21:

Table 21: Illustration of two remaining experiments

Fixed Action Parameters									AP	**Evaluation Parameters**			
A1	**A2**	**A4**	**A5**	**A6**	**A7**	**A8**	**A9**	**A10**	**A3**	**EP1**	**EP2**	**EP3**	**EP4**
0	1	0	0	1	1	1	0	1	1	1	1	0	0
0	1	0	0	1	1	1	0	1	0	0	0	1	1

As our purpose is to extract GPC for target GTC, the action parameter and experiments are simple enough to perform the GPC extraction. We also have to ensure that with the new reduced set of variables and experiments, we still get the GTC whose GPC are supposed to be extracted. Therefore, the fifth step is GTC extraction.

Seven GTCs were extracted, and our target GTC {E1 and E2 vs E3 and E4} is involved. Thus, we are satisfied with our simplified simulation model and more precisely, the model of the simulator remains the same, with only a few values that are vary while the solution space is smaller. As we have our targeted GTC in our experiment, we can use it for searching the GPCs, otherwise we would have to come back and choose more action parameters or values.

The last step is GPC extraction of the targeted GTC. In our example, the extraction result is very simple and it can be seen as classical physical contradiction with context in Figure 51.

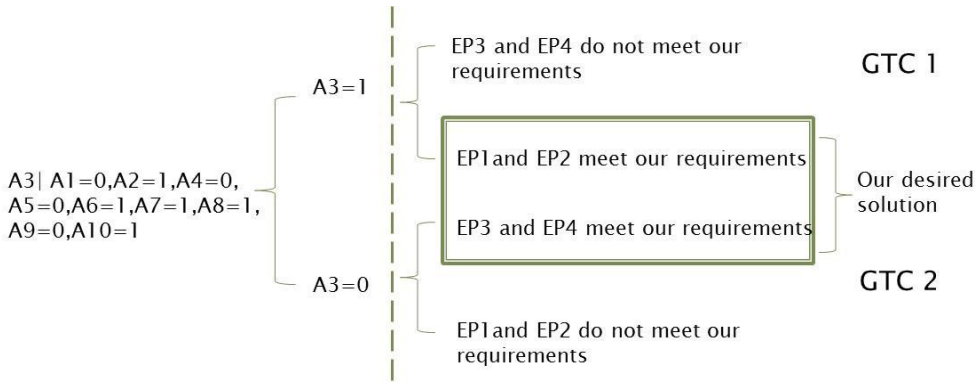


Figure 51: System of contradiction with context

What is the contribution of our method in contradiction formulation in this case? There are only two experiments in the formulation of contradiction. Can we just select two experiments respectively from E1 and E2 in Appendix 1 to form the same contradiction? The answer to this question is no. We have checked E1 and E2, and they only cover one of the two experiments. Thus this contradiction could not be obtained from the initial set of experiments. If we had another random set of experiments, it may even happen that none of these two experiments belong to the sample of experiments set. However with the proposed SVM methodology we would have found it.

The second contribution of this approach concerns the noise provided by the variables and values that were removed by this methodology. Thus, the methodology filters provides unusable physical contradictions. Let us explain. If we randomly select

two experiments from E1 and E2 to form contradictions, the conflicts among action parameters are several. Such as in Table 22, where six action parameters (i.e., A3, A6, A7, A8, A9, A10) have conflicts and could be candidates for classical physical contradiction as defined by the context of A1 = 0 and A2 = 1 and A4 = 0 and A5 = 0. Experts looking to these two experiments may not be able to choose among them. According to the chosen GTC, among the six conflicts, action parameters A7, A8, A9, and A10 are noise action parameters. These parameters should not belong to the concept of AP for explaining the GTC. A deeper illustration of this point requires more information regarding the problem context and studied system description.

Table 22: Randomly selection two experiments from E1 and E2

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	EP1	EP2	EP3	EP4
0	1	**0**	0	0	**1**	**1**	**0**	**1**	**1**	-1	-1	1	1
0	1	**1**	0	0	**0**	**0**	**1**	**0**	**0**	1	1	-1	-1

Thereby, our method indeed helps in removing not only noise action parameters, but may also help in filtering simple physical contradictions that could not be provided by experts.

4-6.2 EP based extraction

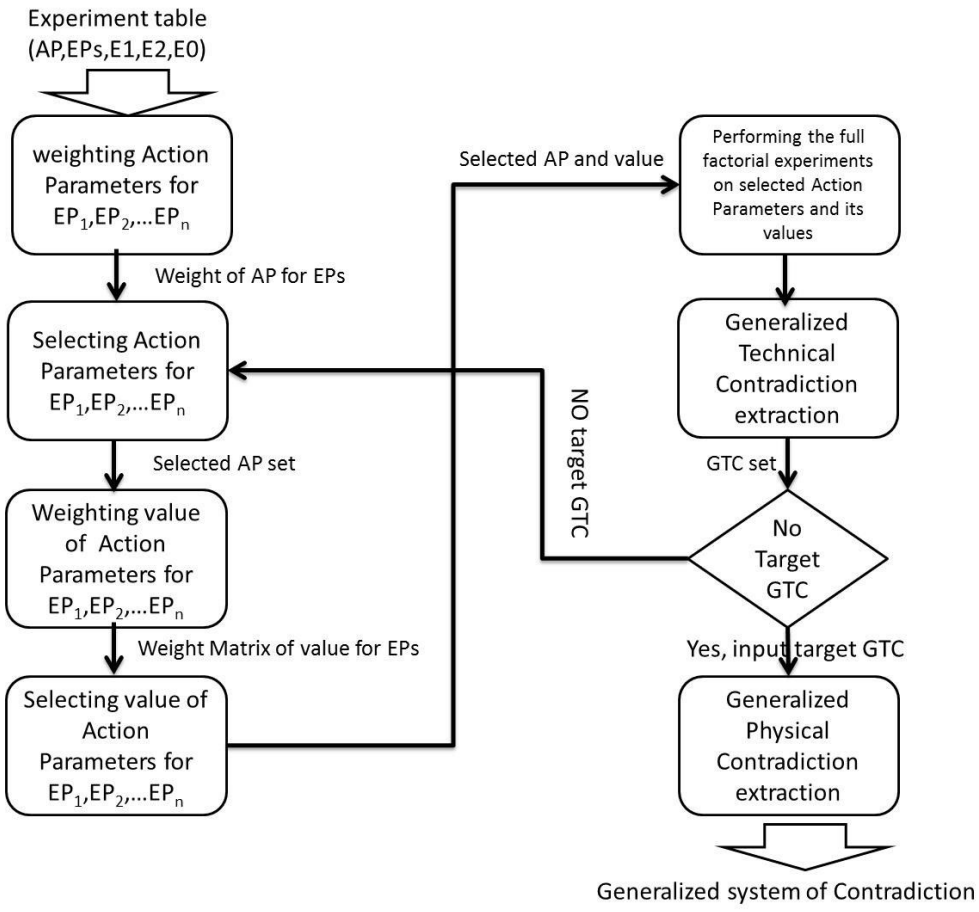


Figure 52: Process of EP-based extraction

The second strategy is the evaluation parameter based extraction. The difference from the previous strategy is that it is performed without considering the target GTC. In fact we only consider the experiments and the binary values of the EPs. The entire process of this strategy is similar to the GTC-based extraction and is presented in Figure 52.

For this process, we do the same thing to each EP as we did to Y1 and Y2 in the previous section. Using the same example in Appendix 1, the weighting results of the action parameter for each evaluation parameter are presented in Table 23.

Table 23: Weighting results of action parameter for example

Weight	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	3 fold Cross-validation
EP1	-6,5	6,6	6,5	0,1	-0,3	-0,2	0,1	-0,3	-0,4	0,3	100%
EP2	0,0	6,8	6,9	-6,5	-0,2	-0,1	0,3	-0,2	-0,1	0,5	100%
EP3	-0,1	7,3	-6,9	0,0	-7,7	0,0	0,8	0,4	0,4	0,0	100%
EP4	-0,3	0,1	-7,4	0,1	-7,8	7,5	0,7	-0,3	-0,5	0,2	100%

The selection set is the same set {A1, A2, A3, A4, A5, A6}. Then, we evaluate the values of {A1, A2, A3, A4, A5, A6} with the weights presented in Table 24.

Table 24: Weight matrix for value selection to EP of the example

weight	Action Parameter												3- folderCross-validation
	A1=1	A2=1	A3=1	A4=1	A5=1	A6=1	A1=0	A2=0	A3=0	A4=0	A5=0	A6=0	
EP1	-3	3	3	0	0	0	3	-3	-3	0	0	0	100%
EP2	0	3	3	-3	0	0	0	-3	-3	3	0	0	100%
EP3	0	3	-3	0	-3	0	0	-3	3	0	3	0	100%
EP4	0	0	-3	0	-3	3	0	0	3	0	3	-3	100%

The selection principle is the same as previously where the value selection results are {A2 = 1, A3 = 1, A1 = 0} for E1, {A2 = 1, A3 = 1, A4 = 0} for E2, {A2 = 1, A3 = 0, A5 = 0} for E3, {A5 = 0, A3 = 0, A6 =1} for E4. The integrated result is A1= {0}, A2 = {1}, A3 = {1,0}, A4 = {0}, A5 = {0}, and A6 = {1}, the same as for the GTC based extraction. The following steps are also same as that for GTC selection. In this example, the approach provides the same results as we do not see the differences between them. However, they actually have their own advantages and disadvantages.

4-6.3 Comparing the strategies

No matter the GTC-based extraction or EP-based extraction, they both are aiming at extracting the GPC for the target GTC. The advantage of the GTC-based approach is the more targeted selection of the action parameters, which will limit the selection to the action parameters that only have effects on two sides of the targeted GTC. The disadvantage of GTC-based approach is the limit of the number of positive instances in the SVM training. This is due to E1 or E2 having an insufficient number of experiments that would affect the training results of the SVM model. The advantage of EP-based extraction though, is the consideration of all of the evaluation parameters during the stage of action parameter selection. We should find more GTC with the second approach; thus, we should not need to repeat the simplification process for each GTC. More practical experiments are required to check this assertion and make the comparison.

Additionally, it will better shape the Pareto set of the “reduced” system. In actuality, the second strategy can be considered a strategy for searching the main action parameters that influence the Pareto set of the evaluation parameters or at least to remove those that are not influencing.

4-7 Conclusive remarks

The previous work has documented the origin of generalized technical contradiction and generalized system of contradiction as well as their extraction algorithm regarding the results of experiments or simulations. However, each algorithm is based on exhaustive search, and as such, they are not usable when there are too many parameters. Especially, the previous GPC extraction algorithm that limits the binary action parameters to twelve. In this chapter, we introduced feature selection algorithm for one of most important algorithms of SVM, by which all the features are weighted by contribution. Based on the weight provided by the SVM approach, we proposed to make a selection of action parameters and their values as to simplify the system that reduces the computational complexity of the GPC exhaustive searching algorithm. Based on this tool, which can handle several thousands of APs, two strategies were proposed in this chapter: GTC-based GPC-extraction and evaluation

parameters based GPC-extraction. For GTC-based extraction, the selection of action parameters and values were orientated towards the two-side of generalized technical contradiction (Y1 and Y2); whereas the evaluation parameter based extraction orientated toward each evaluation parameter.

In order to better interpret our methodology, a double Kanban case study will be illustrated in the next chapter, the entire process will be introduced based on the GTC extraction, GPC extraction and action parameters selection. For this case, solution from TRIZ will be proposed, and compared with experts' solution and original system. We can see the Pareto of the evolved system from TRIZ solution cover the Pareto of the original system and experts proposed system in the next chapter.

The parameter selection method proposed in this chapter has a contribution to answer Q5, how can we extract the relevant contradictions without exhaustive research, because the selection process filters many unimportant action parameters so that many states from these action parameters have been removed, like the example in 4-6.1, we fixed action parameter {A7=1,A8=1,A9=0,A10=1}, that means we actually remove the states in which { A7=0,A8=0,A9=1,A10=0}.

Chapter 5 Process of inventive problem solving

This chapter first presents a general framework of the inventive design problem solving process, which integrates simulation, optimization, and TRIZ based methods. Second, a case study discussing a double Kanban system illustrates how the algorithms presented in chapters 2, 3 and 4 can be used within the previous general framework.

5-1 Entire process

As mentioned in the previous chapter, we used the dialectical approach from the TRIZ theory to solve inventive design problems that cannot be solved solely by optimization methods. Practical evidence has shown that sometimes it may be less expensive to use an inventive problem solving method even if the problem can be solved by optimization. However, in many situations, both approaches are required to provide satisfactory solutions and are used in sequence. A theoretical or general framework for design problem solving process based on the simulation-optimization-invention loops when simulation or experimental means are available is proposed and shown on Figure 53. The solving process involves seven functions as represented by boxes in Figure 53. The functions can be performed independently by using different methods and tools such as design of problem models (quality function deployment, design of experience …); simulator models (CAD, Witness); simulation algorithms (stochastic optimization, genetic algorithms); multi-criteria decision analysis; and changing model methods such as TRIZ, etc. The first function represents the problem formulation and definition through the requirements and the dissatisfaction of the customer. The evaluation parameters (EPi) are used to describe the objectives and are measured to check whether the customer's requirements are satisfied. The action parameters (APi) with their possible values represent the system variables on which one can act. Some relations (Ri) between system variables and parameters are described through the system constraints.

5-1 Entire process

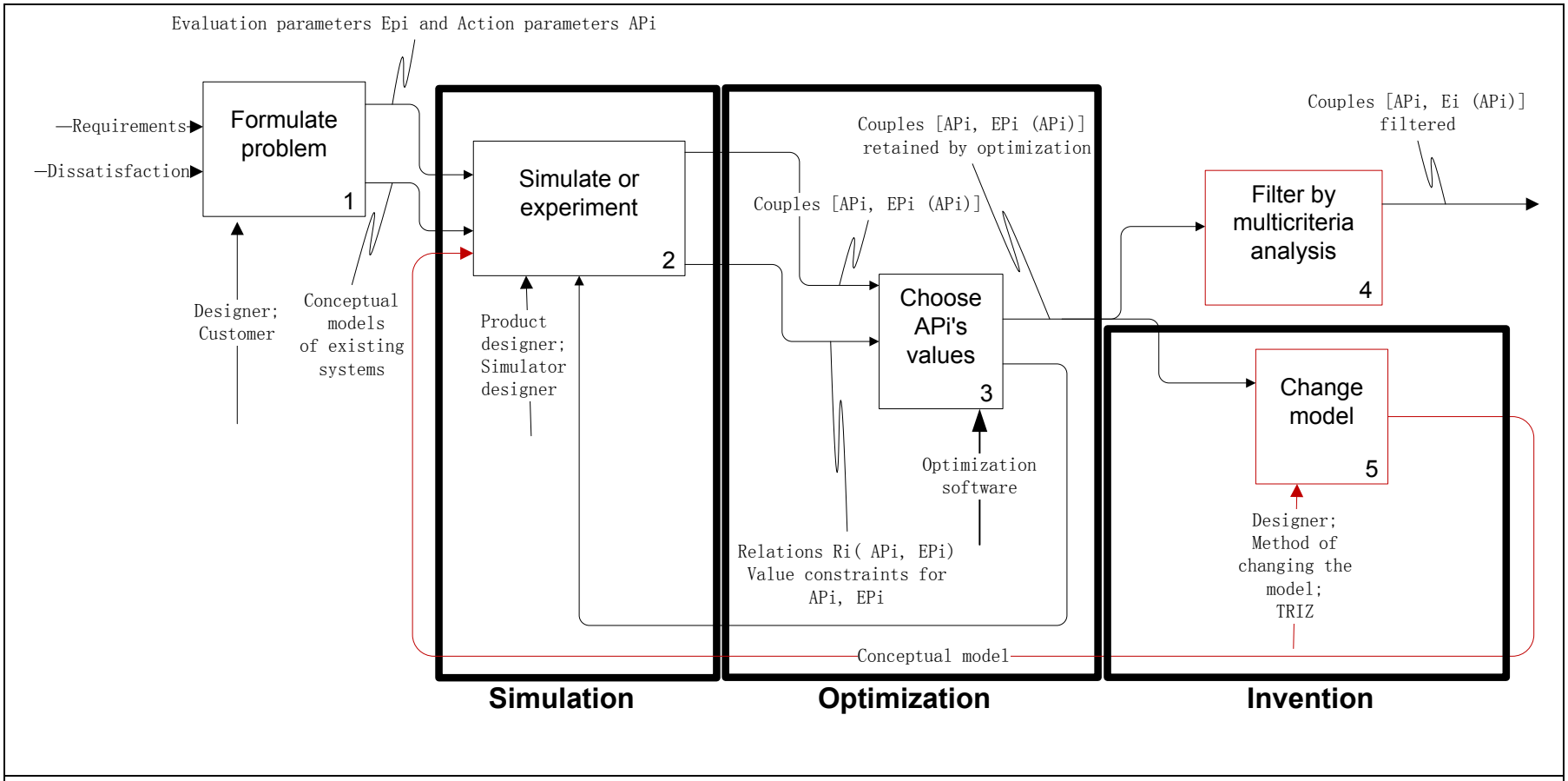


Figure 53: General framework for inventive design problem solving

The second function generates the experiments with valuation of possible solutions, which are obtained by the aid of a simulator or by physical experimentation. Sometimes a large number of action parameters and their values are accessible making it is impossible to process all data. Thus, the third function should choose the relevant action parameters for the valuated experiments with possible solutions (i.e., design of experiment or optimization algorithms). When satisfactory couples (APi, EPi(APi)) are obtained (i.e., their evaluation parameters achieve the expected values) or when the time allowed for experiment is over, the results are filtered for example by multi-criteria analysis, which is the fourth function. When the evaluation parameters do not achieve the expected values after the time allowed for the experiment or because of proven limitations of the system as pointed out by the experiments; it is necessary to change the conceptual model to reach the requirements, which is the fifth function. By following this process, the use of dialectical inventive solving approach is proposed. Then the loop starts again with a new model by performing Step 2.

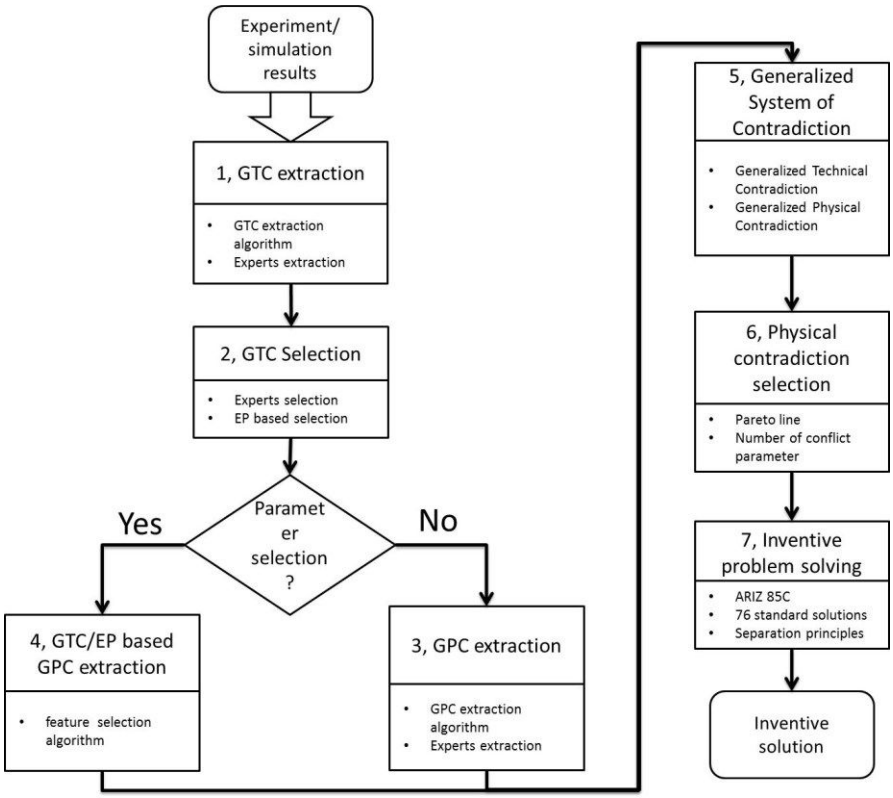


Figure 54: Process of model change

Our contribution is focusing on the function 5 model change, in Figure 54, a process of function 5 is proposed, where inputs experiments or simulation results.

In this process, we perform the GTC extraction as first step. The GTC can be extracted by experts or by the GTC extraction algorithm discussed in Chapter 2. For Step 2, the target GTC can be selected by experts, or we select GTC based on the number of evaluation parameters. If we completely solve this contradiction with more evaluation parameters involved then there will a higher level of satisfaction based on the achievement of evaluation parameters. Once the target GTC is determined, we need to select the GPC. If the time complexity is suitable for GPC extraction, we can directly extract the GPC or let the experts extract the GPC. Otherwise, we perform the GTC/EP-based GPC extraction as proposed in Chapter 4. When we obtain the GTC and corresponding GPC, the generalized system of contradiction is formed and the classical TRIZ physical contradiction can be obtained by comparing the two concepts of the GPC. Among these physical contradictions, we propose to select a physical contradiction in which only one action parameter has conflict. Additionally, the context that develops the conflicting state located within the Pareto line should be selected for the physical contradiction. The last step is to solve this physical contradiction by TRIZ methodology.

5-2 Case study of double Kanban

In this section, the study case is presented to illustrate the use of the proposed model change process with the tools introduced in previous chapters.

5-2.1 Problem formulation and simulation

This case involves a simple manufacturing system managed by Kanban cards simulated with the aid of the flow simulation software Witness 14 and the simulation layout is shown in Figure 55.

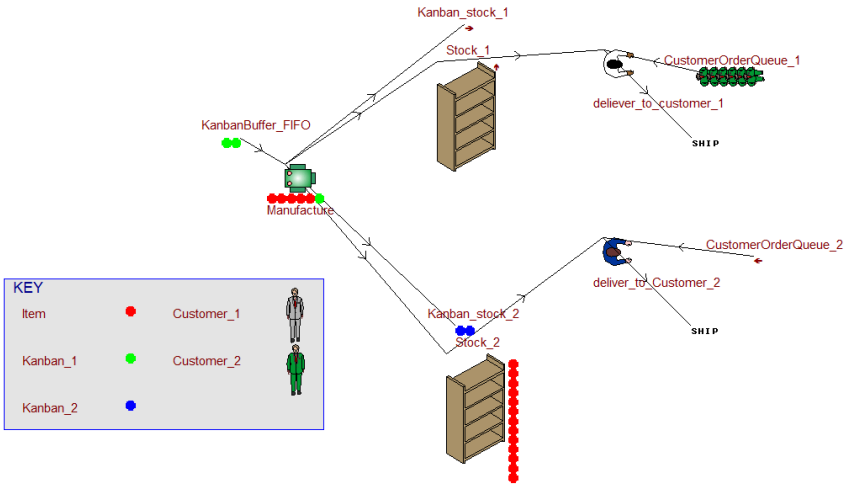


Figure 55: Double Kanban system

The system is composed of one manufacturing machine producing two different items stored in two independent inventories and delivered to two kinds of customers, customer_1 and customer_2. The interval time of customers' arrival time follows a uniform random distribution i.e. interval of customer_1's arrival time follows a uniform distribution (0.18, 0.22), and interval of customer_2's arrival time follows a uniform distribution (0.9; 1.1). It is a simple pull system where Kanban cards are sent back to the manufacture machine when the container is empty. The Kanban card provides the quantity to be put into a container. The manufacture machine has a two different set up times for each item (i.e., 1 time unit for the first item and 2 time units for the second item), the operation time is 0.1 time unit for both items and the transportation times from manufacturing machine to the inventories is neglected. The transportation Kanban Size from the machine to the customers' inventories is one Kanban. The scheduling rule for the machine is FIFO (first in first out) according to the arriving Kanban cards. Thus a setup is performed any time the type of item (Kanban) changes in the Kanban waiting line. The total running time of one simulation is 1000 time units (1 time unit corresponds to 1 hour), with 10 time units required for warm up time. During the warm up time, all the Kanban cards come into the model and go to the manufacture machine, thus the initial stock of two inventories is from the manufacture of warmup time.

This study is modeled as follows: the Kanban size (quantity of items corresponding to one Kanban card) and the number of Kanban cards are the action parameters of the

system on which we can act. To evaluate the performance of this system, the average stock in the inventories and the average waiting time for customers for both items are defined as the evaluation parameters. In summary, four action parameters are defined: number of Kanban for item 1 (NK1), number of Kanban for item 2 (NK2), Kanban size for item 1 (KS1), and Kanban size for item 2 (KS2). In addition, the four evaluation parameters are average stock in stock 1 (AS1), average stock in stock 2 (AS2), service breakdown for item 1 (SB1) corresponding to the average lateness of the delivery to customer 1 and service breakdown for item 2 (SB2). The definition of the range of action parameters for optimization should be defined by engineer first, then deduced if there is a good solution out of the defined range through a statistical analysis of generated solutions. For our case, the range is defined from 1 to 10 for each action parameter.

5-2.2 Optimization and solution filter

Once the action parameters (NK1, NK2, KS1, KS2) and evaluation parameters (AS1, SB1, AS2, SB2) are determined, the experiments are performed. The Witness provides tools for random experiments. We used a random experiment as an optimization and 1039 experiments were performed. For a convenient view, we used AS1+AS2 and SB1+SB2 as two axes with the results shown in Figure 56.

In our problem, the goal is to have a situation with no lateness and no stock, which means that the value of each evaluation parameter is ideally targeted to be zero. However, this goal is impossible to be achieved by the system described above. The best values that we did obtain separately by optimization means for each evaluation parameters of this system was SB1 = 0, AS1 = 0, SB2 = 0, and AS2 = 0. No experiments provided a solution where all the evaluation parameters were simultaneously within the interval [0; 0.1]. This solution could not be reached by the above described system. Thus, our objective was then to modify the model of the existing system as little as possible such that a solution where each evaluation parameter stands within the range [0, 0.1] existed. Such a solution should outperform any solution of the actual system.

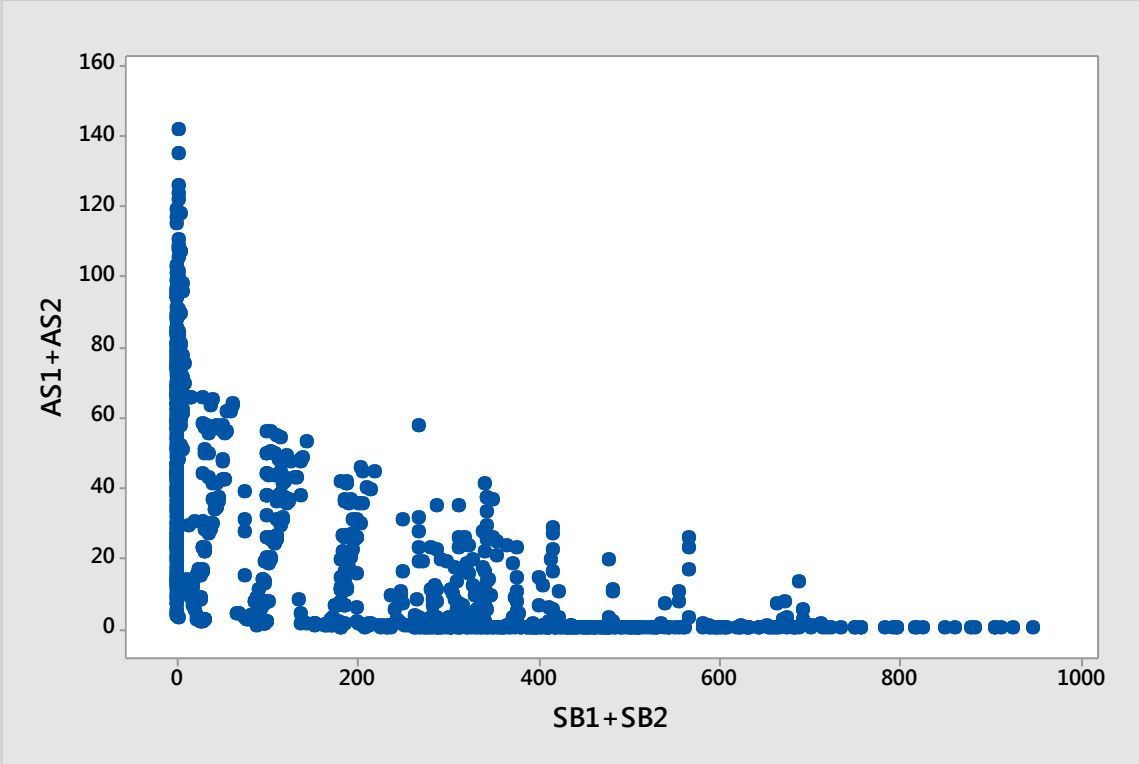


Figure 56: 1039 random experiments results

5-2.3 Model change

5-2.3.1 GTC extraction and selection

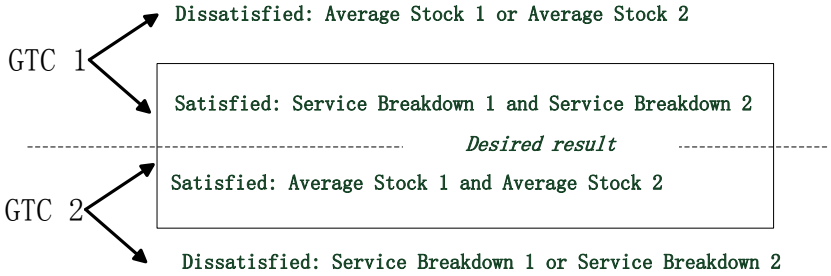
The results of experiments are transformed into a matrix similar to Table 25.

Table 25: Binarization principle of the evaluation parameters

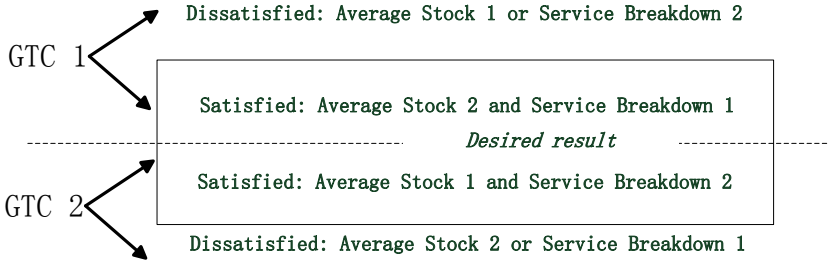
Action parameters				Evaluation parameters experimental values				Evaluation parameters binarized from objectives point of view			
KS1	**KS2**	**NK1**	**NK2**	**SB1**	**AS1**	**SB2**	**AS2**	**SB1**	**AS1**	**SB2**	**AS2**
2	8	1	1	1,67	0,05	0,11	2,76	0	**1**	0	0
7	9	1	1	1,50	0,26	0,13	3,07	0	0	0	0

If the value of the evaluation parameter drops in the interval [0; 0.1], it is satisfying and the value of the evaluation parameters for this experiment is set to '1', otherwise it is unsatisfying and denoted '0'. For example, in the first situation, KS1= 2, KS2=8, NK1=1 and NK2=1 as indicated in , as determined, SB1=1.67, which is greater than 0.1; therefore, it is denoted as '0', and as AS1 = 0.05, it is within the interval [0; 0.1],

hence it is binarized as '1', as are SB2 and AS2. The GTC extraction algorithm proposed in Chapter 2 was performed. As a result, twelve pairs of GTCs were found where only two of them included all the evaluation parameters as shown in Figure 57. That indicates if any of the two GTC were overcome, all the objectives would be reached.



a) GTC: SB1 and SB2 VS AS1 and AS2



b) GTC: SB1 and AS2 VS AS1 and SB2

Figure 57: GTCs involving all the evaluation parameters

From a problem solving point of view, we should choose one of GTC in Figure 57, because if we solve one of them completely, we can obtain our final ideal solution. From a GPC extraction point of view, the Chapter 3 indicates us to choose the GTC which has a large number of experiments in E1 and E2. Thus, in the next step we choose the GTC in Figure 57a) as the target GTC for searching for a solution. As the experiments are not full factorial, and performing full factorial experiments and GPC extraction algorithm cost a large amount of time, we use parameter selection to reduce the number of action parameters and values.

5-2.3.2 EP based GPC extraction

In this part, we performed the EP-based GPC extraction strategy discussed in chapter 4.

5-2.3.2.1 Weighting and selecting action parameter

As Chapter 4 described, we used the SVM model to weight the action parameters, and result are shown in Table 26.

Table 26: SVM weights

Weight		Evaluation Parameter				Key
		SB 1	AS 1	SB 2	AS 2	
Action Parameter	KS1	59	-89	-21	25	KS1 is the size of Kanban 1
	KS2	42	-78	108	-146	KS2 is the size of Kanban 2
	NK1	84	-104	-51	48	NK1 is the number of Kanban 1
	NK2	-43	7	155	-57	NK2 is the number of Kanban 2
3-fold Cross- validation		97.88%	98.36%	96.53%	98.07%	

In Table 26, the intuitive relation between the action parameters and the evaluation parameters is that the KS1 and the NK1 have a positive effect on the SB1, and a negative effect on the AS1. Moreover, as item 1 competes with item 2 for the limited capacity of the manufacture machine, they have reverse effects on SB2 and AS2. However, for the different KS2 and the NK2 values, the KS2 has a positive effect on SB1 and SB2. The reason is existence of the part time charge. The larger the KS2 is, the less the part time charge. The large KS2 can lead to a large quantity of item 2 in the buffer Stock_ 2, thereby increasing the interval between times of producing the item 2. Not surprisingly, the long interval for producing the item 2 indirectly decreased the total setup time, and increased the capacity of machine for producing item 1. This enhanced the service quality but increased items for average stock. From the weight matrix, we can see that all the action parameters are important. Therefore, we kept all the action parameters {KS1, KS2, NK1, NK2}.

5-2.3.2.2 Weighting and selecting the values of action parameters

In this section, the same method was performed as selecting the value of action parameter by the SVM-Weight. First, coding each value of action parameter (40 binary parameters in this case), then creating the SVM-model with more than 95% cross validation-accuracy. Finally, presenting the weight matrix for each action parameter (in Table 27,Table 28,Table 29 and Table 30). We selected the value by using the principle presented in Chapter 4. For example, in Table 27, the selection result was 1, 2, 8 and 10. Among the four values, 1 and 2 have 1st and 2nd positive

influence on AS1 and SB2, whereas, 8 and 10 have 1st and 2nd positive influence on AS2 and SB1. The selection results are shown in Table 31.

Table 27: Weight matrix for KS1 (selected 1,2,8,10)

SVM Weight		Value of action parameter										
		KS1=1	KS1=2	KS1=3	KS1=4	KS1=5	KS1=6	KS1=7	KS1=8	KS1=9	KS1=10	SVM Accuracy(%)
evaluation parameter	AS1	17.7	11.2	5.1	0.7	0.3	-4.4	-5.8	-8.1	-8.5	-8.2	95.6
	SB1	-16	-11.7	-3.4	-1.1	3.9	4.4	5.5	6.2	5.7	6.4	97.8
	AS2	-5.9	-0.5	0	0.6	-0.1	0.1	1.1	1.7	1.2	1.9	98.1
	SB2	4.5	2.7	1.7	-0.3	0.3	-1.2	-0.8	-0.8	-2.1	-4	95.5
		KS1 is the size of Kanban 1										

Table 28: Weight matrix for KS2 (selected 1,2,8,9)

SVM Weight		Value of action parameter										
		KS2=1	KS2=2	KS2=3	KS2=4	KS2=5	KS2=6	KS2=7	KS2=8	KS2=9	KS2=10	SVM Accuracy(%)
evaluation Parameter	AS1	12.4	11.4	10.6	9.1	6.4	4.2	-2	-16.9	-15.3	-20.1	95.6
	SB1	-6.5	-6.4	-6.2	-7.4	-6	-6.5	-6.5	15.9	15.1	14.6	97.8
	AS2	26.9	11.3	6.4	3.8	-2.1	-6.8	-9.5	-10.1	-10.5	-9.4	98.1
	SB2	-20.8	-16.4	-13.1	-2.9	2.4	4.8	8	13.6	11.8	12.7	95.5
		KS2 is the size of Kanban 2										

Table 29: Weight matrix for NK1 (selected 1,2,7,10)

SVM Weight		Value of action parameter										
		NK1=1	NK1=2	NK1=3	NK1=4	NK1=5	NK1=6	NK1=7	NK1=8	NK1=9	NK1=10	SVM Accuracy(%)
evaluation Parameter	AS1	19.8	14.2	7.2	-0.2	-2.8	-3.9	-6	-8.6	-9	-10.9	95.6
	SB1	-15.4	-15.4	-2.7	-1.2	1.4	4.4	6.7	7.3	6.9	7.9	97.8
	AS2	-14	-2.6	0.7	2.2	2.3	1.4	3.5	2.5	1.8	2.2	98.1
	SB2	14	8.9	5.5	3	-2.4	-3.5	-4.5	-6.3	-6.5	-8.1	95.5
		NK1 is the number of Kanban 1										

Table 30: Weight matrix for NK2 (selected 1,3,8,10)

SVM Weight		Value of action parameter										
		NK2=1	NK2=2	NK2=3	NK2=4	NK2=5	NK2=6	NK2=7	NK2=8	NK2=9	NK2=10	SVM Accuracy(%)
evaluation Parameter	AS1	1.2	-0.1	2.8	-1.7	-1.2	-0.6	-1.2	0.1	0.2	0.4	95.6
	SB1	7.5	-0.2	-0.9	-1.7	-0.5	-0.5	-1	-1.6	-0.5	-0.6	97.8
	AS2	15.3	8.8	7	0.3	-1.5	-6.4	-6	-4.6	-5.5	-7.4	98.1
	SB2	-30.1	-6.8	-2.1	0.9	3.5	3.6	6	8	7.8	9.2	95.5
		NK2 is the number of Kanban 2										

Table 31: Value selection by SVM weights

	KS 1	KS 2	KB 1	KB 2
Selection result	1, 2, 8, 10	1, 2, 8, 9	1, 2, 7,10	1, 3, 8, 10

5-2.3.2.3 Performing full factorial experiments

Based on selected value in Table 31, the 256 simulations were performed on the reduced system with the results displayed in Figure 58. Observing the distribution of results in Figure 58, the obvious conclusion is that the 256 results trend forms the Pareto line. In contrast to a random solution of the original system, most of results in the reduced system are located on or close to the Pareto front, they not only focus

on the solution toward to original point, but also the extreme points on the two sides of the Pareto line, which allows for better formulation of the technical contradiction.

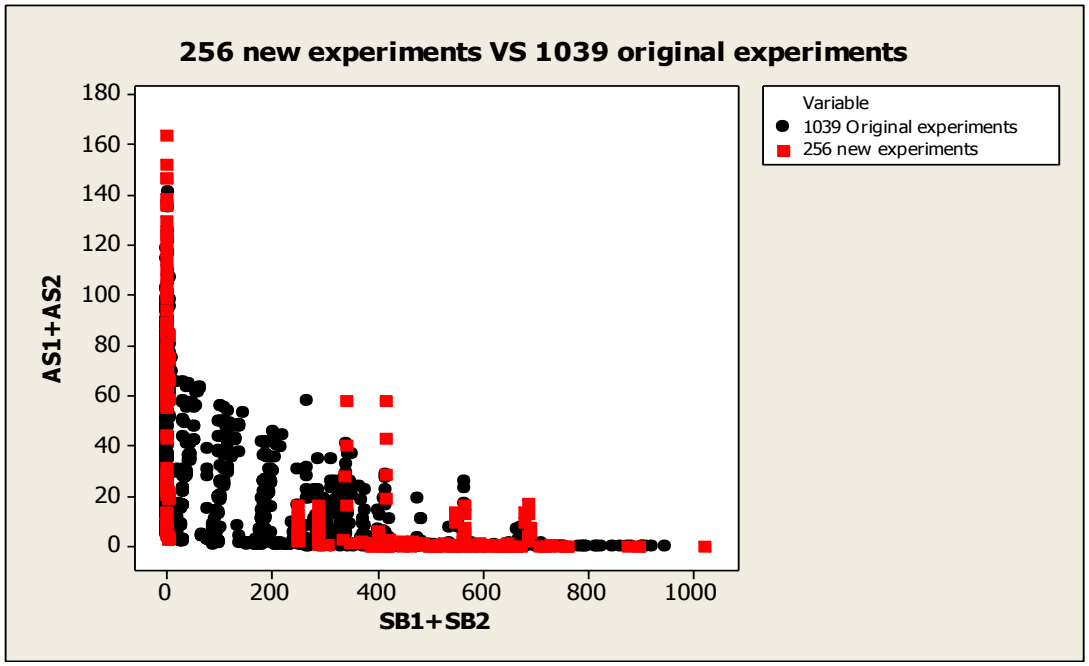


Figure 58: 256 full factorial experiments

In terms of the sum of the four evaluation parameters, it was discovered the solution was close to the ideal point than random solution as well. In Table 32, 3 better solutions (4,516) were discovered in term of the sum of the four evaluation parameters than the random solution of original system (4,582).

Table 32: Best values from reduced system

AP				EP				
KS1	KS2	NK1	NK2	SB1	AS1	SB2	AS2	Sum of EP
2	8	2	1	**1.323**	0.323	0.107	2.762	**4.516**
8	8	1	1	1.536	0.234	0.149	**2.609**	**4.528**
1	8	2	1	1.505	0.146	**0.104**	2.784	**4.538**
2	8	1	1	1.667	**0.046**	0.107	2.762	4.582

5-2.3.2.4 GTC extraction and GPC extraction

The GTC extraction algorithm was performed on the 256 full factorial experiments. Twelve GTCs were extracted and they are the same as the 12 GTCs of original system. That would indicate both of them are acceptable models in regard to GTC extraction. Based on the results of the GTC extraction on the reduced system, the GPC extraction was performed for the target GTC in Figure 57a. Once we had the GPC, the GSC

model was formulated as an initial point of the TRIZ analysis. As the reduced system had the same GTCs as the original system, and its experiments are full factorial experiments; the GPC could be extracted towards the target GTC. As a result, 68 states in concept held GTC1, and 30 states held GTC2. Based on the above calculation, the generalized system of contradiction was displayed in Figure 59. In the next section, we will select a physical contradiction with context from generalized system of contradiction, and attempt to solve it.

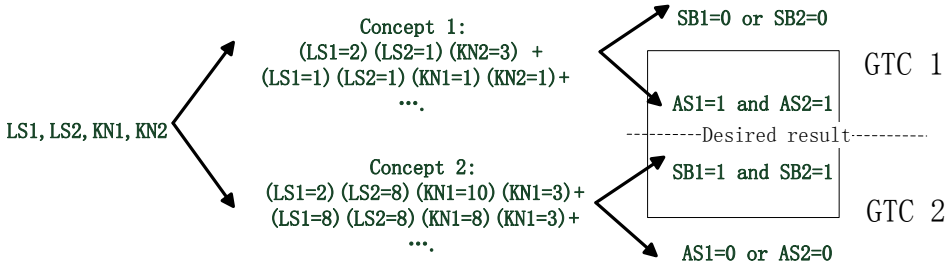


Figure 59: Generalized System of Contradiction

5-2.3.3 Extract physical contradiction with context from two concepts

Through comparison of two concepts, 2040 physical contradictions with context were obtained. The selected physical contradiction should have one conflict action parameter and there were 16 contradictions with one conflict action parameter. However, the conflict action parameter was just one action parameter KS2 with different values (Kanban Size of Kanban 2); thus, KS2 was selected as the physical contradiction. We choose the context that ensures two conflict states around the Pareto line. Therefore, we choose the contradiction that state KS1 = 2, KS2 = 1, KN1 = 10, and KN2 = 3 versus state KS1 = 2, KS2 = 8, KN1 = 10 and KN2 = 3 shown in Figure 60. This conflict provides us a physical contradiction KS2=1 vs KS2=8 with context that KS1=2, KN1=10 and KN2=3.

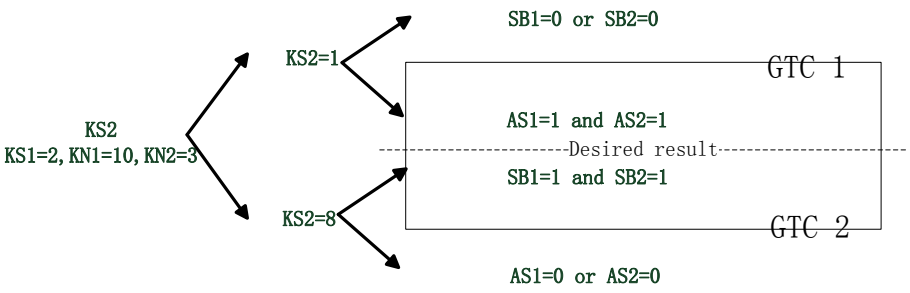


Figure 60: Physical contradiction with context

5-2.3.4 Inventive problem solving by using TRIZ

In this section, we formulated the TRIZ model for solution then list the TRIZ solutions. First, we formulated the two GTCs shown in Figure 61. Because of the priority of the service breakdown, GTC 2 in Figure 61b was chosen as our initial technical contradiction. In GTC 2, Concept 2 made the service breakdown satisfied (SB1 = 0 and SB2 = 0). However, it had a harmful effect on the two stocks.

a. GTC 1 b. GTC 2

Figure 61: Two sides of target GTC

ARIZ85C[1], [73] was performed and some details are described below:

- Operation zone: stock and machine
- Operation time: when producing the items for Kanban 2 and when Kanban 2 in the Kanban stock
- Ideal final result: The X-element, without complication of the system and without harmful side effects, eliminates <Average stock> during the < when produce the items for Kanban and when Kanban in the Kanban stock > inside the <machine and stock> and keeps the tool's ability to provide <items to customer requirements in time>.

Several partial solutions were obtained, which are summarized below:

- Solution indicated by 76 inventive standards (3.1.1, Formation of bi or poly-systems): Indicated to use two machines to produce items. Although this solution can technically solve our problem, bringing another machine is a large change that does not follows the principle of minimal change in TRIZ or that could be considered as too expensive. Thus it could only be an alternative solution in case of no satisfying solution.
- Partial solution from ideal final result 1 (IFR1): Kanban for item 2, without complication of system and without harmful side effects, eliminates <average stock> during the < time when producing the items for Kanban 2 or when Kanban 2 is in the Kanban stock > inside the <machine and stock> and keeps the ability to provide <items to customer requirements in time>. On the analysis of this partial solution, if we want to remove the average stock by Kanban during <time when producing the items for Kanban 2 and when Kanban 2 is in the Kanban stock>, we should decrease the Kanban size when producing items for Kanban 2 or reducing the items when Kanban is at customer stock.
- Ideal final result 2 (IFR2): The machine has to <provide item> when there is <customer requirement>, and machine has to <provide no item itself> <when there is no customer requirements>.

Synthesizing of partial solutions:

Through analysis of IFR2 the hidden reason why the machine cannot directly manufacture the item when the customer is coming, is that setup times and cycle time prevent machine from provide item immediately when the customer comes. The solution of double machine is just an approach for removing of the setup time (except initial one).

The solution synthesized from partial solution Kanban_2: if we keep context: KS1=2, KN1=10, KS2=8, and define KN2=3, the result is shown in Table 33. we also can find that, after 1000 time units running, the minimum stock in stock_2 is 14, which means there are 14 items in the stock all the 1000 time units. Thus, we can make KN2=2 to remove 8 items, the minimum stock in stock_2 becomes to 6. The following problem

is how to use Kanban_2 to remove the minimum stock. If we make KN2=1, although the minimum stock becomes 0, and SB2 increases.

Table 33: The result of state (KS1=2)(KS2=8)(KN1=10)(KN2=3)

KS1	KS2	KN1	KN2	AS1	AS2	SB1	SB2
2	8	10	3	9,71	18,65	0	0

The strategy for addressing this problem is to keep the supplement as two Kanban_2, but keep the stock as one Kanban_2 and 2 items as shown in Figure 62.

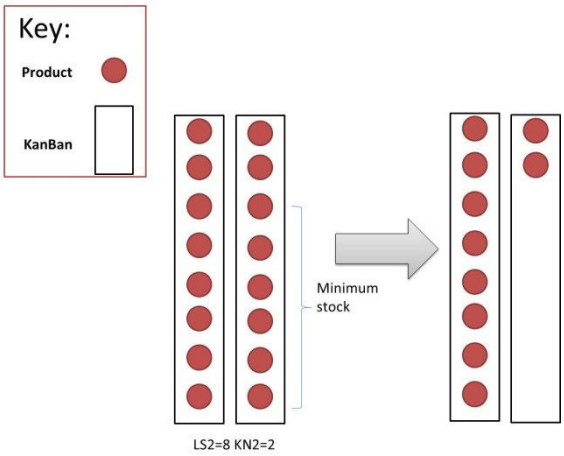


Figure 62: Illustration of supplying strategy

To realize the solution and keep the context, we made 10 items for stock_2 during the warming up time, and defined Kanban_2=1. In order to keep the supplementary rhythm, which is to send Kanban to machine after delivering eight items, we let the Kanban go to the machine after delivering each eight items, and keep two items in the stock during the supplementary time of producing for Kanban_2. Thus, the mechanism is to make Kanban Size_2 = 8, and there is security stock = 2, when the Kanban_2 with items return to stock_2. Kanban_2 must supplement the security stock if its items less than 2. By this approach, the simulation result was SQ1 = 0, SQ2 = 0, AS1 = 9.71, and AS2 = 4.65, whereas the best simulation solution of original model that made SQ1 = 0 and SQ2 = 0 had a result of AS1 = 11.7 and AS2 = 10.65.

5-2.3.5 System evolution from partial solution

Inspired by the partial solution of Kanban in previous section, the simulation model evolved by adding two parameters; Security_Stock_1 and Security_Stock_2. The

purpose of security stock (Security_Stock) is to keep delivering items to the customer even when all the Kanban are sent to the machine. However, when the items of Kanban arrive in the stock, it must fulfill the security stock first.

In contrast to original simulation model, an obvious conclusion that can be drawn is that if security stock is equal to 0, the evolved model is equal to the original model. That means that Pareto front of original model must be dominated by the evolved model. In order to show the optimal Pareto front of original model, we define the range of four action parameters from 1 to 15 and perform all the combination of experiments (50625 combinations). In regard to the experiments of evolved model, the simulated annealing was performed on the minimization problem SB1+SB2+AS1+AS2. In Figure 63, we only removed the results in region [0, 1]×[0, 17]. However, the Pareto known from simulated annealing was covered and went beyond the optimal Pareto front of the original model. Therefore, the improvement on this model pushes the system evolution, and overcame the Pareto front of the initial model. Unfortunately there was no solution to arrive at our final goal as each EP was below 0.1.

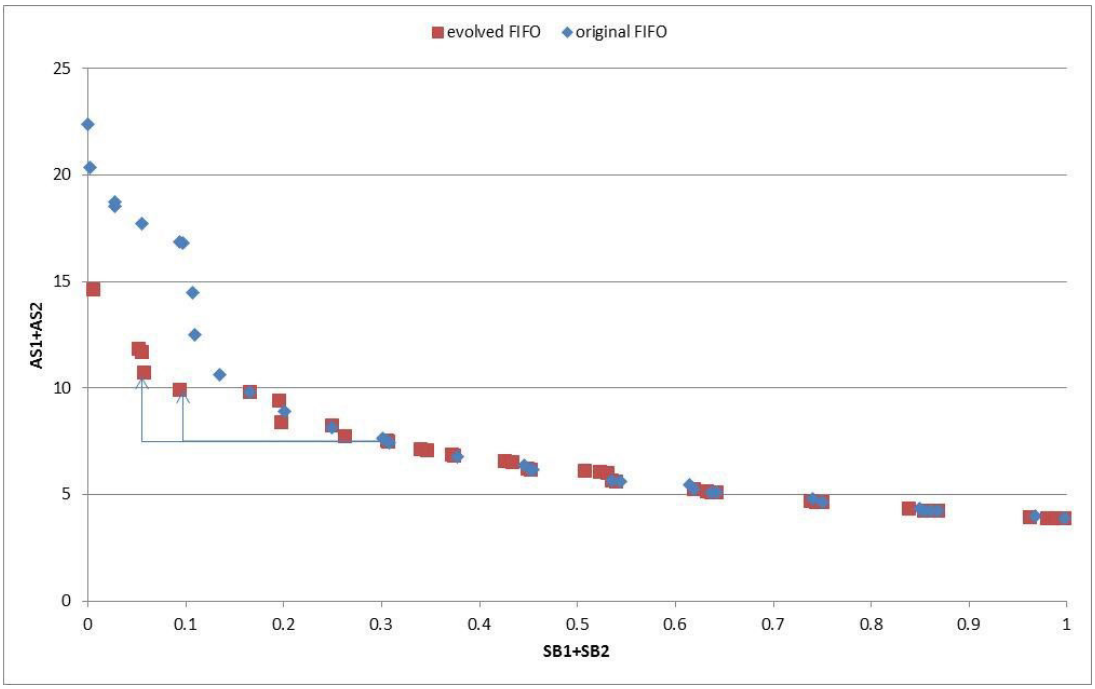


Figure 63: Comparison of partial Pareto front of evolved and original model

5-3 Solution from inventory experts

The inventory experts also proposed a solution according the arriving time of customer_2 was far more than Customer_1. The solution was to specify a producing priority for Kanban_2. That means when Kanban_2 arrived to the machine Kanban buffer, the machine chose to prioritize the item for Kanban_2. Based on this mechanism, we built a new model in Witness and maintained the other parameters. A series of 1039 random experiments were performed, and a comparison of the results to the original experiments is presented in Figure 64.

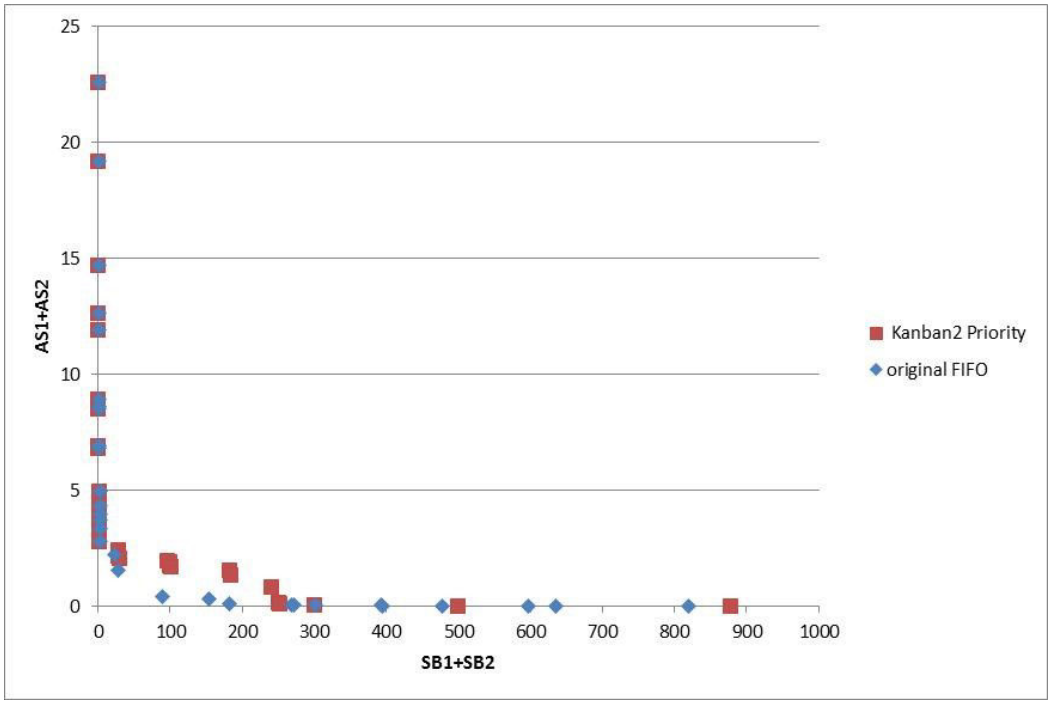


Figure 64: Kanban_2 priority Pareto vs original random FIFO Pareto

From Figure 64, we know that the two Pareto in the two dimensions (SB1+SB2, AS1+AS2) are similar, which means the limit states of two systems are similar. However, by comparing the solution in a horizontal way, we can find that the priority solution is better than original solution on the SB1+SB2.

To make a comparison to the evolved FIFO, we present the solution from evolved FIFO and Kanban2 Priority system in Figure 65. In the region [0, 1]×[0, 25], no solution from Kanban2 Priority system can dominate the Pareto line the of evolved FIFO. That means the evolved FIFO is superior to the Kanban2 Priority system.

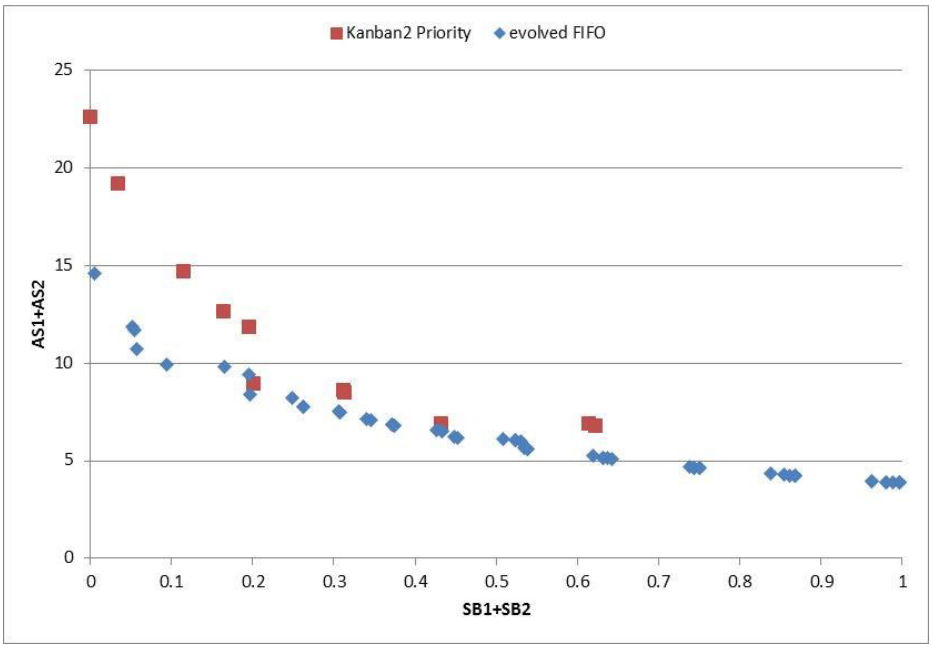


Figure 65: Kanban2 priority Pareto vs evolved model Pareto

In order to demonstrate the superior of evolved FIFO, we selected the experiments results by SB1<0.1 and SB<0.1 for evolved FIFO, Kanban2 priority and original FIFO (full experiments). We only considered the rest two evaluation parameter AS1 and AS2, and the Pareto of three groups of results are presented in Figure 66. From this figure, we can easily see that when SB1 and SB2 is satisfied (<0.1), the evolved FIFO can provide the best resolutions among three systems.

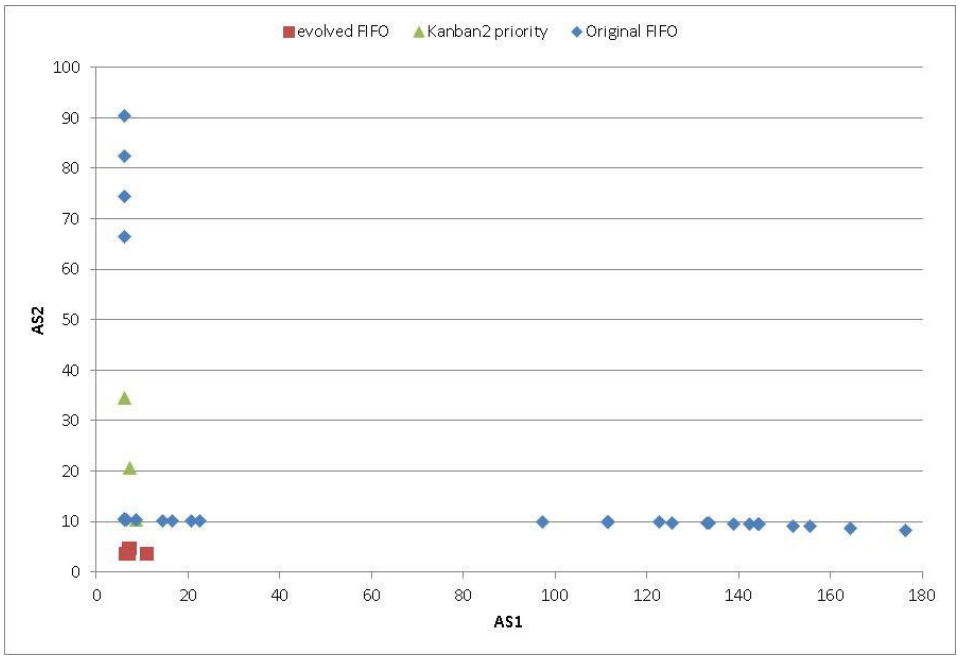


Figure 66: The Pareto of AS1 and AS2 for three systems(SB1<0.1 and SB2<0.1)

5-4 Conclusion

In this chapter, first we reviewed the general framework of inventive design process as proposed in Chapter 1. In this process, the inventive design is performed in the fifth step-model change. For this step, we proposed a systematic inventive process, which integrated GTC extraction (Chapter 2), GPC extraction (Chapter 3), EP/GTC-based GPC extraction (Chapter 4) and inventive problem solving (TRIZ). To illustrate this model change process, we made an example of a double Kanban, one machine supplying two types of items for two different stocks. After analysis, this model evolved by a new supplementary strategy. In the end, we also made a comparison against the solution the experts proposed. The results indicated that the evolved system had better performance than the experts proposed system. Unfortunately, our evolved system still did not achieve the ideal final results as the four evaluation parameter located in [0, 0.1].

However, in the case study, there are two technical contradictions and twelve generalized technical contradictions. This supports the answer to Q1, that GTC is more numerous than TC. It also proposed to answer to Q3, where choosing the GTC based on the number of associated evaluation parameters and the number of experiments in E1 and E2. The solving process of physical contradiction with context of the double Kanban indicated to us that the physical contradiction should be solved by TRIZ, and the context should be seen as superior system parameters during the inventive problem solving process.

Chapter 6 Conclusion and future prospectives

The introduction chapter introduced the problematic of our work via ten questions. The purposes of this chapter are to summarize the contributions and limitations of our work that leads us to discuss the answers to these questions and to propose a prospective for future research.

6-1 Reminder of the initial questions

In Chapter 1, we introduced the background for our thesis and our motivations to extend the concept of TRIZ contradiction in inventive problem solving in order to better formulate and analyze the inventive design problem. In order to organize our problems and present our contributions, Chapter 1 also proposed ten general questions that locate our problematic in three different levels. Let us recall the ten questions.

The first level of questions regarding the design theory is concerned with the concept of contradiction, which is one of the foundations of TRIZ:

Question 1: When no classical technical or physical contradiction exists, do the generalized technical and/or physical contradictions exist and are there a significant number of those contradictions? Can we always extract the generalized contradictions intrinsically the same way as Pareto, for example, from the behavioral representation and the objectives of the system? If so, then what are the consequences?

Question 2: How can the generalized contradictions be exhaustively identified and extracted?

Question 3: Once all generalized contradictions are known, how can the relevant contradictions be chosen or defined? Alternately, how can a relevant contradiction be defined in a straightforward manner?

The second level of questions regarding the methodology concerns the practical consequences of the new definition of contradictions for the inventive problem solving:

Question 4: Once the contradictions have been identified and extracted, how can we use them in the inventive problem solving process?

Question 5: How can we extract the relevant contradictions without exhaustive research, which is often too expensive and time consuming despite a posteriori filtering?

Question 6: Can we use the concept of a generalized contradiction to express the implicit knowledge from a system expert?

Finally, the third level of questions discusses the exploration of the relationship between optimization methods and TRIZ in order to develop cross-fertilization from a theoretical and/or a practical point of view.

Question 7: Can we use methods and concepts from the optimization in order to facilitate the identification of generalized contradictions?

Question 8: Is there a relationship between the Pareto concepts and the generalized contradictions?

Question 9: If this link exists, could it be used to identify generalized contradictions?

Question 10: Alternately, if this link exists, could it be exploited to facilitate the optimization process?

6-2 Contributions by chapters

6-2.1 Chapter 2: Extraction of generalized technical contradiction

In Chapter 2, we proposed a GTC identification and extraction model in the form of optimization programs and two solution strategies. The first solution strategy is an exhaustive searching algorithm that can identify and extract all GTC from an experimental table when the number of evaluation parameter is fewer than fifteen. The second extracts one GTC by an existing binary programming algorithm as a supplementary strategy when the number of evaluation parameters is greater than

fifteen. However, to the best of our knowledge, there is no powerful tool that can effectively solve the binary integer programming problem for the second strategy.

The exhaustive algorithm for GTC extraction presented in this chapter makes it possible to answer Q2 when the number of variables is fewer than fifteen. With this algorithm, we can also consider answering Q1 for technical contradiction by performing appropriate experiments. From these experiments, we made the following conclusions:

- The more available solutions there are that meet multiple evaluation objectives, the smaller is the possibility of obtaining a classical technical contradiction. When over 50% of the targets in the experiment table is satisfied, there is virtually no possibility of obtaining a classical technical contradiction. This point explains the difficulties encountered by human experts to formulate contradictions based on the classical model of TRIZ contradiction in some situations.
- The number of generalized technical contradictions seems to increase until the disappearance of the classical technical contradictions; however, it also gradually decreases as we increase the number of satisfied targets. There is always at least one generalized contradiction in the data, unless the problem is completely solved (i.e. all targets are satisfied; there is no Pareto).

The contradictions from the real case data did not fundamentally differ from those of the random data on similar characteristics. Thus, it seems that we can use randomly generated data to construct and validate the heuristic identification of technical contradictions. The data problem does not arise in the exact resolution algorithm proposed in the thesis, but it will arise for the heuristic algorithm designed to solve the same problem for systems that contain a large number of variables.

The application of the results of this chapter also provides partial answers to Q4 and Q6. As described in Chapter 2, the case study for the electrical circuit breaker was processed by GTC extraction, and the results were validated by experts. Among 117 GTCs, four were found and interpreted by experts, such interpretation essentially helps experts express implicit knowledge of the system, and we define GTC as our objective for future investigation.

6-2.2 Chapter 3: Extraction of generalized physical contradiction

In Chapter 3, we proposed a GPC identification and extraction algorithm, through the extraction of two GPC concepts, each of which involves many system states. The classical physical contradiction with context was proposed in this chapter by comparing the two states in each concept. The concept extraction was initially transformed into a binary integer programming problem, and then because its constraint is a logical equation, we proposed a filter method for filtering all possible concept states.

A possible use for this algorithm, which was not expected at the beginning of the study, is the possibility of identifying the classical or generalized physical contradictions in the context of setting action parameters. This allows us to consider more applications as part of the inventive design and assistance in understanding the contradictions. Indeed, in the classical approach, the contradictions provided by experts never provide explicitly the context of validity of contradiction (the values of those variables not involved in the contradiction). For example, the contradictions found by experts could be understood or completed by defining the limits of their validity, or vice versa, the simple GTC contradictions close to the TRIZ classical contradictions could be proposed to the expert by adding the limits of their validity.

Sometimes, searching for a sample of contradictions helps identify the solutions of the optimization problem. Indeed, it is possible that in the set of experiments, searching Concepts 1 and 2 is separate, and two un-conflicting states could appear in each concept. In this case, the combination of the two concepts could be a potential solution for the original problem, but this part needs more statistical studies; we do not further discuss this situation.

In this chapter, Q2 is addressed by the GPC extraction algorithm. The case study for the electrical circuit breaker has no TC, but GTC; we performed GPC extraction to a given GTC, two concepts were obtained that could create numerous physical contradictions with context, and thus the result of the case study partially answers Q1. For the case study of a simple Kanban, we extracted GPC for two conflict evaluation parameters (Service Breakdown and Average Stock). There are 12.768 physical contradictions with context, we selected one and formulated the system of

contradiction. The situation of the contradiction was analyzed by experts, and thus the system of contradiction organized the knowledge of experts for the system and positively answered Q6.

6-2.3 Chapter 4: Identification of key parameters

The aim of Chapter 4 was to propose heuristics for selecting relevant parameters and their values for searching relevant generalized contradictions. For doing this, the identification of parameters, which is to classify discriminative parameters for two separate sets, was addressed in terms of machine learning methods in artificial intelligence. We first showed that the identification of parameters and values used in the concept of generalized physical contradiction is similar to a series of problems known as the “feature selection” problem in some areas of research. Among the methods of “feature selection”, those “signed feature selection” seem to best suit our case. Two methods were implemented and tested: Pearson correlation coefficient and Support Vector Machine (SVM); the best results were obtained by the SVM that we adopted. We developed an SVM-based methodology customized to our problems.

In addition, the feature selection-based action parameter selection provided a partial solution for Q5 because the selection process removed many irrelevant parameters that did not participate in the contradiction formulation. In Q5 we want to select the relevant contradictions from numerous contradictions, but thus far, we have not found criteria that define the relevant contradiction. The proposed selection method helps us make data preprocessing for filtering a large amount of irrelevant contradictions and reduce contradiction size, which indirectly contributes to the relevant contradiction extraction.

6-2.4 Chapter 5: Process of model change using three different algorithms

In Chapter 5, we first recalled the general framework for inventive design problem solving, and then a process for obtaining an inventive solution from the experiments results was proposed, that combines the GTC extraction algorithm (from Chapter 2), GPC extraction algorithm (from Chapter 3) and parameter selection strategy (from

Chapter 4). Under this process, we connected the two algorithms of general contradiction extraction and inventive solution. In the illustration part, a case study for double Kanban was introduced and improved under this inventive process, which is a promising result for the proposed inventive process. In the parameter selection step of double Kanban, we adopted EP-based GPC extraction. We selected action parameters and values oriented to satisfying each evaluation parameter. The full factorial experiments on the selection results indicated that this selection result could help us obtain a solution beyond the original Pareto experiments in order to improve the optimization results.

In the case study, we presented how to use ARIZ85C to solve the general physical contradiction with context, and finally obtain the improved solution. This process undoubtedly suggests the availability of the TRIZ solution on the general contradiction obtained with context; and this is partially what we want to answer for Q4.

6-3 Discussion by questions and prospective

In this part, we summarize the extent to which we answered the ten questions, and discuss what we should do to attempt to completely answer all ten questions in the future. Q1 is partially answered by the experiments conducted in Chapters 2 and 3. In the future, we would like to develop a multi-optimization algorithm that can approach Pareto, while facilitating GTC extraction.

The classical TRIZ model of contradiction is easier to solve because it is easier to interpret for a human expert of the domain: it is more significant. On the other hand, GSC is more difficult to interpret, and thus to solve, but it exists in any problematic situation for which no solution is known. This contradiction is illustrated in Figure 67.

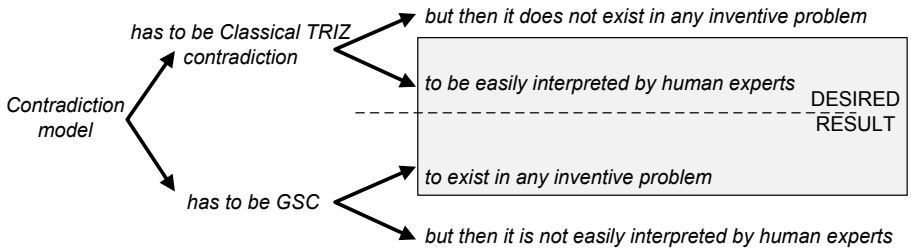


Figure 67: Contradiction of model to be used for inventive problems

As mentioned in 6-2.1 and 6-2.2, we answered Q2 by the GTC and GPC extraction algorithm when the number of evaluation parameters is fewer than fifteen and the binary action parameters are fewer than twelve. For other situations, because of computational limitation, we could not search the contradictions exhaustively, but proposed strategies to obtain at least some generalized contradictions.

In order to answer Q3, we need to define the relevant contradiction. The concept of "relevant" may differ with the specific problem. In order to become aware of the different situations and interesting selecting criteria, several practical experiments should be performed. Working with the actual software could allow this new knowledge in the future. As a starting point, we propose the following options for analyzing the set of contradictions:

- Filtering (variables involved in GTC or GPC), which is the goal of Chapter 4. Some action parameters do not always have an effect on the evaluation parameter involved in GTC. After removing these action parameters, we can remove several physical contradictions.
- Selecting contradictions with specific characteristics (EP set and experiment sizes). There are two key characteristics that are important for describing contradictions: experiment and evaluation parameter sizes for GSC, i.e. the size of (E1+E2) and that of (Y1+Y2). GSC experiment size indicates the amount of knowledge about the problematic situation, and thus we propose choosing the GSC associated with the largest experiment size. On the other hand, solving a contradiction also means satisfying all the evaluation parameters implied in the contradiction. Therefore, the other factor we need to consider is the size of the GSC evaluation parameter. We propose choosing the GSC with the largest size.
- Selecting a GPC with context and one or several variables; the notion of context is proposed in Chapter 3.

With time, if some specific outputs appeared to be interesting, heuristics could be built for working on more variables or providing the selected set directly without explicitation of all generalized contradiction.

For Q4, we conducted two trials on the resolution of simple Kanban (in Chapter 3) and double Kanban (in Chapter 5). For simple Kanban, the resolution we obtained

from ARIZ85C was to change single Kanban to several Kanban running in the model; subsequently, Pareto goes beyond the original Pareto of simple Kanban. However, this proposed solution only reinvented the existing simple Kanban system. Nevertheless, the solution might be cheaper, because it takes from the existing solutions only that which is required for the specific situation. In order to advance and test the practical problem of Q4, more case studies have to be performed. At the same time, for answering Q6, we need to inquire the experts, and determine whether they can validate the obtained contradictions or learn about the system's behavior.

For Q5, Chapter 2 proposed a binary programming model for single GTC extraction, if we were to transform our requirements into a constraint for the binary program model, we would change our relevant GTC extraction problem to a binary programming problem, which would provide us with the opportunity of solving such problem by the binary programming algorithm, similar to branch and cut method. In addition, we want to develop a heuristic algorithm that directly searches for GTC in the experiment results. For GPC filtering, we will progressively develop a utilization of the feature selection algorithm in GTC-based GPC extraction; the ideal situation is to directly use the results to formulate a generalized physical contradiction without performing new experiments.

For Q7, the answer is obvious: GTC or GPC extraction is a searching problem; we proposed using the optimization model to find, GTC and GPC in Chapter 2 and 3, and proposed an SVM model (where the learning process is an optimization process) for action parameter selection, which is to facilitate GPC extraction.

For investigating Q8, as mentioned in Chapter 2, if we were to define an acceptable range, the real Pareto could be changed into "binary" Pareto. The relationship between the concept of Pareto and generalized contradiction could be connected by binary Pareto. From the GTC perspective, binary Pareto involves the entire GTC set. From the GPC perspective, with the approach of Chapter 4, we can extract some values of action parameters to help us locate the Pareto points. In the future, we will attempt to further investigate the relationship between Pareto and binary Pareto.

Q9 could be partially answered by binary Pareto because the latter involves all GTCs. Thus, in the future, we will attempt to use a multi-objective optimization algorithm

to maximize the evaluation parameters of our system and output the binary value, similar to the two-objective problem shown in Figure 68. Unlike the real value output, the solution falls in the same area that has the same binary outputs. This way, we can determine whether the optimization algorithm can help us effectively and efficiently obtain binary Pareto, which facilitates GTC extraction. From this figure, we also know that, after binarization of the evaluation parameters, the GTC extraction is on the binary result of experiments, but the binary result (binary Pareto) cannot be the points on Pareto. Therefore, GTC extraction, after defining the acceptable range, does not truly need the real Pareto.

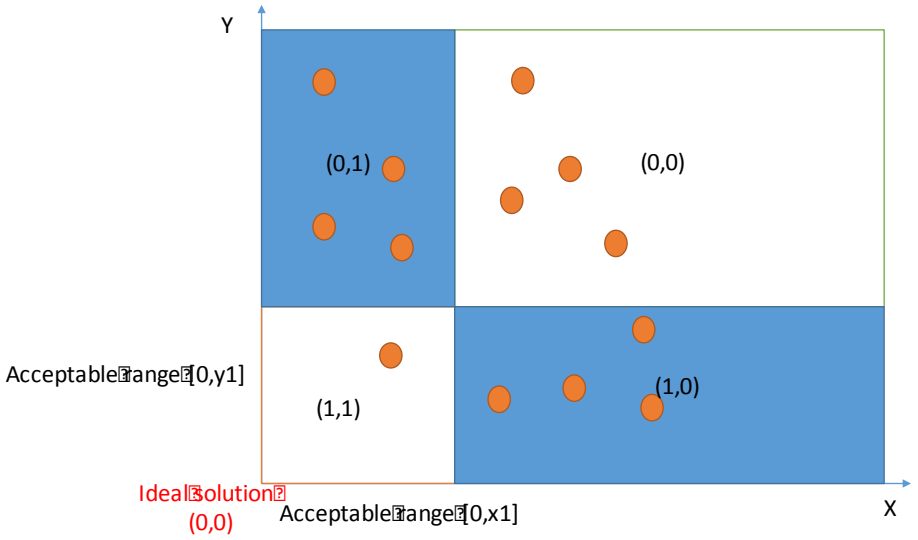


Figure 68: Binary output for multi-objective optimization

With regard to Q10, the EP-based GPC extraction strategy showed that it intensifies the extreme points for any combination of those evaluation parameters not obtained by the optimization process. Thus, we suppose that the GTC-based GPC extraction strategy in Chapter 4 can also help us reveal the extreme point for Y1 and Y2 (the two sides of GTC) as well as Y1+Y2 (candidate solution). Therefore, this strategy could assist the multi-objective optimization to search for objective Y1+Y2. This means that after optimization, we could use the action parameter with values from the EP-based GPC extraction to overcome the Pareto of optimization.

Bibliography

1 Altshuller, G.S. Algorithm of inventive problem solving. 1985. Available from: http://www.evolocus.com/textbooks/ariz85c.pdf [Accessed 3 February 2016]

2 Bonnardel, N. Understanding and Supporting Creativity in Design. *Knowledge-based system*, 2000, pp. 505–513. Available from: http://opus.bath.ac.uk/15196/

3 Gano, D.L. *Effective problem solving: A New Way of Thinking*. 2001.

4 Dorst, K. *Describing Design: A Comparison of Paradigms*. industrial engineer,Delft university of technology, 1997.

5 Reitman, W.R. *Heuristic decision procedures, open constraints, and the structure of ill-defined problem*,in M.W.Shelly. *Human judgments and optimality*, New York: Wiley, 1964. pp. 282-315.

6 Restrepo, J. *et al.* Problem Structuring and Information Access in Design. *Journal of Design Research of Design Research*, 2004, 4 (2).

7 Simon, H. A. The structure of ill structured problems. *Artificial Intelligence*, 1973, 4 (3-4), pp. 181–201.

8 Hatchuel, A. *et al.* A New Approach of Innovative Design : an Introduction To C-K Theory. In: Folkeson, A. et al. (eds.) *International Conference on Engineering Design*. Stockholm, 2003, pp. 1–15.

9 Cavallucci, D. *et al.* Integrating Altshuller's development laws for technical systems into the design process. *CIRP Annals - Manufacturing Technology*, 2001, 50, pp. 115–120.

10 Rosenman, M. A. *et al.* Creativity in Design Using a Design Prototype Approach. In: Associates, L. E. (ed.) *Modeling Creativity and Knowledge based Creative Design*. LAWRENCE ERLBAUM ASSOCIATES: NJ,USA, 1993, pp. 111–138.

11 Dorst, C.H. Design Problems and Design Paradoxes. *Design Issues*, 2006, 22 (3), pp. 4–17.

12 Hatchuel, A. Towards design theory and expandable rationality: The unfinished programme of Herbert Simon. *Journal of Management and Governance*, 2001, 5 (3-4), pp. 260–273.

13 Maher, M. Lou *et al.* Formalising Design Exploration as Co-evolution. In: *Advances in Formal Design Methods for CAD: Proceedings of the IFIP WG5.2*

Workshop on Formal Design Methods for Computer-Aided DesignCAD. 1996, pp. 3–30.

14 Gero, J.S. Design Prototypes: A Knowledge Representation Schema for Design. *AI Magazine*, 1990, 11 (4), pp. 26-36.

15 Boden, M. a. *The Creative Mind: Myths and Mechanism*. Group, T. francis (ed.) London and New York, 2004.

16 Altshuller, G.S. *Creativity as an exact science: the theory of the solution of inventive problems*. New York, NY: Gordon & Breach Science Publishing, New York, 1984.

17 Goel, V. *et al.* The structure of design problem spaces. *Cognitive science*, 1992, 16 (3), pp. 395–429.

18 Collette, Y. *et al. Multiobjective Optimization: Principles and Case Studies*. Springer New York, 2004.

19 Spence, L. *et al. Hegel for biggnners*. Trumpington, Cambridge: Icon, 1996.

20 Khomenko, N. OTSM -TRIZ Problem Solving Process : Solutions and Their Classification. In: Proceedings of TRIZ future Conference 2002. Strasbourg, 2002, pp. 295–299.

21 Altshuller, G. *The innovation Algorithm: TRIZ, systematic innovation and technical creativity*. Technial innovation ctr, 1998.

22 Ilevbare, I.M. *et al.* A review of TRIZ, and its benefits and challenges in practice. *Technovation*, 2013, 33 (2-3), pp. 30–37.

23 Dubois, S. *et al.* Comparison of non solvable problem solving principles issued from CSP and TRIZ. In: Cascini, G., IFIP 20th World Computer Congress, Proceedings of the Seconde Topical Session on Computer-Aided Innovation, Milano, Italy, September 7-10, 2008. Springer, pp. 1–12.

24 Cavallucci, D. *et al.* On contradiction clouds. In: *Procedia Engineering*. 2011, pp. 368–378.

25 Baldussu, A. *et al.* Network of contradictions analysis and structured identification of critical control parameters. In: *Procedia Engineering*, 2011, 9, pp. 3–17.

26 Cavallucci, D. *et al.* Towards inventive design through management of contradictions. In: *15th International CIRP Design Seminar.* 2005, p. 10.

27 Khomenko, N. *et al.* OTSM Network of Problems for representing and analysing problem situations with computer support. In: *Trends in computer aided innovation*. 2007, pp. 77–88.

28 Khomenko, N. *et al.* A framework for OTSM TRIZ-based computer support to be used in complex problem management. *International Journal of Computer Applications in Technology*, 2007, 30 (1/2), p. 88.

29 Rousselot, F. *et al.* Towards a formal definition of contradiction in inventive design. *Computers in Industry*, 2012, 63 (3), pp. 231–242.

30 Dubois, S. *et al.* A dialectical based model coherent with inventive and optimization problems. *Computers in Industry*, 2009, 60 (8), pp. 575–583.

31 Dubois, S. *et al.* Interpretation of a General Model for Inventive Problems , the Generalized System of Contradictions. In: *Proceedings of the 19th CIRP Design Conference-Competitive Design*. 2009.

32 Dubois, S. *et al.* From Simulation to Invention, Beyond the Pareto-Frontier. In: Weber, C. *et al*. *Proceeding of the 20th International Conference on Engineering Design*, *Milan, Italy. 2015,* pp. 245-254.

33 Rasovska, I. *et al.* Study of different principles for automatic identification of generalized system of contradiction out of design of experiments. In: *8th international Conference of modeling and simulation*. *Hammamet, May 2010,* pp. 1096-1101

34 Dubois, S. *et al.* Towards an automatic extraction of Generalized System of Contradictions out of solutionless Design of Experiments. In: Tan, R., *et al. Grouth and Development of Computer-Aided Innovation, Harbin, China,* 2009, Springer, pp. 1–10.

35 Cascini, G. *et al.* Computer-aided analysis of patents and search for TRIZ contradictions. *International Journal of Product Development*, 2007, 4 (1), pp. 52–67.

36 Cavallucci, D. *et al.* Initial situation analysis through problem graph. *CIRP Journal of Manufacturing Science and Technology*, 2010, 2 (4), pp. 310–317.

37 *Final Report Summary - TECH-IT-EASY (IT tool to support SMEs in systematic innovation, based on consolidated methodology and innovation-knowledge domain structured through specific ontologies)*. 2013.

38 Cugini, U. *et al.* Enhancing interoperability in the design process, the PROSIT approach. *IFIP International Federation for Information Processing*, 2007, 250, pp. 189–199.

39 Altshuller, G. *et al. 40 Principles: Triz Keys to Technical Innovation*,Technical Innovation Center, Inc. 1997. p.144

40 Conrardy, C. *et al.* Automatic extraction of a contradiction genealogic tree from optimization with an object-oriented simulator. In: Cvallucci, D. et al., *4th IFIP WG5.4 Working Conference, CAI 2011, Strasbourg, France, 2011*. Springer, 2011, pp.71-94.

41 Shlens, J. *A tutorial on Principal Component Analysis*. 2005. [Online]. Available: http://arxiv.org/pdf/1404.1100v1.pdf

42 Commander, C. *A survey of the quadratic assignment problem, with applications.* Bachelor of science, University of Florida, 2003, P.27.

43 Larose, D.T. *Data Mining Methods and Models*. Springer, 2006.

44 Bjorke, J.T. *et al.* SERIATION : AN IMPLEMENTATION AND CASE STUDY. *Computers, Environment and Urban System*, 1996, 20 (6), pp. 427–438.

45 Kusiak, A. *Integer programming approach to clustering problem*. Working paper, department of industrial engineer, Technical university of Nova Scotia, Halifax, 1983.

46 Liiv, I. Seriation and Matrix Reordering Methods : An Historical Overview. Statistical Analysis and Data Mining, 2010,3,pp.70-91.

47 Jain, A.. *et al.* Data Clustering: A Review. *ACM computing surveys*, 1999, pp.264-323.

48 Van Mechelen, I. *et al.* Two-mode clustering methods: a structured overview. *Statistical methods in medical research*, 2004, 13 (5), pp. 363–394.

49 Marotorchino, F. A unified approach of the Block-Seriation Problems. *Journal of Applied Stochastic Models and Data Analysis*, 1987, 3 (2).

50 Doreian, P. An intuitive introduction to blockmodeling with examples. *Bulletin de Méthodologie Sociologique*, 1999, 61, pp. 5–34.

51 Chen, D.-S. *et al. Applied Integer Programming*, Wiley-blac. 2011, p.468

52 Cormen, T. *et al. Introduction to algorithm*, MIT Press. 2009, p.1312.

53 Knuth, D.. The art of computer programming. In: Boston: Addison-Wesley Professional, 2011, pp. 281–319.

54 Montgomery, D. *Design and Analysis of Experiments*. Willey, 5th Edition. 2001. p.752

55 Khomenko, N. *et al.* Enhancing ECN ' s abilities to address inventive strategies using OTSM-TRIZ. *International journal of Collaborative Engineering*, 2009, 1, pp. 98–113.

56 Lin, L. *et al.* Algorithm for Identifying generalised technical contradiction in experiments. *Journal Européen des Systèmes Automatisés*, 2012, 24 (1), pp. 67–85.

57 Burgard, L. *et al.* Sequential experimentation to perform the Analysis of Initial Situation. In: *TRIZ Future 2011, Dublin, Ireland, 2011*, pp. 35–44.

58 Lin, L. *et al.* An exact algorithm to extract the generalized physical contradiction. *International journal on interactive design and manufacturing*, 2015, 9 (3), pp. 185–191.

59 Lin, L. *et al.* An algorithm to extract the Generalized Physical Contradiction. In: *Proceedings of the 2014 International Conference on Innovative Design and manufacturing, Montreal, 2014*, pp. 129–135.

60 Lewis, D.D. *et al.* A comparison of two learning algorithms for text categorization. In: *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, NV, 1994,* pp. 81–93.

61 Yang, Y. *et al.* A comparative study on feature selection in text categorization. In: Danyluk A., ed. *Proceedings of the 14th International Conference of Machine Learning. San Francisco: Morgan Kaufmann Publisher, 1997,* pp. 412–420.

62 Guyon, I. *et al.* Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, 2002, 46 (1-3), pp. 389–422.

63 Saeys, Y. *et al.* A review of feature selection techniques in bioinformatics. *Bioinformatics (Oxford, England)*, 2007, 23 (19), pp. 2507–17.

64 Liu, H. *et al. Feature selection for knowledge discovery and data mining*. New York, NY: Kluwer Academic Publishers, 1998.

65 Guyon, I. An Introduction to Variable and Feature Selection. *the journal of Machine Learning Research*, 2003, 3, pp. 1157–1182.

66 Dash, M. *et al.* Feature Selection for Classification. *Intelligent Data Analysis*, 1997, 1 (97), pp. 131–156.

67 Ding, C. *et al.* Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology,* 2005, 3 (2), pp. 185–205.

68 Kenji, K. *et al.* The Feature Selection Problem: Tradictional Methods and a New Algorithm. In: *Proceedings of the 10th National Conference on Artificial Intelligence*. San Jose, CA, 1992, pp. 129–134.

69 Hall, M.A. Correlation-Based Feature Selection for Discrete and Numeric class Machine Learning. In: *ICML 2000 Proceedings of the Seventeenth International Conference on Machine learning*. 2000, pp. 359–366.

70 Vladimir N, V. *The Nature of Statistical Learning Theory*. Springer New York, 1995.

71 Hsu, C. *et al. A Practical Guide to Support Vector Classification*. National Taiwan University, 2003.

72 Cristianini, N. *et al. an introduction to support vector machines and other kernel-based learning methods*. New York, NY: cambridge university press New York, NY, USA, 2000.

73 Cameron, G. *ARIZ Explored: A step-by-step guide to ARIZ the Algorithm for Solving Inventive Problems*,Createspace, August 2015, p.152

Appendix

Appendix 1: Example for illustrating two strategies

		Action Paramter										Y1		Y2	
		A1	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**	**A9**	**A10**	**EP1**	**EP2**	**EP3**	**EP4**
E1	e1	0	1	1	0	0	1	0	1	1	1	1	1	0	0
	e2	0	1	1	0	1	1	0	0	0	0	1	1	0	0
	e3	0	1	1	0	1	1	0	0	0	0	1	1	0	0
	e4	0	1	1	0	1	1	0	1	0	0	1	1	0	0
	e5	0	1	1	0	1	1	0	0	1	1	1	1	0	0
	e6	0	1	1	0	1	1	0	1	1	0	1	1	0	0
	e7	0	1	1	0	0	0	1	0	0	1	1	1	0	0
	e8	0	1	1	0	0	0	0	1	0	0	1	1	0	0
	e9	0	1	1	0	0	0	0	0	1	1	1	1	0	0
	e10	0	1	1	0	0	0	0	1	0	1	1	1	0	0
	e11	0	1	1	0	0	0	0	0	0	1	1	1	0	0
	e12	0	1	1	0	0	0	0	1	1	0	1	1	0	0
	e13	0	1	1	0	1	0	0	1	1	0	1	1	0	0
	e14	0	1	1	0	1	0	0	1	0	0	1	1	0	0
	e15	0	1	1	0	1	0	0	0	0	0	1	1	0	0
	e16	0	1	1	0	1	0	0	1	1	0	1	1	0	0
	e17	0	1	1	0	1	0	0	1	0	0	1	1	0	0
	e18	0	1	1	0	1	0	0	0	0	0	1	1	0	0
	e19	0	1	1	0	1	0	0	0	0	1	1	1	0	0
E2	e20	0	1	0	0	0	1	1	0	1	1	0	0	1	1
	e21	0	1	0	0	0	1	1	1	0	0	0	0	1	1
	e22	0	1	0	0	0	1	1	0	1	0	0	0	1	1
	e23	0	1	0	0	0	1	1	1	0	1	0	0	1	1
	e24	0	1	0	0	0	1	1	0	1	1	0	0	1	1
	e25	0	1	0	0	0	1	1	1	0	0	0	0	1	1
	e26	0	1	0	0	0	1	1	0	1	0	0	0	1	1
	e27	0	1	0	0	0	1	1	1	0	1	0	0	1	1
	e28	0	1	0	0	0	1	1	0	0	1	0	0	1	1
	e29	0	1	0	0	0	1	0	1	0	0	0	0	1	1
	e30	0	1	0	0	0	1	0	1	1	0	0	0	1	1
	e31	0	1	0	0	0	1	0	0	0	0	0	0	1	1
	e32	0	1	0	0	0	1	0	0	1	0	0	0	1	1
	e33	0	1	0	0	0	1	0	0	1	0	0	0	1	1
	e34	1	1	0	0	0	1	1	0	1	0	0	0	1	1
	e35	1	1	0	0	0	1	1	0	0	1	0	0	1	1
	e36	1	1	0	0	0	1	0	0	0	0	0	0	1	1

	e37	1	1	0	0	0	1	0	0	0	0	0	0	1	1
	e38	1	1	0	0	0	1	0	0	0	1	0	0	1	1
	e39	1	1	0	0	0	1	0	1	0	0	0	0	1	1
	e40	0	1	0	1	0	1	0	0	1	1	0	0	1	1
	e41	0	1	0	1	0	1	0	1	1	0	0	0	1	1
	e42	0	1	0	1	0	1	1	0	1	1	0	0	1	1
	e43	0	1	0	1	0	1	1	1	1	0	0	0	1	1
	e44	0	1	0	1	0	1	0	0	1	1	0	0	1	1
	e45	0	1	0	1	0	1	0	1	1	0	0	0	1	1
	e46	0	1	0	1	0	1	0	0	1	0	0	0	1	1
	e47	1	1	0	1	0	1	1	0	1	1	0	0	1	1
	e48	1	1	0	1	0	1	1	1	1	1	0	0	1	1
	e49	1	1	0	1	0	1	0	1	1	1	0	0	1	1
	e50	1	1	0	1	0	1	0	1	1	0	0	0	1	1
	e51	1	1	0	1	0	1	0	0	0	1	0	0	1	1
	e52	1	1	0	1	0	1	0	1	0	1	0	0	1	1
	e53	0	0	0	0	0	1	1	0	0	0	0	0	0	1
	e54	0	0	0	0	0	1	0	0	1	0	0	0	0	1
	e55	0	0	0	0	0	1	0	0	1	0	0	0	0	1
	e56	0	0	0	0	0	1	0	0	0	0	0	0	0	1
	e57	0	0	0	0	0	1	1	0	0	0	0	0	0	1
	e58	0	0	0	0	0	1	0	0	1	0	0	0	0	1
	e59	0	0	0	0	0	1	0	0	1	0	0	0	0	1
	e60	0	0	0	0	0	1	0	0	0	0	0	0	0	1
	e61	1	0	0	0	0	1	1	1	0	0	0	0	0	1
	e62	1	0	0	0	0	1	1	1	0	0	0	0	0	1
	e63	1	0	0	0	0	1	1	1	1	1	0	0	0	1
	e64	1	0	0	0	0	1	0	0	1	0	0	0	0	1
	e65	0	0	0	1	0	1	1	0	0	0	0	0	0	1
	e66	0	0	0	1	0	1	1	1	0	1	0	0	0	1
E0	e67	0	0	0	1	0	1	1	0	1	0	0	0	0	1
	e68	0	0	0	1	0	1	1	0	0	0	0	0	0	1
	e69	0	0	0	1	0	1	1	1	0	1	0	0	0	1
	e70	0	0	0	1	0	1	1	0	1	0	0	0	0	1
	e71	0	0	0	1	0	1	1	0	0	1	0	0	0	1
	e72	0	0	0	1	0	1	0	0	1	0	0	0	0	1
	e73	0	0	0	1	0	1	0	0	0	1	0	0	0	1
	e74	0	0	0	1	0	1	0	1	0	1	0	0	0	1
	e75	1	0	0	1	0	1	0	0	1	1	0	0	0	1
	e76	1	0	0	1	0	1	1	1	0	0	0	0	0	1
	e77	1	0	0	1	0	1	0	0	1	1	0	0	0	1
	e78	1	0	0	1	0	1	0	1	0	0	0	0	0	1
	e79	1	0	0	1	0	1	0	0	0	0	0	0	0	1
	e80	0	1	0	0	0	0	1	0	1	0	0	0	1	0
	e81	0	1	0	0	0	0	0	1	0	0	0	0	1	0

	e82	0	1	0	0	0	0	0	0	1	0	0	0	1	0
	e83	0	1	0	0	0	0	0	1	1	1	0	0	1	0
	e84	1	1	0	0	0	0	1	1	1	0	0	0	1	0
	e85	0	1	0	1	0	0	1	1	0	0	0	0	1	0
	e86	0	1	0	1	0	0	1	0	1	0	0	0	1	0
	e87	0	1	0	1	0	0	1	1	0	0	0	0	1	0
	e88	0	1	0	1	0	0	1	0	1	0	0	0	1	0
	e89	0	1	0	1	0	0	1	1	1	1	0	0	1	0
	e90	0	1	0	1	0	0	0	1	1	1	0	0	1	0
	e91	0	1	0	1	0	0	0	1	0	0	0	0	1	0
	e92	1	1	0	1	0	0	1	1	1	1	0	0	1	0
	e93	1	1	0	1	0	0	0	0	1	1	0	0	1	0
	e94	1	1	1	0	0	1	1	0	1	1	0	1	0	0
	e95	1	1	1	0	0	1	0	1	0	1	0	1	0	0
	e96	1	1	1	0	0	1	0	1	0	1	0	1	0	0
	e97	1	1	1	0	0	1	0	1	0	0	0	1	0	0
	e98	1	1	1	0	0	1	0	0	1	0	0	1	0	0
	e99	1	1	1	0	1	1	0	1	1	0	0	1	0	0
	e100	1	1	1	0	1	1	0	0	1	1	0	1	0	0
	e101	1	1	1	0	1	1	0	0	0	0	0	1	0	0
	e102	1	1	1	0	0	0	0	1	0	0	0	1	0	0
	e103	1	1	1	0	0	0	0	1	0	0	0	1	0	0
	e104	1	1	1	0	0	0	0	1	1	0	0	1	0	0
	e105	1	1	1	0	0	0	0	0	0	1	0	1	0	0
	e106	1	1	1	0	1	0	0	0	0	0	0	1	0	0
	e107	1	1	1	0	1	0	0	1	1	0	0	1	0	0
	e108	1	1	1	0	1	0	0	1	1	1	0	1	0	0
	e109	0	1	1	1	0	1	0	1	1	0	1	0	0	0
	e110	0	1	1	1	0	1	0	1	1	0	1	0	0	0
	e111	0	1	1	1	0	1	0	0	0	1	1	0	0	0
	e112	0	1	1	1	1	1	0	0	1	0	1	0	0	0
	e113	0	1	1	1	1	1	0	0	0	1	1	0	0	0
	e114	0	1	1	1	0	0	1	0	0	0	1	0	0	0
	e115	0	1	1	1	0	0	0	0	1	1	1	0	0	0
	e116	0	1	1	1	0	0	0	0	1	1	1	0	0	0
	e117	0	1	1	1	0	0	0	0	0	0	1	0	0	0
	e118	0	1	1	1	0	0	0	1	1	0	1	0	0	0
	e119	0	1	1	1	0	0	0	1	1	0	1	0	0	0
	e120	0	1	1	1	1	0	0	0	1	0	1	0	0	0
	e121	1	1	1	1	0	1	0	1	1	1	0	0	0	0
	e122	1	1	1	1	0	1	0	1	1	1	0	0	0	0
	e123	1	1	1	1	0	1	0	0	0	1	0	0	0	0
	e124	1	1	1	1	0	1	0	0	1	1	0	0	0	0
	e125	0	0	1	0	0	1	1	0	1	0	0	0	0	0
	e126	0	0	1	0	0	1	0	1	0	1	0	0	0	0

	e127	0	0	1	0	0	1	0	1	0	1	0	0	0	0
	e128	0	0	1	0	0	1	0	0	1	0	0	0	0	0
	e129	0	0	1	0	0	1	0	0	0	1	0	0	0	0
	e130	0	0	1	0	0	1	0	1	1	1	0	0	0	0
	e131	1	0	1	0	0	1	1	1	1	0	0	0	0	0
	e132	1	0	1	0	0	1	0	1	1	0	0	0	0	0
	e133	1	0	1	0	0	1	0	0	1	1	0	0	0	0
	e134	1	0	1	0	0	1	0	1	1	0	0	0	0	0
	e135	1	0	1	0	0	1	0	0	1	1	0	0	0	0
	e136	1	0	1	0	0	1	0	1	0	0	0	0	0	0
	e137	1	0	1	0	0	1	0	1	0	1	0	0	0	0
	e138	1	0	1	0	0	1	0	0	1	0	0	0	0	0
	e139	0	0	1	1	0	1	0	1	1	1	0	0	0	0
	e140	0	0	1	1	0	1	0	1	1	0	0	0	0	0
	e141	0	0	1	1	0	1	0	1	0	1	0	0	0	0
	e142	0	0	1	1	0	1	0	1	1	1	0	0	0	0
	e143	0	0	1	1	0	1	0	1	1	0	0	0	0	0
	e144	0	0	1	1	0	1	0	1	0	1	0	0	0	0
	e145	1	0	1	1	0	1	0	1	1	1	0	0	0	0
	e146	1	0	1	1	0	1	0	1	0	0	0	0	0	0
	e147	1	0	1	1	0	1	0	0	1	1	0	0	0	0
	e148	0	1	0	0	1	1	1	1	0	0	0	0	0	0
	e149	0	1	0	0	1	1	0	0	1	0	0	0	0	0
	e150	1	1	0	0	1	1	0	0	0	1	0	0	0	0
	e151	1	1	0	0	1	1	0	0	0	1	0	0	0	0
	e152	1	1	0	0	1	1	0	1	1	1	0	0	0	0
	e153	1	1	0	0	1	1	0	0	0	0	0	0	0	0
	e154	0	1	0	1	1	1	1	1	1	0	0	0	0	0
	e155	0	1	0	1	1	1	1	1	1	0	0	0	0	0
	e156	0	1	0	1	1	1	0	1	0	0	0	0	0	0
	e157	1	1	0	1	1	1	1	0	0	1	0	0	0	0
	e158	1	1	0	1	1	1	0	1	0	0	0	0	0	0
	e159	1	1	0	1	1	1	0	1	0	1	0	0	0	0
	e160	1	1	0	1	1	1	0	0	1	0	0	0	0	0
	e161	0	0	0	0	1	1	0	1	1	0	0	0	0	0
	e162	0	0	0	0	1	1	1	1	1	1	0	0	0	0
	e163	0	0	0	0	1	1	1	1	0	1	0	0	0	0
	e164	0	0	0	0	1	1	0	1	1	0	0	0	0	0
	e165	0	0	0	0	1	1	0	1	0	1	0	0	0	0
	e166	1	0	0	0	1	1	0	0	0	0	0	0	0	0
	e167	1	0	0	0	1	1	1	0	0	1	0	0	0	0
	e168	1	0	0	0	1	1	1	0	0	0	0	0	0	0
	e169	1	0	0	0	1	1	1	1	0	1	0	0	0	0
	e170	1	0	0	0	1	1	0	0	0	0	0	0	0	0
	e171	1	0	0	0	1	1	0	1	1	1	0	0	0	0

	e172	0	0	0	1	1	1	1	1	0	0	0	0	0	0
	e173	0	0	0	1	1	1	0	1	1	1	0	0	0	0
	e174	0	0	0	1	1	1	0	1	0	0	0	0	0	0
	e175	0	0	0	1	1	1	1	1	0	0	0	0	0	0
	e176	0	0	0	1	1	1	0	1	1	1	0	0	0	0
	e177	0	0	0	1	1	1	0	1	0	0	0	0	0	0
	e178	0	0	0	1	1	1	0	0	1	1	0	0	0	0
	e179	1	0	0	1	1	1	1	0	0	1	0	0	0	0
	e180	1	0	0	1	1	1	1	1	0	0	0	0	0	0
	e181	1	0	0	1	1	1	1	1	1	1	0	0	0	0
	e182	1	0	0	1	1	1	1	0	0	1	0	0	0	0
	e183	1	0	0	1	1	1	1	1	0	0	0	0	0	0
	e184	1	0	0	1	1	1	1	1	1	1	0	0	0	0
	e185	1	0	0	1	1	1	1	1	1	0	0	0	0	0
	e186	1	0	0	1	1	1	1	0	1	0	0	0	0	0
	e187	1	0	0	1	1	1	0	1	0	0	0	0	0	0
	e188	1	0	0	1	1	1	0	1	1	0	0	0	0	0
	e189	1	1	1	1	1	1	0	1	0	1	0	0	0	0
	e190	1	1	1	1	1	1	0	1	0	0	0	0	0	0
	e191	1	1	1	1	1	1	0	0	0	1	0	0	0	0
	e192	1	1	1	1	1	1	0	1	1	1	0	0	0	0
	e193	1	1	1	1	1	1	0	0	1	1	0	0	0	0
	e194	0	0	1	0	1	1	1	1	0	0	0	0	0	0
	e195	0	0	1	0	1	1	0	0	0	1	0	0	0	0
	e196	0	0	1	0	1	1	0	1	0	1	0	0	0	0
	e197	0	0	1	0	1	1	0	1	0	0	0	0	0	0
	e198	1	0	1	0	1	1	0	0	1	0	0	0	0	0
	e199	1	0	1	0	1	1	0	0	1	0	0	0	0	0
	e200	1	0	1	0	1	1	0	0	1	1	0	0	0	0
	e201	1	0	1	0	1	1	0	1	0	0	0	0	0	0
	e202	0	0	1	1	1	1	1	0	0	0	0	0	0	0
	e203	0	0	1	1	1	1	0	1	1	0	0	0	0	0
	e204	0	0	1	1	1	1	0	0	1	0	0	0	0	0
	e205	0	0	1	1	1	1	0	1	0	0	0	0	0	0
	e206	0	0	1	1	1	1	0	0	0	0	0	0	0	0
	e207	0	0	1	1	1	1	0	1	1	1	0	0	0	0
	e208	1	0	1	1	1	1	0	0	1	1	0	0	0	0
	e209	0	0	0	0	0	0	1	1	0	1	0	0	0	0
	e210	0	0	0	0	0	0	1	1	0	1	0	0	0	0
	e211	0	0	0	0	0	0	1	0	1	1	0	0	0	0
	e212	1	0	0	0	0	0	1	0	0	1	0	0	0	0
	e213	1	0	0	0	0	0	0	1	0	0	0	0	0	0
	e214	0	0	0	1	0	0	0	0	1	1	0	0	0	0
	e215	0	0	0	1	0	0	0	0	1	1	0	0	0	0
	e216	1	0	0	1	0	0	1	0	1	1	0	0	0	0

	e217	1	0	0	1	0	0	1	0	1	1	0	0	0	0
	e218	1	0	0	1	0	0	1	0	0	0	0	0	0	0
	e219	1	0	0	1	0	0	1	0	1	0	0	0	0	0
	e220	1	0	0	1	0	0	0	1	1	0	0	0	0	0
	e221	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	e222	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	e223	1	1	1	1	0	0	0	1	1	0	0	0	0	0
	e224	1	1	1	1	0	0	0	1	0	0	0	0	0	0
	e225	0	0	1	0	0	0	1	1	1	1	0	0	0	0
	e226	0	0	1	0	0	0	1	0	0	0	0	0	0	0
	e227	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	e228	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	e229	0	0	1	0	0	0	0	0	1	1	0	0	0	0
	e230	1	0	1	0	0	0	0	1	0	0	0	0	0	0
	e231	1	0	1	0	0	0	0	0	1	0	0	0	0	0
	e232	1	0	1	0	0	0	0	1	1	0	0	0	0	0
	e233	1	0	1	0	0	0	0	1	0	1	0	0	0	0
	e234	0	0	1	1	0	0	1	0	1	0	0	0	0	0
	e235	0	0	1	1	0	0	0	1	0	1	0	0	0	0
	e236	0	0	1	1	0	0	0	0	1	0	0	0	0	0
	e237	0	0	1	1	0	0	0	1	0	1	0	0	0	0
	e238	0	0	1	1	0	0	0	0	1	0	0	0	0	0
	e239	1	0	1	1	0	0	0	1	0	0	0	0	0	0
	e240	1	0	1	1	0	0	0	0	1	1	0	0	0	0
	e241	1	0	1	1	0	0	0	1	0	0	0	0	0	0
	e242	1	0	1	1	0	0	0	0	1	1	0	0	0	0
	e243	1	0	1	1	0	0	0	0	0	1	0	0	0	0
	e244	1	0	1	1	0	0	0	0	0	0	0	0	0	0
	e245	0	1	0	0	1	0	1	0	1	1	0	0	0	0
	e246	0	1	0	0	1	0	1	0	0	1	0	0	0	0
	e247	0	1	0	0	1	0	1	0	1	1	0	0	0	0
	e248	0	1	0	0	1	0	1	0	0	1	0	0	0	0
	e249	0	1	0	0	1	0	1	0	1	0	0	0	0	0
	e250	0	1	0	0	1	0	1	1	1	0	0	0	0	0
	e251	0	1	0	0	1	0	1	1	0	0	0	0	0	0
	e252	0	1	0	0	1	0	0	1	1	1	0	0	0	0
	e253	0	1	0	0	1	0	0	0	0	0	0	0	0	0
	e254	1	1	0	0	1	0	1	1	0	1	0	0	0	0
	e255	1	1	0	0	1	0	1	1	1	1	0	0	0	0
	e256	1	1	0	0	1	0	0	0	0	0	0	0	0	0
	e257	1	1	0	0	1	0	0	0	0	0	0	0	0	0
	e258	1	1	0	0	1	0	0	1	1	1	0	0	0	0
	e259	1	1	0	0	1	0	0	1	1	1	0	0	0	0
	e260	0	1	0	1	1	0	1	0	1	1	0	0	0	0
	e261	0	1	0	1	1	0	0	0	1	1	0	0	0	0

	e262	0	1	0	1	1	0	0	1	1	1	0	0	0	0
	e263	0	1	0	1	1	0	1	0	1	1	0	0	0	0
	e264	0	1	0	1	1	0	1	0	1	0	0	0	0	0
	e265	0	1	0	1	1	0	1	0	0	0	0	0	0	0
	e266	0	1	0	1	1	0	0	0	1	1	0	0	0	0
	e267	0	1	0	1	1	0	0	1	1	1	0	0	0	0
	e268	0	1	0	1	1	0	0	0	0	1	0	0	0	0
	e269	1	1	0	1	1	0	0	1	1	0	0	0	0	0
	e270	1	1	0	1	1	0	0	1	0	1	0	0	0	0
	e271	1	1	0	1	1	0	0	1	0	0	0	0	0	0
	e272	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	e273	1	0	0	0	1	0	1	1	1	0	0	0	0	0
	e274	1	0	0	0	1	0	1	0	1	0	0	0	0	0
	e275	1	0	0	0	1	0	1	1	1	0	0	0	0	0
	e276	1	0	0	0	1	0	1	0	1	0	0	0	0	0
	e277	1	0	0	0	1	0	1	0	0	0	0	0	0	0
	e278	1	0	0	0	1	0	1	1	0	0	0	0	0	0
	e279	0	0	0	1	1	0	1	1	1	1	0	0	0	0
	e280	0	0	0	1	1	0	0	0	1	0	0	0	0	0
	e281	0	0	0	1	1	0	1	1	1	1	0	0	0	0
	e282	0	0	0	1	1	0	0	0	1	0	0	0	0	0
	e283	0	0	0	1	1	0	0	0	1	1	0	0	0	0
	e284	0	0	0	1	1	0	0	1	0	0	0	0	0	0
	e285	1	0	0	1	1	0	1	1	0	0	0	0	0	0
	e286	1	0	0	1	1	0	1	1	0	1	0	0	0	0
	e287	1	0	0	1	1	0	0	0	1	1	0	0	0	0
	e288	1	0	0	1	1	0	0	1	0	0	0	0	0	0
	e289	1	1	1	1	1	0	1	0	1	0	0	0	0	0
	e290	1	1	1	1	1	0	1	1	0	0	0	0	0	0
	e291	1	1	1	1	1	0	1	1	0	0	0	0	0	0
	e292	1	1	1	1	1	0	0	1	1	0	0	0	0	0
	e293	1	1	1	1	1	0	0	0	0	1	0	0	0	0
	e294	1	1	1	1	1	0	0	1	1	1	0	0	0	0
	e295	0	0	1	0	1	0	1	1	0	0	0	0	0	0
	e296	0	0	1	0	1	0	1	0	0	0	0	0	0	0
	e297	0	0	1	0	1	0	0	1	1	1	0	0	0	0
	e298	0	0	1	0	1	0	0	1	1	1	0	0	0	0
	e299	0	0	1	0	1	0	0	0	1	1	0	0	0	0
	e300	0	0	1	0	1	0	0	0	0	1	0	0	0	0
	e301	1	0	1	0	1	0	0	1	1	1	0	0	0	0
	e302	1	0	1	0	1	0	0	1	1	0	0	0	0	0
	e303	1	0	1	0	1	0	0	1	1	1	0	0	0	0
	e304	1	0	1	0	1	0	0	1	1	0	0	0	0	0
	e305	1	0	1	0	1	0	0	0	0	1	0	0	0	0
	e306	1	0	1	0	1	0	0	0	1	0	0	0	0	0

	e307	0	0	1	1	1	0	1	1	1	0	0	0	0	0
	e308	0	0	1	1	1	0	1	1	1	1	0	0	0	0
	e309	0	0	1	1	1	0	0	0	0	0	0	0	0	0
	e310	0	0	1	1	1	0	0	0	0	0	0	0	0	0
	e311	0	0	1	1	1	0	0	1	1	0	0	0	0	0
	e312	0	0	1	1	1	0	0	0	1	1	0	0	0	0
	e313	1	0	1	1	1	0	1	1	0	1	0	0	0	0
	e314	1	0	1	1	1	0	0	1	0	0	0	0	0	0
	e315	1	0	1	1	1	0	0	1	0	0	0	0	0	0
	e316	1	0	1	1	1	0	0	1	1	1	0	0	0	0

Résumé en français

Méthodes d'optimisation pour la conception inventive

Ce texte résume en français le contexte et les résultats des travaux de Lei Lin dans le cadre de ses travaux de thèse.

1. Introduction

1.1. Contexte de la thèse

Les travaux présentés dans le mémoire s'inscrivent dans les recherches menées depuis une quinzaine d'année au LGECO dans le cadre des applications de la théorie de résolution des problèmes d'invention (TRIZ) et de son extension dans divers domaines. La TRIZ est un ensemble de méthodes organisées en système, élaborées par G. Altshuller dans le but de faciliter l'invention d'objets physiques. La théorie sous-jacente à ces combinaisons de méthodes repose sur un ensemble d'hypothèses fondamentales issues de la dialectique et de l'analyse de l'évolution des systèmes techniques. La majorité des travaux «d'innovation» menés dans ce domaine sont d'ordre incrémental; ils ont pour but d'améliorer les diverses composantes des démarches existantes. Comme certains fondements de la TRIZ tels que la réflexion dialectique sont très génériques, quelques travaux ont tenté une démarche analogue à celle d'Altshuller pour d'autres domaines d'application tels que le management, la publicité, la logistique etc. Si tous ces travaux connaissent un certain succès, des difficultés récurrentes apparaissent dans la mise en œuvre de la formulation des problèmes par leur contradiction sous-jacente ; ce fut également le cas lorsque le laboratoire s'est engagé dans une recherche sur l'exploitation des modèles numériques de comportement de systèmes pour identifier par un calculateur les contradictions sous-jacentes aux limites du système. Ce faisant, il est clairement apparu que la définition du concept de contradiction tel que défini par la TRIZ était trop vague pour être utilisé comme tel par des outils informatiques, voire qu'elle pouvait même être à l'origine des difficultés rencontrées par les groupes d'utilisateurs humains pour définir les contradictions. Ces raisons pratiques et théoriques nous ont amené à revoir la définition de la contradiction et à proposer la contradiction généralisée. Cette nouvelle contradiction est encore plus difficile à

identifier pour l'humain ; en revanche elle est suffisamment précise pour être traitée par un ordinateur dès lors qu'on dispose de données sur le comportement d'un système. L'objet premier cette thèse est de proposer des moyens d'identifier ces contradictions à partir de données expérimentales ou de simulation de systèmes en vue de leur utilisation pour la compréhension des problèmes et leur résolution. Dans un second temps, ces nouveaux outils peuvent être utilisés pour tenter d'analyser et d'expliquer certaines difficultés pratiques rencontrées par l'humain lors de l'identification des contradictions. Enfin, les recherches menées pour la construction de la contradiction généralisée nous ont confortés dans l'idée qu'il y a un lien entre la théorie de l'optimisation et la TRIZ au travers du concept de contradiction. Les outils développés dans cette thèse visent également à explorer le lien entre la théorie de l'optimisation et celle de la résolution des problèmes d'invention via les concepts de contradiction généralisée et d'ensemble de Pareto. Afin de montrer la portée générale de l'approche, les exemples d'illustration pris dans la thèse sont tirés aussi bien de la conception d'objets que celle des processus de logistique interne. Les paragraphes suivants de cette introduction précisent progressivement (en résumé) le concept de contradiction, l'ensemble des questions auxquelles veut répondre la thèse, l'organisation de la recherche pour y arriver et l'organisation de la restitution des travaux dans le mémoire.

1.2. Background

La résolution de problèmes occupe une place prépondérante dans les activités de conception (Bonnardel 2000). Différentes méthodes et outils sont proposés pour formuler et résoudre les problèmes de conception. Parmi les différentes approches, une classification possible est de distinguer les méthodes d'optimisation de celles d'invention. Une fois le problème modélisé, l'optimisation consiste à rechercher des valeurs pour un ensemble fixe de variables, sans remise en cause de ce modèle. Les méthodes d'optimisation ont prouvé leur efficacité dans de nombreuses situations mais ne sont pas conçues pour résoudre les problèmes nécessitant de transformer le système à optimiser en ajoutant de nouvelles variables et de nouvelles relations entre les variables. Dans la thèse, les problèmes qui ne peuvent être résolus par les

méthodes d'optimisation et qui nécessitent des transformations plus profondes des modèles seront considérés comme des problèmes d'invention.

La conception inventive se base sur des principes spécifiques de résolution de problèmes afin de proposer de nouvelles solutions créatives répondant au cahier des charges. Parmi ces méthodes, la TRIZ (acronyme russe pour Théorie de Résolution des Problèmes d'Invention) repose sur l'approche dialectique des problèmes (Altshuller 1988). L'un des axiomes de la TRIZ stipule que l'évolution des systèmes techniques passe par l'identification puis la résolution de contradictions. Ces contradictions émanent de l'apparente impossibilité à satisfaire les besoins d'évolution dans un contexte spécifique ; elles se manifestent par des conflits de valeurs entre les paramètres des systèmes techniques. La TRIZ propose différents niveaux de formulation des contradictions dont seuls deux sont utiles à la résolution du problème : la contradiction technique et la contradiction physique. OTSM-TRIZ (Khomenko et al. 2007), qui a été développé dans le but d'appliquer les axiomes de la TRIZ pour bâtir des méthodes de résolution de problèmes d'invention applicables aux systèmes non techniques, propose en outre un système de contradictions qui lie les deux niveaux précédents de contradiction. La contradiction technique exprime l'opposition entre deux paramètres d'un système, et le fait que l'amélioration de l'un des paramètres entraine la dégradation du second. La contradiction physique définit deux états contradictoires requis pour un seul et même paramètre. Le système de contradictions relie une contradiction physique à deux contradictions techniques qui expriment pourquoi les deux états contradictoires sont requis. Les deux contradictions techniques sont complémentaires et correspondent à l'amélioration d'un premier paramètre qui entraine la dégradation du second, et réciproquement, à l'amélioration du second paramètre qui entraine la dégradation du premier. Dans (Eltzer and De Guio 2007) les deux paramètres impliquées dans les contradictions techniques ont été nommés paramètres d'évaluation, car ils participent à la définition de l'objectif du problème à résoudre, tandis que le paramètre caractérisant la contradiction physique définit un moyen d'agir sur la situation, il est nommé paramètre d'action. Notons que dans l'évolution de la TRIZ la contradiction technique est apparue en premier avec des méthodes de résolution adaptées, puis plus d'une dizaine d'année après est apparu le concept de contradiction physique accompagné

d'une nouvelle famille de méthodes. Pour Altshuller la contradiction physique traduisait une contradiction plus profonde (au sens de la dialectique). L'hypothèse était que derrière toute contradiction technique se cachait une contradiction plus fondamentale, la contradiction physique.

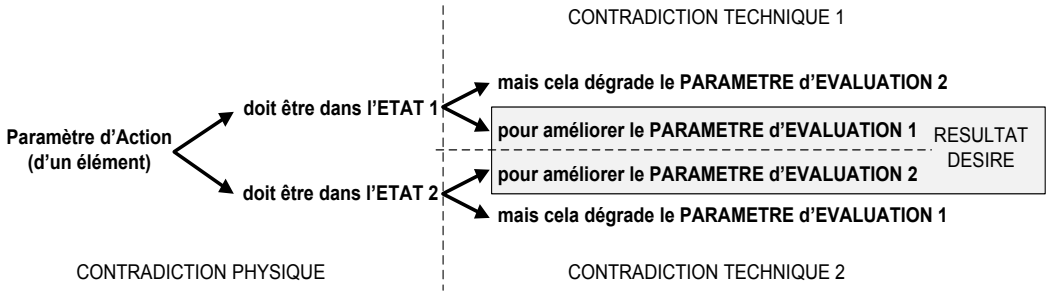


Figure 69: système de contradictions classique de la TRIZ

Nos motivations principales de l'utilisation des approches basées sur la dialectique sont doubles : d'une part, identifier les contradictions afin de mieux comprendre, exprimer les problèmes de la conception inventive, et, d'autre part, d'aborder la résolution de ces problèmes avec des approches de résolution des contradictions. Les contradictions doivent être clairement définies et compréhensibles au début du processus de résolution afin de pouvoir choisir celle(s) permettant de résoudre le problème. En effet, la pratique a montré qu'un mauvais choix de la contradiction peut conduire à diminuer l'efficacité du processus de résolution de problème. D'où le rôle essentiel de l'extraction et l'interprétation des contradictions dans l'approche dialectique. La TRIZ ne résout pas la question du choix pertinent de la contradiction. Notre hypothèse expliquant ce problème pratique était notamment que le concept de contradiction de la TRIZ doit être retravaillé et précisé.

Les travaux passés ont permis de souligner une lacune dans la définition et l'utilisation des contradictions dites « classiques ». Parmi les limites on notera l'inexistence avérée dans certaines situations de contradictions correspondant à la définition donnée par la TRIZ classique dans les relations disponibles entre les variables d'un système, et l'absence de définition explicite du contexte de validité des contradictions dans les méthodes de résolution. Pour combler ces limites la contradiction généralisée a été proposée. La structure de la contradiction généralisée est analogue à celle de la TRIZ classique (Figure 69) en remplaçant les paramètres par

des concepts (Figure 5). Les concepts sont des combinaisons logiques de variable et de valeurs. Le paramètre d'action et ses deux états de la TRIZ classique est remplacé par deux ensembles de paramètres d'action et leurs états associés notés Concept 1 et Concept 2 dans la Figure 5. Les paramètres d'évaluation 1 et 2 de la Figure 69 sont remplacés par des ensembles de paramètres d'évaluation Y1 et Y2 respectivement dans la Figure 5. Ainsi, lorsque les paramètres d'action répondent au Concept 1, l'ensemble des valeurs prises par les paramètres d'évaluation Y1 est satisfaisant alors que l'ensemble des valeurs prises par les paramètres d'évaluation Y2 est insatisfaisant et vice versa. Notons que la contradiction de TRIZ classique est un cas particulier de contradiction généralisée.

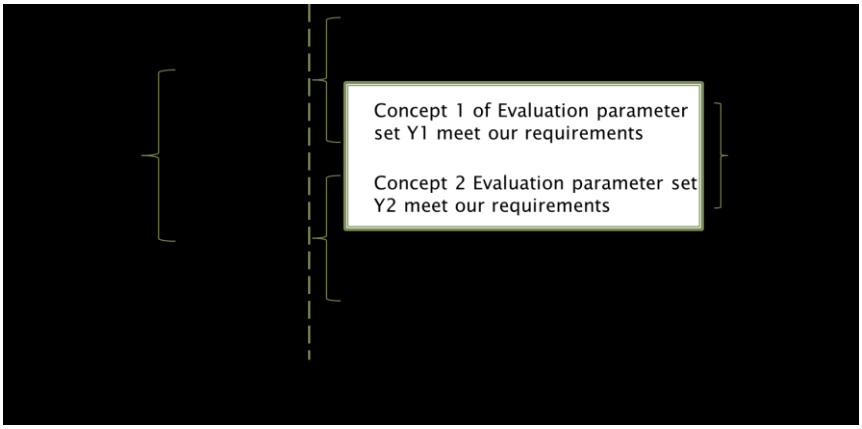


Figure 70: Système de contradictions généralisées

1.3. Problématique

Dans la pratique actuelle le système de contradictions est recherché par interview d'experts. Dans nos travaux de recherche précédents, nous avons démontré qu'il existe des cas où aucune contradiction de la TRIZ classique n'existe et pourtant les problèmes ne peuvent pas être résolus par l'optimisation. De ce fait, le concept des contradictions généralisées a été introduit. Ces contradictions généralisées ne sont pratiquement pas recherchées car leur expression est difficile à préconcevoir pour l'esprit humain. En effet, un expert humain peut assez facilement valider une contradiction généralisée, mais il est beaucoup plus difficile pour lui de la définir. De plus, même lorsqu'on recherche des contradictions physiques ou techniques simples (selon modèle de la TRIZ classique), le praticien humain peut être confronté à ses

propres limites d'expertise lorsque le système est trop complexe ou s'il n'a pas d'expertise sur le système considéré.

Cette thèse contribue à répondre à une problématique se situant à plusieurs niveaux. Les paragraphes suivants précisent ces niveaux et les questions associées.

Le premier niveau de questions de l'ordre de la théorie de la conception concerne le concept de contradiction l'un des fondements de la TRIZ :

Quand aucune contradiction technique ou physique classique n'est disponible les contradictions techniques généralisées et les contradictions physiques généralisées existent elles et sont-elles nombreuses ? Peut-on toujours à partir de la représentation comportementale d'un système et d'objectifs déterminer ses contradictions généralisées de manière intrinsèque (au même titre qu'un Pareto par exemple). Quelles sont les conséquences ?

Dans ce cas, comment les identifier et les extraire de manière exhaustive ?

Une fois connues, comment choisir (définir) celles qui sont pertinentes ? Ou encore, comment définir une contradiction pertinente ?

Le second niveau d'ordre méthodologique concerne les conséquences pratiques d'une nouvelle définition des contradictions pour la résolution des problèmes inventifs :

Une fois les contradictions identifiées et extraites, comment pouvons-nous les utiliser dans la résolution des problèmes inventifs ?

Comment extraire des nombreuses contradictions celles qui sont pertinentes de manière efficace et efficiente sans recherche exhaustive souvent trop couteuse en temps de recherche et de filtrage a postériori ?

Pouvons-nous utiliser le concept de la contradiction généralisée pour exprimer la connaissance implicite d'experts d'un système ?

Enfin, le troisième niveau des questions concerne l'exploration du lien entre les méthodes d'optimisation et la TRIZ pour faire une fertilisation croisée d'un point de vue théorique et/ou pratique a été abordé.

Pouvons-nous utiliser les méthodes et les concepts d'optimisation afin de faciliter l'identification des contradictions généralisées ?

Y-a-t-il un lien entre le concept de Pareto et la contradiction généralisée ?

Si ce lien existe, peut-il être mis à profit pour identifier les contradictions généralisées

Inversement si ce lien existe, sa connaissance peut-elle être mise à profit pour faciliter les processus d'optimisation ?

Le but de cette thèse est répondre partiellement ou entièrement aux questions précédentes. La stratégie de recherche adoptée pour y répondre est décrite dans la section suivante.

1.4. Démarche de recherche

Pour répondre à ces questions il a été décidé de construire dans un premier temps un outil d'extraction exhaustive des contradictions généralisées techniques puis physiques à partir de données. Pour ce faire, les problèmes d'identification des contradictions technique et physiques généralisées ont été modélisés par des problèmes d'optimisation combinatoire, puis un algorithme de résolution a été proposé pour chaque cas. Une fois ces algorithmes réalisés, ils ont pu être utilisés pour traiter les questions (Q1) et (Q2) de notre problématique par des études empiriques ciblées. Ils ont également permis de faire des hypothèses sur les réponses aux questions (Q4) et (Q6). De fait, nous avons répondu positivement à la question (Q7).

Les méthodes exhaustives ont leurs limites : d'une part, elles ne peuvent être mises en œuvre que pour des systèmes au nombre de variables limités (temps de calcul chronophage), et d'autre part, comme le montre les réponses aux questions (Q1) et (Q2) ces contradictions sont extrêmement nombreuses. Cette limite se fait surtout ressentir au niveau des paramètres d'action. Pour reculer la limite liée au nombre de variables, notre stratégie a consisté à analyser les données de sorte à identifier les variables d'action et leur valeurs impliquées dans les concepts des contradictions physiques généralisées sans avoir à déterminer les concepts eux-mêmes, puis d'utiliser l'algorithme exhaustif sur le système réduit aux variables identifiées. La démarche permettant le prétraitement et la réduction des variables d'action en vue de l'identification des contradictions a été réalisée en tentant d'apporter des réponses aux questions (Q8) et (Q9). Une alternative à cette séquence, qui n'est pas développé dans cette thèse, sera la conception d'heuristique de recherche directe de

contradictions pertinentes à partir de données sans recherche exhaustive de celles-ci. La mise au point de ces algorithmes suppose l'identification de propriétés propres aux contradictions « pertinentes ». La recherche de ces propriétés peut se faire par la réalisation de cas avec des experts de conception inventive couplée à l'analyse des résultats des données exhaustives (ici nous prendrons un cas d'école académique du domaine de la logistique). Nous espérons ainsi répondre à la question (Q5).

Dans le cadre de la thèse nous ne pensons pas pouvoir donner une réponse complète à la question (Q4) mais nous pensons pouvoir y contribuer en expérimentant les heuristiques précédentes sur des cas académiques et réels lorsque le nombre de variables s'y prête. La question (Q10) est discutée au vu de la synthèse des résultats aux questions précédentes.

1.5. Organisation du mémoire

Afin de restituer ces travaux et les réponses à la problématique ci-dessus, les chapitres suivant l'introduction du mémoire sont organisés de la manière suivante. Les chapitres 2 et 3 traitent respectivement de l'identification des contradictions techniques et physiques. Ils formulent le problème d'optimisation, proposent un algorithme de résolution qui est ensuite utilisé pour répondre aux questions ci-dessus. Le chapitre 4 aborde le problème de l'identification des variables participant aux contradictions physique, propose une heuristique de base qui est évaluée, propose une des démarches d'identification à partir de cette heuristique puis discute les contributions aux questions de la problématique. Le chapitre 5 propose une manière de combiner les outils développés dans les chapitres précédents au sein d'une méthodologie de résolution de problèmes d'invention. Cette proposition est illustrée à l'aide d'un problème sur un système de Kanban. Le chapitre 6 propose une discussion de synthèse sur les contributions et les limites des résultats des chapitres précédents pris dans leur ensemble. Il propose également des perspectives à la recherche présentée dans le mémoire.

2. Extraction des contradictions techniques généralisées

Le problème d'identification est d'abord décrit et reformulé sous la forme d'un programme d'optimisation en nombre entier (binaire) (Chen et al., 2010) dont la résolution est organisée en sous problèmes. Ce problème combinatoire se révèle NP-

difficile. Les solutions à ce problème sont les contradictions généralisées. Un algorithme dont l'objectif est l'identification des toutes les solutions à ce problème d'identification et d'extraction des contradictions techniques généralisées est proposé. Il utilise comme données d'entrée les données issues de plan d'expériences sur des systèmes physiques ou sur des simulateurs. Sa complexité et ses limites sont présentées et évaluées sur l'exemple d'un disjoncteur électrique.

L'espace de recherche des contradictions techniques généralisées est défini par le nombre des paramètres du système et par le nombre d'expériences impliquées. L'algorithme présenté dans ce chapitre permet donc de répondre à la question (Q2) lorsque le nombre de variables est limité à 15.

Disposant de cet algorithme nous pouvons envisager de répondre également à la question (Q1) pour les contradictions techniques en réalisant des expériences adaptées. De ces expériences, il ressort les éléments suivants :

Plus on dispose de solutions répondant à des objectifs d'évaluation multiples, moins il y a de chance de pouvoir formuler une contradiction technique classique. Lorsqu'on répond à plus de 50% des objectifs il n'y a pratiquement plus de chance de trouver une contradiction technique classique. Ce point explique les difficultés rencontrées par les experts humains pour formuler des contradictions selon le modèle de la TRIZ classique dans certaines situations.

Le nombre de contradictions techniques généralisées semble croitre jusqu'à la disparition des contradictions classiques, il diminue cependant également ensuite au fur et à mesure qu'on augmente le nombre d'objectifs atteints. Il existe toujours au moins une contradiction généralisée dans les données sauf si le problème est totalement résolu (i.e. tous les objectifs sont atteints, il n'y a pas de Pareto).

Les contradictions issues de données de cas réels ne se distinguent pas fondamentalement de celles des données aléatoires ayant des caractéristiques semblables. Il semble donc qu'on puisse utiliser des données générées aléatoirement pour construire et valider des heuristiques d'identification des contradictions techniques. Le problème des données ne se pose pas pour l'algorithme de résolution exacte proposé dans le mémoire, mais il se posera pour des heuristiques qui visent à résoudre le même problème pour des systèmes contenant un grand nombre de variables.

L'application des résultats de ce chapitre permet également de donner des réponses partielles aux questions (Q4) à (Q6). Pour la recherche exhaustive avec l'algorithme proposé nous sommes limités par 15 paramètres d'évaluation.

3. Identification des contradictions physiques généralisées

Pour former un système de contradictions généralisées comme celui de la Figure 5, on peut associer plusieurs contradictions physiques généralisées à une contradiction technique généralisée. Ce chapitre traite de la recherche de toutes les contradictions physiques associées à une contradiction technique généralisée, propose un algorithme, discute ses limites, ses contributions à la problématique de thèse et ses applications potentielles.

La recherche de chaque concept d'une contradiction physique généralisée est d'abord définie comme la recherche de concepts discriminants sur des ensembles définis par la contradiction technique généralisée. En effet, une contradiction technique généralisée partitionne l'ensemble l'espace des actions E en trois ensembles : l'ensemble de l'espace d'action qui satisfait les paramètres d'évaluation Y1 que nous notons E1, celui qui satisfait les paramètres Y2 que nous dénommons E2 et le troisième qui complète E1 et E2 pour former E que nous notons E0. Rechercher un $concept_1$ revient à rechercher une fonction qui discrimine qui reconnait des éléments de E1 et les discrimine de E2UE0. Inversement, rechercher un $concept_2$ revient à rechercher une fonction qui reconnait des éléments de E2 et les discrimine de E1UE0. Le problème de recherche est définit sous la forme d'un problème de programmation binaire en nombres entiers dont on utilise les propriétés pour le résoudre. Ces contradictions peuvent être très nombreuses et impossibles à lister dans un temps raisonnable. Pour pallier ce problème nous montrons qu'une contradiction physique généralisée spécifique peut résumer toutes les contradictions physiques généralisées possibles. La recherche de cette contradiction spécifique peut être faite en ajoutant une fonction objective à l'ensemble des équations à satisfaire dans le programme binaire, ce qui revient à résoudre le problème d'optimisation entier binaire.

Dans ce chapitre, l'algorithme est illustré à l'aide d'exemples concrets notamment le cas du disjoncteur électrique qui montre le nombre très important des systèmes de contradictions généralisées qui peut être trouvé.

La complexité de l'algorithme est définie et les limites pratiques d'utilisations de l'algorithme sont données : par exemple le nombre des paramètres d'action du système à traiter est limité (les cas d'étude menés montrent que l'extraction exhaustive fonctionne si moins de 12 paramètres d'action du système avec au maximum deux niveaux de valeurs à traiter sur les ordinateurs à notre disposition).

L'utilisation de l'algorithme a également permis d'apporter une réponse partielle à la question (Q6) en comparant les contradictions formulées par des experts indépendants et celles définies par les algorithmes précédents.

Une exploitation possible de l'algorithme et inattendue au départ de l'étude est la possibilité d'identifier des contradictions physiques classiques ou généralisées dans un contexte de réglage des paramètres d'action. Ceci permet d'envisager de multiples applications dans le cadre de la conception inventive et de l'aide à la compréhension des contradictions. Ainsi par exemple, on peut comprendre ou compléter des contradictions émises par des experts en définissant leur limite de validité, ou inversement proposer aux concepteurs des contradictions simples se rapprochant de celles de la TRIZ classique en y ajoutant les limites de validité.

La recherche des contradictions sur un échantillon permet parfois d'identifier des solutions au problème d'optimisation. En effet, il se peut que sur l'ensemble d'expériences, les concepts 1 et 2 trouvés séparément ne soient pas contradictoires. Dans ce cas, la combinaison de ces concepts est une solution potentielle au problème initial qui n'est peut-être pas un problème d'invention.

4. Identification des paramètres participant aux contradictions physiques

Comme précisé au chapitre 2, l'utilisation des algorithmes de recherche exhaustive des contradictions généralisées a ses limites quant au nombre de paramètres pouvant être traités avec un ordinateur. Pour l'algorithme de recherche des contradictions techniques généralisées nous ne pouvons traiter que 15 paramètres d'évaluation et dans le cas de l'algorithme de recherche des contradictions physiques

généralisées nous ne pouvons traiter que 12 paramètres d'action avec 2 valeurs chacun. En revanche, le traitement des résultats obtenus avec ces algorithmes dans les cas réels, nous amène vers l'hypothèse que dans la plupart des cas, seulement quelques paramètres d'actions sont impliqués dans les contradictions physiques généralisées. Une solution serait donc de ne pas considérer tous les paramètres d'action et toutes leurs valeurs pour définir et identifier les contradictions. L'objectif de ce chapitre est de proposer des heuristiques permettant de sélectionner des paramètres pertinents et leurs valeurs pertinentes pour la recherche des contradictions généralisées. Pour cela l'identification des paramètres permettant de classer les éléments dans deux ensembles distincts est abordée sous l'angle des méthodes d'apprentissages en intelligence artificielle. Nous commençons par montrer que l'identification des paramètres et de ses valeurs entrant dans les concepts des contradictions physiques généralisées s'apparente à une série de problèmes connus sous le vocable de « feature selection problem» dans certains domaines de recherche (Liu & Motoda, 1998). Il s'agit de trouver l'ensemble optimal des caractéristiques d'un ensemble de données afin d'améliorer la précision d'un classifieur, d'accélérer les algorithmes d'exploration des données et de fournir une meilleure compréhension de ses résultats. Dans la première partie de ce chapitre, l'état de l'art et le choix d'algorithmes de feature selection adapté à notre objectif sont présentés ; notre objectif étant de chercher l'ensemble des paramètres d'action qui ont un effet positif significatif sur les paramètres d'évaluation. Parmi les méthodes de « feature selection » celles de « signed feature selection » semblent les plus adaptées à notre cas. Deux méthodes ont été implémentées et testées : Pearson Correlation Coefficient et Support Vector Machine (SVM) ; les meilleurs résultats sont obtenus avec la SVM que nous avons adoptée et dont nous présentons l'application à notre situation dans ce chapitre. La méthode SVM transforme un problème non-linéaire en un problème linéaire c'est-à-dire fournit une classification linéaire dans l'espace de recherche des caractéristiques non-linéaire en fonction des poids de ces caractéristiques. Plus la caractéristique (le paramètre d'action) a du poids dans la méthode SVM plus il a de l'influence sur l'ensemble étudié (le paramètre d'évaluation et sa valeur objective). Un autre avantage de la méthode SVM est que

nous pouvons considérer chaque paramètre d'évaluation comme résultant des combinaisons des paramètres d'action.

Deux démarches à base de l'utilisation répétée d'algorithmes de type SVM sont proposées. La première s'appuie sur les familles d'expériences définies par une contradiction technique généralisée ; la seconde s'appuie sur les familles définies par les performances atteintes ou non de chaque variable d'évaluation.

Pour évaluer l'algorithme central conçu à base de SVM, un ensemble de cas dont on connait la réponse (le concept à identifier) est construit. Les résultats de l'application de l'algorithme à cette base de cas sont fournis et discutés.

L'exploitation des résultats obtenus nous amène également à réfléchir à l'exploration du lien entre la démarche d'identification des paramètres et les méthodes d'optimisation, notamment sur la recherche des variables clés pour trouver le Pareto.

5. Processus de résolution de problème d'invention

Le chapitre commence par positionner les activités de résolution de problème ciblées dans la boite « model change » du processus de conception résumé ci-dessous.

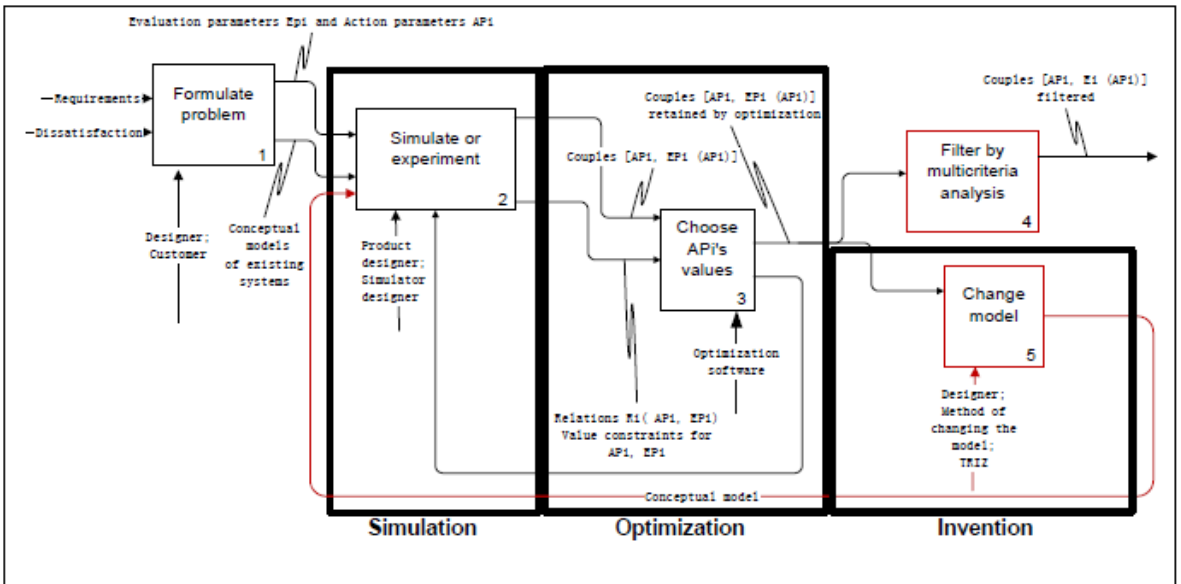


Figure 53: General framework for inventive design problem solving

La figure ci-dessous résume la démarche proposée à partir de données obtenues par l'expérimentation physique ou par la simulation.

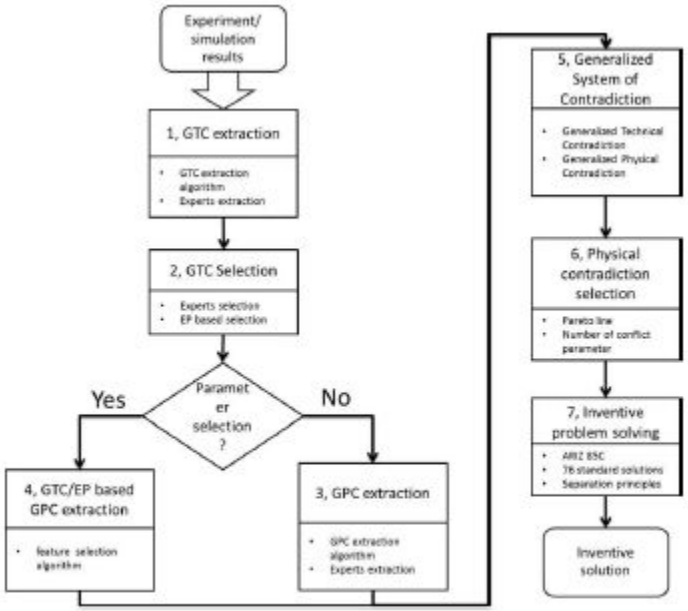


Figure 54: Process of model change

Puis nous avons illustré le processus de la résolution des problèmes inventifs sur un exemple de système logistique à double Kanban. Ce processus intègre l'extraction des contradictions techniques généralisées (chapitre 2), l'extraction des contradictions physiques généralisées (chapitre 3), l'identification des paramètres participants aux contradictions physiques et finalement la résolution basée sur les principes de TRIZ de la contradiction choisie. L'exemple consiste d'une machine fabriquant deux types de produits et approvisionnant deux stocks différents. A travers l'expérimentation sur ce cas d'étude nous avons illustré l'utilisation des différents algorithmes proposés dans la thèse pour pouvoir répondre aux questions posées dans la problématique de notre travail de recherche et auxquelles nous avons répondu dans le chapitre 6.

6. Discussion conclusive et perspectives

Notre démarche de recherche nous a emmené à progressivement mettre en place des outils/algorithmes/expérimentations pour pouvoir répondre aux questions classées à différents niveaux de notre problématique de résolution de problème dans la conception inventive. Afin de répondre au premier niveau de questions concernant le concept de la contradiction dans la TRIZ nous avons proposé des algorithmes exhaustifs d'extraction des contradictions généralisées basés sur les problèmes

d'optimisation combinatoire (comment extraire les contradictions généralisées de manière exhaustive Q2) et nous avons mené des études empiriques. Les résultats montrent que si le cas où aucune contradiction technique ou physique classique n'est disponible est assez commun et croit avec le nombre des solutions disponibles, au moins une contradiction généralisée existe toujours, et, dans la majorité des cas, celles-ci sont extrêmement nombreuses (réponse à la question (Q1)). Dans ce cas, la question comment choisir/définir les contradictions pertinentes se pose (Q3). Nous essayons d'y répondre en exploitant et utilisant les liens entre les méthodes et les concepts d'optimisation et la TRIZ, tel que le concept de Pareto. En même temps le fait de générer beaucoup de résultats limite les méthodes exhaustives d'extraction des contradictions en termes de nombre de paramètres pouvant être pris en compte dans les algorithmes. Pour faire face aux limites des méthodes exhaustives, notre stratégie a consisté à mettre en place des heuristiques qui visent à résoudre le même problème pour des systèmes contenant un grand nombre de variables. Une exploitation possible de l'algorithme est la possibilité d'identifier des contradictions physiques classiques ou généralisées dans un contexte de réglage des paramètres d'action. Ceci permet d'envisager de multiples applications dans le cadre de la conception inventive et de l'aide à la compréhension des contradictions. Ainsi par exemple, on peut comprendre ou compléter des contradictions émises par des experts en définissant leur limite de validité, ou inversement, proposer aux concepteurs des contradictions simples se rapprochant de celles de la TRIZ classique en y ajoutant les limites de validité.

Suite à la mise en place successive des algorithmes et des heuristiques nous avons été emmenés à traiter les questions concernant les conséquences pratiques des nouvelles contradictions pour la résolution des problèmes inventifs (Q4). L'utilisation de l'algorithme a également permis d'apporter une réponse partielle à la question (Q6) en comparant les contradictions formulées par des experts indépendants et celles définies par les algorithmes précédents. Au fur et à mesure de nos travaux de recherche nous sommes de plus en plus amenés à réfléchir sur la manière de combiner les algorithmes et les heuristiques précédents dans une démarche générale d'extraction des contradictions généralisées pertinentes de manière efficace et efficiente sans recherche exhaustive (Q5).

Il s'agit d'analyser les données de sorte à identifier les variables d'action et leur valeurs impliquées dans les concepts des contradictions physiques généralisées sans avoir à déterminer les concepts eux-mêmes, puis d'utiliser l'algorithme exhaustif sur le système réduit aux variables identifiées. La démarche permettant le prétraitement et la réduction des variables d'action en vue de l'identification des contradictions pertinentes est réalisée en explorant le lien entre les méthodes d'optimisation et la TRIZ ce qui permet d'apporter des réponses aux questions (Q7), (Q8) et (Q9). L'identification des variables et valeurs impliquées dans les contradictions permet d'apporter une meilleure compréhension voire l'identification des éléments (variables) clés pour chercher le Pareto (Q10).

Le chapitre se termine par une discussion prospective.

Lei LIN

Optimization methods for inventive design

Résumé

La thèse traite des problèmes d'invention où les solutions des méthodes d'optimisation ne satisfont pas aux objectifs des problèmes à résoudre. Les problèmes ainsi définis exploitent, pour leur résolution, un modèle de problème étendant le modèle de la TRIZ classique sous une forme canonique appelée «système de contradictions généralisées ». Cette recherche instrumente un processus de résolution basé sur la boucle simulation-optimisation-invention permettant d'utiliser à la fois des méthodes d'optimisation et d'invention. Plus précisément, elle modélise l'extraction des contractions généralisées à partir de données de simulation sous forme de problèmes d'optimisation combinatoire et propose des algorithmes donnant toutes les solutions à ces problèmes. De plus, elle propose des heuristiques pour sélectionner les variables et les valeurs pertinentes intervenant dans les contradictions généralisées et/ou utiles pour l'optimisation. Les contributions concernent la théorie et la pratique de la conception inventive. La thèse explore également la fertilisation croisée entre l'optimisation et la TRIZ.

Mots clés : TRIZ, Conception inventive, Optimisation, Feature selection.

Abstract

The thesis deals with problems of invention where solutions optimization methods do not meet the objectives of problems to solve. The problems previously defined exploit for their resolution, a problem extending the model of classical TRIZ in a canonical form called "generalized system of contradictions." This research draws up a resolution process based on the loop simulation-optimization-invention using both solving methods of optimization and invention. More precisely, it models the extraction of generalized contractions from simulation data as combinatorial optimization problems and offers algorithms that offer all the solutions to these problems. In addition, it provides heuristics to select variables and its relevant values involved in generalized contradictions and/or useful for optimization. The contributions concern theory and practice of the inventive design. The thesis also explores cross-fertilization between optimization and TRIZ.

Keywords: TRIZ, inventive design, optimization, feature selection.