POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Constraint Programming as a Technology to Support Project Management

TESI DI LAUREA MAGISTRALE IN
MANAGEMENT ENGINEERING
INGEGNERIA GESTIONALE

Author: **Zeynep Beste Guran**

Student ID: 945860
Advisor: Prof. Mauro Mancini
Co-advisor: Prof.ssa Costanza Mariani
Academic Year: 2021-2022

Abstract

Nowadays, the companies have been facing with severe challenges such as the scarcity of resources, limited profit margins, unstable socio-economic status-co. In such a harsh business condition, making good decisions and solving problems effectively are vital for the survival of operations. To do so, businesses have to benefit from the breakthrough technologies in their processes to gain a significant advantage. In this manner, Artificial Intelligence tools applied to decision support and problem solving, provide exciting opportunities to the managers. In this thesis, a noteworthy practice of AI which is constraint programming (CP) has been examined as a potential support mechanism for project management (PM) practices such as resource allocation, scheduling and planning. This study aims to provide a clear positioning of CP technology in the managerial decision making and combinatorial problem-solving context among other operation research like mathematical programming and AI techniques such as machine learning. In this manner, an extensive literature review has been conducted and a practical case has been implemented to verify CP is a promising method in the PM area especially for finding feasible and optimal solutions. The case consists of designing NP-hard resource constrained project scheduling problems with varying conditions and solving them by using different solution techniques; critical path method, Gantt chart, CP, linear programming and mixed integer programming. Then, the performances of these solution techniques are examined based on the constructed framework which intends to introduce a comprehensive matrix for evaluation and selection depending upon quantitative (computational time, memory usage) and qualitative (ease of modeling, closeness to natural language, availability of complementary materials) metrics. As a result of the case and the findings from the literature show that CP is a valuable approach with its theoretical properties, conceptual simplicity, and practical success to assist rational managerial decision making in the PM domain.

Key-words: Constraint Programming, Project Management, Combinatorial Problem Solving, Scheduling, Decision-Making

Abstract in italiano

Al giorno d'oggi, le aziende stanno affrontando sfide severe come la scarsità di risorse, i margini di profitto limitati, lo stato socioeconomico instabile. In una condizione di business così dura, prendere buone decisioni e risolvere i problemi in modo efficace è vitale per la sopravvivenza delle operazioni. Per fare ciò, le imprese devono beneficiare delle tecnologie di svolta nei loro processi per ottenere un vantaggio significativo. In questo modo, gli strumenti di Intelligenza Artificiale applicati al supporto decisionale e alla risoluzione dei problemi, forniscono opportunità interessanti ai manager. In questa tesi, una pratica degna di nota dell'AI che è la programmazione vincolata (CP) è stata esaminata come un potenziale meccanismo di supporto per le pratiche di gestione dei progetti (PM) come l'allocazione delle risorse, la programmazione e la pianificazione. Questo studio mira a fornire un chiaro posizionamento della tecnologia CP nel processo decisionale manageriale e nel contesto della risoluzione di problemi combinatori tra altre ricerche operative come la programmazione matematica e le tecniche AI come l'apprendimento automatico. In questo modo, è stata condotta un'ampia revisione della letteratura ed è stato implementato un caso pratico per verificare che il CP è un metodo promettente nell'area PM specialmente per trovare soluzioni fattibili e ottimali. Il caso consiste nel progettare problemi di programmazione di progetti con vincoli di risorse NP-hard con condizioni variabili e risolverli usando diverse tecniche di soluzione; metodo del percorso critico, diagramma di Gantt, CP, programmazione lineare e programmazione intera mista. Poi, le prestazioni di queste tecniche di soluzione sono esaminate sulla base del quadro costruito che intende introdurre una matrice completa per la valutazione e la selezione in base a metriche quantitative (tempo di calcolo, uso della memoria) e qualitative (facilità di modellazione, vicinanza al linguaggio naturale, disponibilità di materiali complementari). Il risultato del caso e i risultati della letteratura mostrano che il CP è un valido approccio con le sue proprietà teoriche, la semplicità concettuale e il successo pratico per assistere il processo decisionale manageriale razionale nel dominio del PM.

Parole chiave: Programmazione dei Vincoli, Gestione dei Progetti, Risoluzione dei Problemi Combinatori, Pianificazione, Processo Decisionale

Executive Summary

Artificial intelligence (AI) is becoming a cornerstone of most of the modern technological systems. More than ever, AI-enhanced technologies are being widely used in the industries and mentioned in academia. The rising use of vast amounts of processing power and data enable the organizations to benefit from AI technologies. However, throughout the evolution of AI, few people understand the opportunities and drawbacks of AI paradigms. Some people think that AI will replace all the operations, processes, and systems in the world radically and their perception is that just self-driving cars, robots, automatic chatbots can be considered as AI advancements. However, the reality is that AI has been around as a technological discipline for more than 60 years in order to interpret events, solve problems, support and automate decisions, and take actions. Therefore, at this point, it is important to understand how AI can be utilized and which needs can be met with these technologies rather than going with the flow.

As a purpose of being, organizations exist to solve problems by a product or a service. During this problem-solving process, they search for feasible or optimal alternatives with satisfying several constraints such as time, budget, assets, specifications, laws. However, mostly, the complexity of the problem exceeds the ability of problem solving of a human being considering the uncertainties, limited resources, interdependent relations between variables and objectives. Indeed, one of the most important disciplines for organizations to cope with these challenges is to excel in project management (PM). Therefore, an intelligent support mechanism is essential to make reliable PM decisions in the pursuit of success.

AI-based systems such as machine learning and deep learning are already used today as a tool for supporting decision making for complex problems. However, despite their increasing efficiency, its explainability in terms of understandability, transparency, and interpretability is an issue especially in the PM area which are not experts in computer science, naturally. Besides ex-post handicaps, there are some critical prerequisites for a successful AI implementation which requires not only time and financial resources but also massive amounts of data, computational processing power, skilled employees and so on. Although there are several obstacles to using AI to make grounded decisions, the expectations from a project manager are expanding and require a knowledge of using technological tools and systems for efficient project management.

In particular, constraint programming (CP) is a technique born from the artificial intelligence domain, initially conceived to solve decision problems where a feasible solution is needed for a set of variables which satisfy a defined set of constraints. CP consists of the modeling of a problem as a constraint satisfaction problem (CSP) and solving the CSP with constraints propagation, search algorithms and heuristics. However, in practice, the user is just responsible for modelling because several available CP software such as Google OR, IBM ILOG CPLEX, Gecode find the solution for the model which is the formulation of the problem. Several industrial applications utilized from CP are available in various domains such as manufacturing, transportation, health care, telecommunications, financial services. Considering the thesis topic, some successful practical examples of CP technology related with PM have been represented for scheduling, resource allocation, planning and product development which search for optimality or feasibility. This thesis mainly aims to introduce CP technology with its theoretical properties, conceptual simplicity and practical success as a decision-making support mechanism in a project domain which always has a nature including constraints and trade-offs for the problems.

With the goal defined, the literature findings indicated that there are many applications of operations research (OR) that have been utilized to tackle with PM problems such as scheduling, resource allocation, planning, routing and portfolio selection. The approaches can be classified into two sections as exact and approximate models. The former finds the solution which is proven to be an optimal or exact solution in case that the input problem may look for a feasible solution rather than optimal. Considering the emphasis in this paper is on exact methods, mathematical programming, goal programming, constraint programming and network models can be given as examples for deterministic approaches. The latter uses heuristics and meta-heuristics algorithms to provide a quicker route to approximate solutions because in some cases the complexity of the problem can cause an unreasonable computational time and memory usage to solve it. Although constraint programming is an exact method, it has an interest in algorithms and heuristics as well to improve performance of solving.

Many interconnections with related techniques of OR and AI have been studied along the years and hybrid methods have received a lot of attention in the last years resulting in improved performance in many combinatorial problems. AI and OR approaches have complementary strengths like AI in domain-specific knowledge representation and OR in efficient mathematical computation. Among these several alternatives, CP which possesses characteristics from both AI and OR, though it is AI based, seems like a great possible fit for dealing with real life PM decision making processes since it is a combination of century old mathematical methods along with various state-of-the-art techniques such as programming languages, automatic solvers. In order to validate this hypothesis, a case study analysis is combined with an extensive literature survey.

In this manner, one of the major analyses performed is related to the comparison of the solution methods and procedures applied to diverse scheduling problems similar to real life cases. Not only due to the difficulty of solving a scheduling problem, but also the reliability of the solution is an issue such that most commonly used software packages such as Microsoft Project, Primavera give different solutions for scheduling. Moreover, considering that scheduling is one of the most successful application areas of constraint programming as well as one of the most challenging responsibilities of a project manager, the main problem of the case has been selected as resource constrained project scheduling problem (RCPSP). It aims to shorten the project duration and assign the resources to the activities while keeping the budget minimum considering the duration reductions of the activities require an expense. To be able to contrast the methods, the case has been separated into 3 parts which are scheduling, scheduling with crashing and scheduling with crashing under resource constraints as a combinatorial NP-hard problem. The reason is that linear programming (LP), and network models are not suitable for combinatorial optimization, and it is better to evaluate the comparison between CP and them in the part - 2 which is scheduling with crashing without resource constraints. Then, to solve the main combinatorial RCPSP, which is part - 3, the CP and MIP models have been developed and compared. IBM ILOG CPLEX Studio has been used with different solvers to evaluate the models and solving techniques. The comparisons of methods have been evaluated using a scoring framework. The framework shows an overall picture of performance of LP, CP and MIP techniques from the viewpoint of the quantitative metrics, time and memory usage, as well as the qualitative metrics, ease of modeling, closeness to natural language and availability of complementary materials.

The problem solved for 105 instances for 3 project sizes, 5, 25, 50 activities. The instance conditions have been selected from the situation which as if there is no restriction, which can be found with critical path method (CPM), to the unfeasible situation which means it is impossible to find a schedule that satisfies given input data time and resources. So, it has been possible to develop a decision assistance system for project crashing problems under resource constraints. The alternatives have been proposed with different target time to completion of a project and worker and employee resources. Then, the crash amounts from activities and optimal cost values have been compared. The aim is to provide a project manager a support to create and select the optimal scheduling program and budget considering the possible capacity resources and project completion time. In this manner, the proposed CP flowchart for decision making and problem solving has been applied. CP utilized decision flowchart enables the user to have more alternatives on the way of dealing with any given scenario that might be faced like an what-if analysis. In such a way that it empowers to see how the changes on the variables and constraints would affect the outcome of a situation.

While doing so, it has been found that CP outperforms LP and MIP technique according to the designed performance framework. First of all, LP does not support discrete optimization problems due to its linear nature and so, it has not been taken into the main focus of the discussions although it has a remarkable superiority for time and memory usage. In comparison between MIP and CP, the latter consumes less memory than the former because CP has greater flexibility to express the constraints of the problem. Moreover, the use of special constraints can replace large number of traditional integer programming constraints and variables which lead to much less memory requirement and more compact formulation. Secondly, there are more intricated issues in terms of time. Even though CP has a time superiority for finding better solutions in limited time (cutoff) and proving infeasibility, the number of instances solved with CP is lower than the ones with MIP. So, it is not correct to make a generalization about their time performance or declare one is better than the other. The reason is that there is not a stable dominance of a model over the other. Moreover, the models are quite sensitive to not only the problem size and method but also the parameters of the instances, algorithms and heuristics selected by the default settings of the software program used.

In terms of qualitative metrics, the natural formulation of constraint programming is not far from the real life and to problem description than the mathematical programming such as MIP and LP formulations. It is not only related with expressiveness of the models, but also it has a high-level declarative language for modelling. It can be acceptable as one of the strengths of CP because it is like a spoken language. Moreover, CP's syntax and keywords such as *cumulative, size, endOf* are more familiar with natural language which are already existing words in our daily language. Moreover, when the case has been taken into consideration in two parts, it has been seen that CP has a flexible and extensible nature since adding a new problem dependent constraint is possible without having to modify or change the solution strategy. Unlike, a technique change has been required from LP to MIP in the case of presence of resource constraints. CP can work with non-linear, arithmetic, higher-order or logical constraints as well as supports the global (a.k.a. special) constraints for expressing some substructures such as cumulative, all different, sequence etc. In other words, CP formulation provides built-in constraints and functions for specific purposes such as expressing precedence relations and presence status which are commonly used while dealing with resource allocation, sequencing, scheduling problems. In addition, in fact, it is not surprising that a project manager would prefer to not write complex codes considering his/her educational background, expertise area and competences. With a basic modeling and programming knowledge, it is possible to automatically solve project problems in CP. In view of the fact that CP packages allow the use of different powerful concepts depending on the requirement. In other words, the user spends the main effort to describe the problem in constraint satisfaction or optimization problem, not solving it because the CP engine selects the proper search algorithms and heuristics are automatically from MP, CP, metaheuristics etc. Last but not least, both LP and MIP solvers do not provide any additional complementary materials apart from just displaying numerical solutions. On the other hand, CP is offering some helpful visualization graphics such as Gantt charts, resource diagrams, etc. and managers can highly benefit from these while making complicated decisions where there is not an obvious one and only solution which is almost always the situation in real life.

In such a way, it is possible to change parameters, evaluate the results and compare the alternatives to select the most fruitful one for the project success by a project manager. Therefore, through the case study, it has been verified that CP can be promising for project management problems especially for scheduling. CP provides a modern combinatorial optimization and constraint satisfaction technology that can be an alternative and complementary for machine learning, CPM/PERT based PM software packages and mathematical programming.

Contents

Introduction

In today's competitive marketplace, material and labor costs are getting higher and product life cycles are becoming shorter. Therefore, organizations and companies apply the most efficient production and management strategies to survive. In such a condition which has besides project's uncertainties and complexities, the role of project management is critical for the success of the project and corporation, in the big picture.

Constraint programming is such a powerful and easy-to-use technology that solves problems and supports decision making when modeled properly. CP is a topic that is developed much more recently by the Computer Science (CS) and Artificial Intelligence (AI) community. Moreover, many problems in CS and AI can be considered as Constraint Satisfaction Problems (CPS). Besides, many combinatorial project problems can be formulated as CSPs considering project constraints and being in search of a conflict-free solution to the problems of the project. Some examples can be given as vehicle routing, resource-constraint scheduling, automated planning, multiple machine scheduling, production optimization, etc.

The integrated approach of Artificial Intelligence and Operations Research has been greatly beneficial for the solutions of large and complex problems. At that stage, CP has been seen as a part of and is gained from OR so that it provides useful applications to the management of projects and organizations. Project managers must have a comprehensive knowledge about the examples and applications of CP because when to select which tool for particular situations is an important managerial and engineering skill.

The thesis focuses on how constraint programming can support project management decision making processes in a limited time and resource environment. To achieve this goal, it explains the relationship of CP with the mentioned fields, AI and OR, which are used to solve project problems considering time and resource constraints. Positioning of the interdisciplinary field and the similar techniques from these fields are presented such as machine learning, integer programming to define the appropriate usage of the problem-solving tools in a project. Indeed, in the main scope of this thesis, some existing implementations from literature for planning, scheduling and optimization as well as a practical case study which is about resource constraint project crashing are presented, analyzed and discussed to justify the usefulness of CP to help managerial decisions.

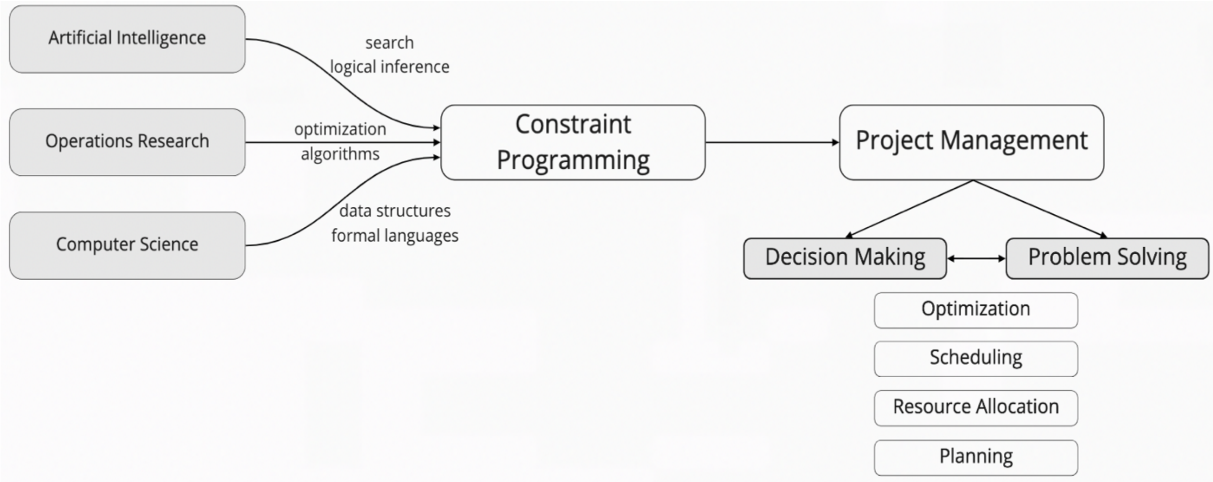


Figure 1: A framework to represent the scope of research topics

In order to grasp the overview of the thesis, the Fig. 1 is designed which demonstrates the roots of the central topic of the thesis, CP, and its commonly applicable and promising areas in the project management area. The thesis scope is solving problems of scheduling, allocations and assignments to select the best efficient alternatives.

Thesis Structure

The thesis starts with an introduction. The scope, purpose and framework of the research topic has been explained. The motivation of the research has been briefly mentioned. Also, it encloses the methodologies applied to answer the research questions by the author.

The following chapter presents Literature Review. It provides the necessary knowledge to develop the research. As a main purpose, it looks for gathering the existing research on the two distinct topics of CP and PM. While doing this, the origins of the topics, the domains which have similar purposes and relationship between them, the basic concepts of constraint programming, and real-life applications have been explained. The visual representation of the chapters and sections are represented in Fig. 2.

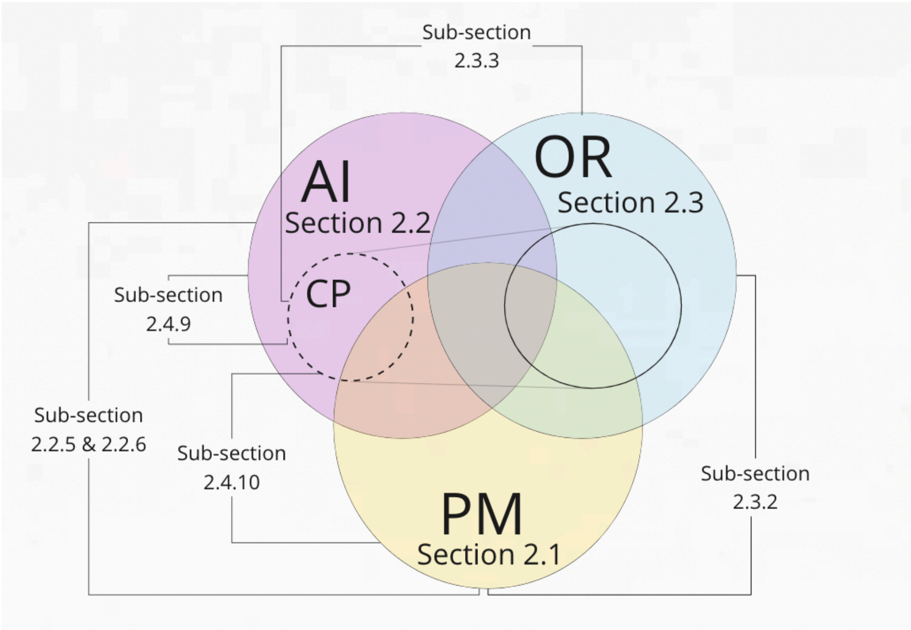


Figure 2: The related fields of CP and the sections that explain the relationships

Firstly, the section mentions project management. It creates a base for the understanding of the processes and phases of project management. It has been highlighted that the projects have several constraints, and the role of project managers is required to solve problems considering the constraints. On this basis, the reader has been prepared for the potential utilization of constraint programming in the area of PM.

Before deep diving to our findings, constraint programming which is based primarily on artificial intelligence and operations research has been explained. Therefore, the other main topic artificial intelligence has been investigated after from the project management section. Therefore, not only an overview of AI including its early history, definition, effects but also its practices and impacts on PM, the challenges to apply AI to project context have been presented in the Section 1.2.

Methods from the Operations Research literature are extensively used in CP approaches. Moreover, the applicability of both areas can be common in terms of the purpose of finding feasible or optimal solutions to the problems. As a result, a literature review of some of the techniques provided by the Operations Research community to handle these problems have been reviewed in section 1.3. Therefore, it consists of OR tools and techniques, the relationship between OR and PM as well as OR and CP.

In section 1.4., the main focus of the thesis CP is examined extensively. The basic knowledge, characteristics, working mechanism have been explained. Moreover, the elements of constraint satisfaction problems, which are how the problems version to be solved with CP are called and solving these problems have been described. In order to witness its real-world applications, some examples have been evaluated from the industry. Also, the models have been presented to demonstrate its usage within a part of a problem solving and decision-making system. Considering research questions., the relationships of CP between AI and PM has also been clarified in this section.

In Chapter 2, Methodology, the "how" of the thesis has been explained starting from the main headings of the research to the conclusion part. The chapter explains how the topic has been narrowed, the information has been gathered, how literature has been reviewed and how these have been analyzed. The objective of the research and associated research questions have been represented which are stemmed from identified literature gaps and possible promising points. Moreover, the practical case study has been exhibited in this chapter with the systems of methods applied to compare mathematical programming techniques and constraint programming in order to verify the thesis objective and illustrate its practice in a decision-making flowchart.

The Chapter 3 presents findings which summarize the study and mention the outcomes. The gathered insights from literature and the practical problem have been indicated. The comments about the findings and answers to the research questions have been presented. Especially, the parallelism of the outcomes of the case and extant knowledge have been discussed.

The last chapter presents a conclusion including the suggestions for CP practitioners as project managers, future improvements on the research area. The limitations and promising ways of the research work has been shared with the reader. Moreover, while concluding the research, novel insights have been conveyed.

1 Literature Review

The overall objective of the chapter is to provide a comprehensive overview of the academic research regarding Project Management and its practices, Operation Research, Artificial Intelligence and their common concept of Constraint Programming. The relationships and the integrations of PM and CP is presented with academic publications, existing models and industrial applications.

The literature review is vital to provide possible project management improvements with the support of CP as a technological enabler for real-time problem solving and decision making. Therefore, the leading outcome of the overview is a draft of the thesis about the current status of CP and PM, as well as recognition of research gaps that are critical to establish the research questions and aim of the work.

1.1. Project Management

Over the recent decades, project management has become a central and critical activity in most organizations. According to the 5th edition of the PMBOK Guide which includes PMI's standards and characteristics of project management, it is the application of knowledge, skills, tools, and techniques to project activities to meet the required project success criteria. The main aim of the project management elements planning, scheduling and controlling, is minimizing expenses by optimizing the goals, performance and time in the context of a specific work and the key to achieve this it is critical to utilize resources productively [12].

1.1.1. Project

Although people use the "project" term a lot in their daily language, a project has typical characteristics that can be differentiated from routine work or business operations. According to Project Management Institute (PMI) which is an international and professional not-for-profit organization, a project is defined as a temporary endeavor undertaken to create a unique product, service, or result [8].

Temporary endeavor indicates that each project has a clear beginning and ending date – despite, in reality, it might not be met due to some problems such as improper planning, scope changes, ineffective communication between stakeholders etc. The duration of a project can be minutes or years such as fixing a closet or construction of Duomo di Milano, respectively. In other words, it does not matter how long the duration of a project is, but it must end. A beginning date refers whenever the first task is scheduled to start. Like, the end date can be accepted as when the project deliverables have been accepted or the project has been prematurely terminated [9]. Moreover, all project documentation has been delivered and archived, and all closing activities such as releasing resources, project retrospective have been finalized. In spite of the temporary nature of a project, the effects can last more than the life of the project. The produced deliverables of the project can create social, economic, and environmental impacts, which can be positively or adversely [10].

The PMBOK Guide defines a deliverable as: "Any unique and verifiable product, result, or capability to perform a service that must be produced to complete a process, phase or project." [8, pp.84] In other words, projects exist to fulfill objectives with producing deliverables such as a product or service that has not done before in the same way. As an example, even producing the same products in different companies does not make the projects the same, because both projects have different shareholders, risks, constraints etc. Therefore, it makes every project unique.

1.1.2. Project Constraints

Like the old idiom "Life is the trade-off", managing a project requires lots of compromises to satisfy stakeholders and meet requirements within boundaries. The iron triangle model which shows three interconnected project constraints that are time, cost and scope helps project managers to respond according to situation and to keep the balance of the project with limited resources. The Fig. 3 iron triangle model represents the constraints which can be pre-defined or elaborated during the project life cycle.

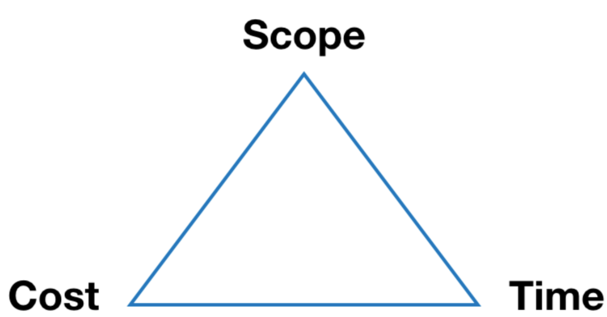


Figure 3: The Triple Constraint model as presented by Van Wyngaard

Scope: Successful completion of a project requires to fulfil the project's deliverables which comes with features and functions, that are determined in the scope.

Cost: The cost refers to the project budget that is a financial constraint to spend for the project including labor costs, material costs, contingency funds, and all other expenses.

Time: It is the overall duration which the project team needs to complete all the tasks and deliver the result of a project.

For instance, if the project is behind the targeted schedule, some mitigation actions come to the stage to recover the delay. As a project manager, you can acquire more people to do the scheduled work that you have not done yet, however it increases the actual labor cost of the project. As another alternative, you can downgrade the scope - however, as a return, it can cause some consequences such as reputational losses, contractual penalties etc. In order word, if you change one constraint, the other interdependent restriction could be affected. Therefore, the possible options and their risks and impacts to other aspects of the project must be evaluated.

Understanding the constraints and boundaries of a project is a vital responsibility for a project manager because no project exists with an unlimited budget and due date. Beyond the basic project triangle constraints, further various constraints can limit project progression. Both the ISO 12500:2013 [9] and a guide to PMBOK 2013 [8] accept quality, resources, and risks as the ones of the competing project constraints. According to context, technology unavailability, regulations or organizational structure can limit the project performance, too. Moreover, some dynamic or unexpected constraints can appear in later stages of the project such as logistics or supply chain problems related to the current pandemic. In pursuit of project success, after knowing the limits and possible shortages, proper management strategies must be applied. However, the high number of constraints, which are also related with each other, and those dynamic nature due to the high level of uncertainties can becloud to select correct strategies. Furthermore, large financial scales, long lifespans, high number of stakeholders, which are also shown as dimensions of megaprojects, increase the complexities of a project [12]. In such an environment, it is very risky and hard to compute the best scenario among several variables, domains, and constraints.

There are lots of parameters such as activities, their precedence, allocations, constraints and so on that affect the project and in return, the project’s problems such as scheduling, resource allocations, routing. In such a problem size, the complexity of the problem exceeds the human ability of problem solving. For example, crashing an activity network could be deficient and hard to apply to large and complex projects. The calculation time and effort increase steeply which means the solving of this problem requires beyond the capacity of a human. Therefore, an intelligent mechanism such as the ones with constraint programming which seeks solutions for the variables that satisfy all the existing constraints in a reasonable effort can play a central role for a project.

1.1.3. The History of PM and its Tools

The foundations of the project management, especially the version that is known today, were laid in between 1900s and 1950s. As a primary modern PM tool, Henry Gantt invented a visual scheduling diagram which is called the Gantt chart in 1917. Starting from these years, organizations began to apply some PM tools and techniques such as CPM, PERT and Work Breakdown Structure (WBS). With the rising interest and need, PMI and International Project Management Association (IPMA) were established in 1965 and 1969, respectively. Since the 1975's, the growth of PM software businesses has started to develop such as Oracle, Artemis and Capability Maturity Software. In the following years, Earn Value Management (EVM) and Projects in Controlled Environment (PRINCE) methods are derived. Also, with the creation of Scrum management style and Six Sigma methodology in 1986, agile models that encourage strong collaborations between team members and customer have developed that Agile methodology take its roots from [70]. In millennium, digital and Internet enriched advancements have developed SaaS and cloud-based technologies for PM such as Asana, Basecamp, Wrike etc. In today's competitive and changing environment, the characteristics, trends and requirements of the PM have evolved. As one of the leading professional services firms, PwC shows project's management history as Fig. 4 by 2018 presentation in a PMDay [65].

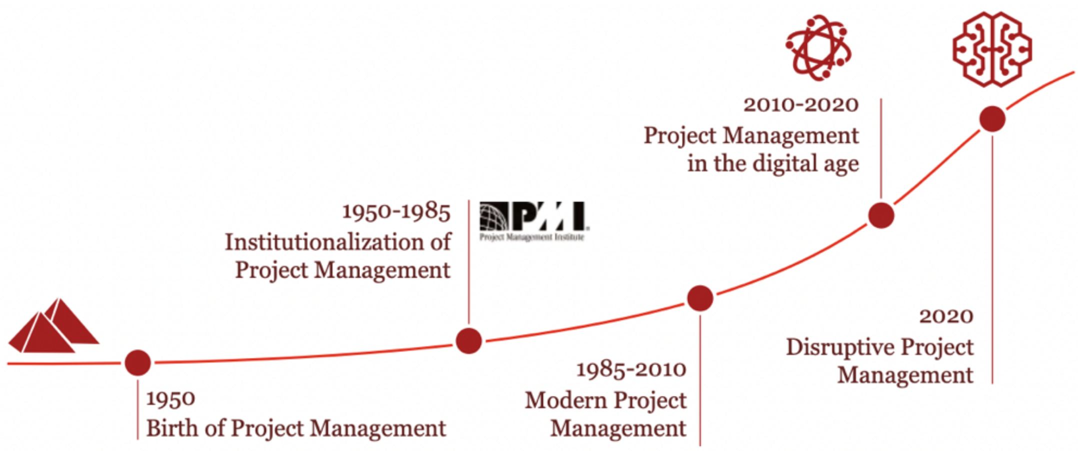


Figure 4: Evolution of project management history

At this point, an explanation can be added that the disruptive technology can be a tool, resource or a finished product or service [71]. In a PMI survey, the technologies are ranked according to the total impact as cloud solutions, IoT and AI. Therefore, AI applications must be known by project managers in these days.

1.1.4. Project Management Knowledge Areas

According to the ISO 21500:2012 Standards [9], a project can be handled with different approaches depending on some specific characteristics such as final goals, risk, size, end time, capabilities of the team, available resources, past knowledge and also industry for the particular case. However, the scope of the PM is framed with PM knowledge areas which consists of the various processes and activities to identify, describe, integrate and coordinate. Organizations must be capable of having all the knowledge areas together into an integrated project management methodology.

These are the 10 Project Management knowledge areas:

a) Project Integration Management

Project Integration Management is the procedure of specifying, categorizing, putting together combinations and organizing different project and process management tasks in the scope of Project Management Process Groups. Integration is a critical success factor in order to achieve an effective project completion where the business value is maximized while customer requirements are satisfied. The processes can be listed as:

- Develop Project Charter
- Develop Project Management Plan
- Direct and Manage Project Work
- Monitor and Control Project Work
- Perform Integrated Change Control
- Close Project or Phase

In addition, integrative actions are not easy to implement as there needs to be different criteria satisfied which asks for some sort of a trade-off between resources and project outcomes. In theory, the tasks of project management are a discrete set of activities with definitive boundaries, but it is not usually true in real life cases which demand controlling, running and completing interconnected jobs together [8, pp. 63-103].

b) Project Scope Management

The work done to meet objectives and deliver an output such as product, service or result with the defined functions and features are main concerns for project scope management. It helps to ensure the project consists of all the validated work, not less and not much. The overview of the project scope management is the following:

- Plan Scope Management
- Collect Requirements
- Define Scope
- Create Work Breakdown Structure
- Validate Scope
- Control Scope

The defined scope based on scope plan and project requirements should be validated and controlled throughout the project. Also, the overall scope can be separated into manageable components with WBS [8, pp. 105-139].

c) Project Time Management

A project consists of several activities which consume time and resources. In schedule management, the required resources are tied to activities, precedence relationships between activities are identified. With these, critical activities and paths are determined. The related documents such as schedules and Gantt charts are revised and updated on a regular basis. Time management is one of the main aspects of effective management due to its practical importance in a project. The following list outlines the processes:

- Plan Schedule Management
- Define Activities
- Sequence Activities
- Estimate Activity Resources
- Estimate Activity Durations
- Develop Schedule
- Control Schedule

A schedule is a listing of a project's milestones, activities, and deliverables, usually with intended start and finish dates. Moreover, it represents the precedence relationships between activities and sequence of work items. For some projects, a detailed scheduling must be prepared according to activities that have to be finalized in an order while abiding to the starting and finishing dates set beforehand.

Poor scheduling could have a negative influence on the project environment, resulting in increased costs, profit losses, reputational damage and losing current and possible contracts. According to McKinsey Productivity Sciences Center, almost all megaprojects suffer cost overruns of more than 30%. The average cost increase is 80% of the original value while the average slippage is twenty months behind the original schedule [73]. In such cases, it is vital for project managers to decide wisely among constraints of project management that are scope, time and cost [8, pp. 141-190].

Project Schedule Management Problem Types

For the scope of this thesis, it is important to expand the scheduling aspect of project management further by explaining different types of problem settings since in the upcoming sections the analysis will be concentrated on these. Various scheduling problems have been trying to be handled since the middle of the 20th century with the developments of several techniques which will be discussed later.

- Scheduling

In the context of the simplest version of the scheduling problems which forms a base for the other variations, there are some project tasks and without any additional constraints the only requirement is obeying the precedence relationships between these tasks and the respective durations thus, the expected solution set is the total time needed to complete the project. However, this is of course not the case in real life and assumes almost unlimited resources so that it calls for a demand to more complex models.

- Scheduling with Project Crashing

The scheduling with project crashing type of problems is also called as time/cost trade off problems in books and articles which introduces the possibility of reducing the durations of some of the project activities by outsourcing or allocating more internal resources etc. on top of the standard scheduling problems. The real-life context of problems referred to as scheduling with project crashing is when there is a strict deadline of the project, and it is practical to cut off the time needed to finish single tasks with a unit cost depending on the overall conditions. The main findings of the solutions of the scheduling with project crashing is the cut off rates from each activity which optimizes the total spending for crashing.

- Scheduling with Project Crashing under Resource Constraint

It is mandatory to analyze scheduling with projects crashing under resource constraints in order to get a more realistic model since in the real business context, project tasks have both time and resource requirements that each task asks for a certain amount of work force or other resources in order to successfully finalize it. So, basically the resources of a company devoted to a single project as well as the project deadline determines how many hours/days should be reduced from each task by paying extra [103].

At this point, it should be mentioned that how to deal with these scheduling problems will be provided in the further sections with the solution techniques after introducing constraint programming and operation research methods. Also, some real-life business scheduling cases are way more sophisticated and dynamic than the problem models mentioned above so, there are also other aspects that should be deemed to increase the degree of accuracy of the solution. With this aim, other types of constraints exist such as cash flows, optional activities combined with the ones above, but these are not the primary interests of this thesis.

d) Project Cost Management

Project cost management is mainly concerned with the costs of the resources required to fulfill all project operations. It's also beneficial to ensure that the project is completed within the planned and agreed budget. This knowledge area is divided into four processes:

- Plan Cost Management
- Estimate Costs
- Determine Budget
- Control Costs

The project cost management governs also the directions and decisions taken in a project such as hiring extra employees, using and maintaining a product/service which cause an expense. Even, in case of not using a monetary resource at that moment does not guarantee a cost saving because it can lead to another potential cash out indirectly. For example, skipping or limiting a number of maintenance sessions of a machine can give rise to a malfunction or less qualified outputs of the machine. Therefore, the cost problems and trade-offs should be decided carefully [8, pp. 193-225].

e) Project Quality Management

Project Quality Management for a business consists of the procedures and tasks that organize the quality infrastructure by defining standards and conformity assessments with related actors liable for a project scope in order to meet the customer requirements or design specifications. Project Quality Management is not only limited with a single project context but also the quality policies are also beneficial for more general terms such as organizational and process transformation so that obtaining efficiency and effectiveness. The related processes include:

- Plan Quality Management
- Perform Quality Assurance
- Control Quality

In simple terms, firms and corporations implement quality management with an ultimate goal of assuring the outcomes of the projects to be aligned with the requirements [8, pp. 227-252].

f) Project Human Resource Management

Project Human Resource Management refers to the methods which deal with the best practices for utilizing a project team in order to achieve the objectives by managing the human aspect of the necessary actions. Project teams are basically people who have different competences as well as availabilities and emotional characteristics from various backgrounds and the most important job of Project Human Resource Management is to assign these staff to the project roles and tasks in the best possible way for a successful completion. The overview of the processes is as follows:

- Plan Human Resource Management
- Acquire Project Team
- Develop Project Team
- Manage Project Team

The management of a project's team members can lead to a significant competitive advantage in the short run since this has no exact math but rather calls for experience. In addition, it is also worth mentioning that for a prospering Project Human Resource Management, involvement of all the people to the initial stages of a project, especially planning in order to benefit from distinct strengths that the individuals can offer [8, pp. 255-284].

g) Project Communications Management

Project communication management basically deals with developing and applying strategies for communication and conflict management. Project Communication Management processes consist of:

- Plan Communications Management
- Manage Communications
- Control Communications

Project managers spend almost all of their time interacting with team members and project stakeholders in several ways such as official, unofficial, formal, informal, written, oral etc. While they are doing this, they need effective communication the most because effective communication draws a connection between various stakeholders who may be different from every aspect and helps to minimize the negative effects of these varieties and conflicts on the project [8, pp. 287-307].

h) Project Risk Management

Every project has different kinds of risks arising from uncertainty. In the context of projects, risk is an uncertain event or condition that, if it occurs, has a positive or negative effect on a project's ability to meet its performance, cost, schedule, financial, or other objectives. Positive risks, which are opportunities, can be exploited and increased by benefit sharing and upselling actions. Negative risks, which are threats, must be mitigated or avoided with proper measures. The positive and negative risks are connected to each other as well, in a way that each threat can turn into an opportunity if the right actions are taken, and each opportunity can turn into a threat if it is not exploited or mishandled. Therefore, risk management holds crucial importance for the projects. In 5th edition of the PMBOK Guide, the risk management processes are summed up as [8, pp. 307-353].:

- Plan Risk Management
- Identify Risks
- Perform Qualitative Risk Analysis
- Perform Quantitative Risk Analysis
- Plan Risk Responses

- Control Risks

i) Project Procurement Management

An organization can need to get products, services or results from external parties. Obtaining these needs from outside of the project team with purchasing or recruitment is called procurement. It requires planning, selecting, coordinating and maintaining these outsourced materials and services. The knowledge area includes contract management, risk management, change control processes such as controlling contractual obligations and evaluating make-or-decisions analysis with risks. Project procurement management encompasses the processes:

- Plan Procurement Management
- Conduct Procurements
- Control Procurements
- Close Procurements

Project management is essential for supply chain processes to complete the project in a timely and efficient manner [8, pp. 355-389].

j) Project Stakeholder Management

Project stakeholders are the people, groups, or organizations that could impact or be impacted by the project. Stakeholder management includes analyzing and answering their expectations, needs and impact on the project with proper management strategies. Also, the process consists of:

- Identify Stakeholders
- Plan Stakeholder Management
- Manage Stakeholder Engagement
- Control Stakeholder Engagement

A stakeholder management strategy is vital to achieve our project success. Project managers can focus more on the important stakeholders and avoid less influential ones while considering the roles and balances can vary with time. Project managers should correctly identify them and effectively enhance engagement with stakeholders [8, pp. 391-413].

1.1.5. Project Process Groups

As per PMBOK Guide, the project management processes are divided into five groups —initiating, planning, executing, monitoring and controlling, and closing, which can be followed on Fig. 5 and defined as:

Initiating: A new phase or new project development processes are carried out to initiate to the project or to the phase of an existing project.

Planning: The scope, time, budget, resources, risks, objectives, strategies, course of actions and such parameters of a project or phase are defined with the planning processes.

Executing: The processes to fulfil the defined work and meet the specifications of the project. This process group consists of integrating and carrying out project activities, coordinating people and managing customers.

Monitoring and Controlling: The process group includes tracking, reviewing, monitoring and controlling of projects in terms of performance and progress. If needed, change requirements and related actions are identified and initiated within this continuous process group.

Closing: The finalization process which could be for a project closure or phases are performed to finish all activities across all Process Groups [8, pp. 47-60].

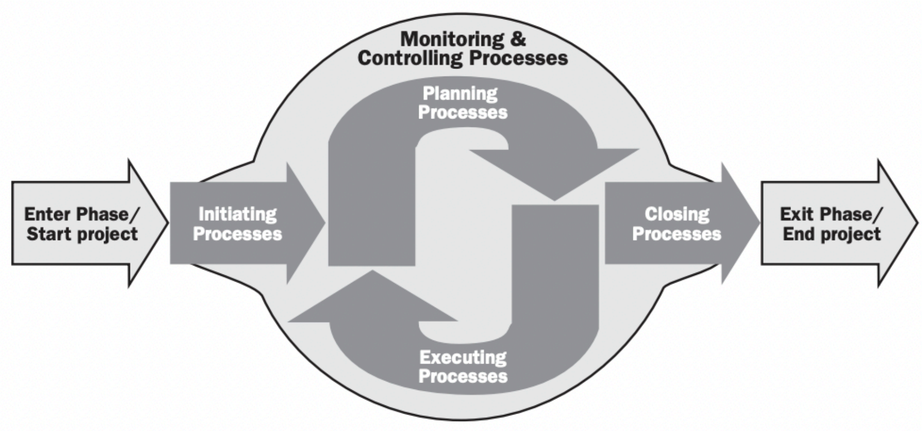


Figure 5: Project Management Process Groups

The categories are grouped considering the processes' integration, interaction and aims. The process groups guide the managers to apply proper PM knowledge and skills, which are independent of application domain or industry focus. In order to avoid confusion at that point, a brief explanation about the project phase is given because they are not the same. Also, in a similar manner, a project can be divided into phases to allow ease of management of the project. There is no standard structure to separate the projects into phases however it is characterized by the completion of one or more of the project's deliverables such as concept development, feasibility study, design, prototype, build, or test, etc.,. Therefore, the phases can differ depending on the nature of a particular project such as its size, complexity, potential impact, and the style of a project team.

1.1.6. Importance of Project Management

Project management is accomplished through the proper conduct of the mentioned knowledge areas during the project life cycle which is a defined set of phases from the beginning to the end of the project. PMI Success in Disruptive Times 2018 report highlights the importance of the role and responsibilities of project manager. The number of organizations that embrace the potential benefits of project management approaches are just a little more than half of the total. Therefore, the underperformance of the projects is inevitable. The following statistics which are collected by PMI validates the results of the poor management [55].

Figure 6: Some of the consequences due to poor management

Besides the quantitative data, absence of an effective project management can experience several dissatisfactions in individual and organization level. On the other hand, when effective project management practices are employed, organizations can get successful outcomes. Some of them are the followings.

- Satisfying the expectations of stakeholders
- Resolving the issues and problems
- Meeting business objectives and intents
- Delivering right products/services at right time with original budget
- Optimizing the usage of resources

- Balancing constraints such as scope, cost, quality, schedule and managing the effects
- Responding the risk and opportunities on time
- Gaining competitive reputation
- Working in a peaceful team and environment

1.1.7. The Competences of a Project Manager

A skilled project manager is vital for project success. Managing a project requires a set of competences including general management, project management and industry skills. The competencies can be categorized in different dimensions. Marando (2012) separated them into soft and hard skills [91]. The latter is defined as technical aspects which include tangible deliverables such as schedules, project charters, risk management reports, critical path analysis, budgets. These deliverables can be expressed with dashboards, spreadsheets, templates, or hard data. On the other hand, soft skills are generally intangible and related to interpersonal skills such as problem solving, decision making, creativity and so forth. Project managers should find a balance of hard and soft skills.

The Future of Jobs Report (2016) of World Economic Forum [88] consistently highlights the importance of complex problem solving and decision making for the required skillsets to flourish in the Fourth Industrial Revolution which brings to the organizations advanced autonomous systems, AI, ML improvements. The Table 1. illustrates the skillset which is prepared according to the leading employers such as chief human resources and strategy officers all around the world across industries and geographies.

At this point, the separation between hard and soft skills can be misleading. It should be highlighted that those are used together in many contexts complementarily. For example, soft skills like negotiating, communicating, adapting and critical thinking are really important to possess in order to successfully implement hard skills like scheduling, budgeting and risk management to a business. This is especially true for the project management aspect of a business that any specific hard tasks of PM could only be done by combining at least some of the soft skills mentioned above. The prerequisite for a successful project environment is a concurrent team which various business functions from different hierarchies in a company operate together by effective communication and usage of leadership skills.

Table 1: Top skillset in years 2020 & 2015

Top skills in 2020	Top skills in 2015
Complex problem solving	Complex problem solving
Creativity	Coordinating with others
People management	People management
Coordinating with others	Critical Thinking
Emotional intelligence	Negotiation
Judgement and decision making	Quality Control
Service orientation	Service orientation
Negotiation	Judgement and decision making
Cognitive flexibility	Creativity

Decision Making and Problem Solving: All the projects tend to encounter problems and selections with multiple objectives and constraints. Even Burke and Barron [93] have seen a project as a big problem which requires to be solved. The solutions that are produced for the project problems is strongly linked with the project success. Likely, Mckinsey's The Art of Project Leadership Report [94] indicates that there is a strong relation between the success of a large project and timely and good quality decisions made. The project team must ensure that processes and procedures are in the position for identifying the problem as soon as possible and assisting problem solving.

There is no exact accepted definition for these terms because it is natural that a problem can be taken into consideration in different dimensions, in other word, it can be divided into sub-problems or taken as a whole issue. Moreover, problem solving and decision making are embedded within each other. Therefore, the processes can be illustrated in various versions. For example, Burke and Barron showed the relationship as Fig. 7 on Project Management Leadership book. The interrelated process starts with problem solving phase to produce practical and technical solutions in divergent manner. The steps consist of defining objectives, recognizing problems, gathering information and identifying options. The gathering information and data step is used to frame and understand the problem better. In this step, data mapping, sense making, and constraint identification are carried out to shape objectives and problem. Then, options, possibilities and (range of) solutions are generated with problem solving techniques. The next step is a convergent decision making. In the wider perspective of problem-solving, decision-making is a selection among feasible solutions for the problem.

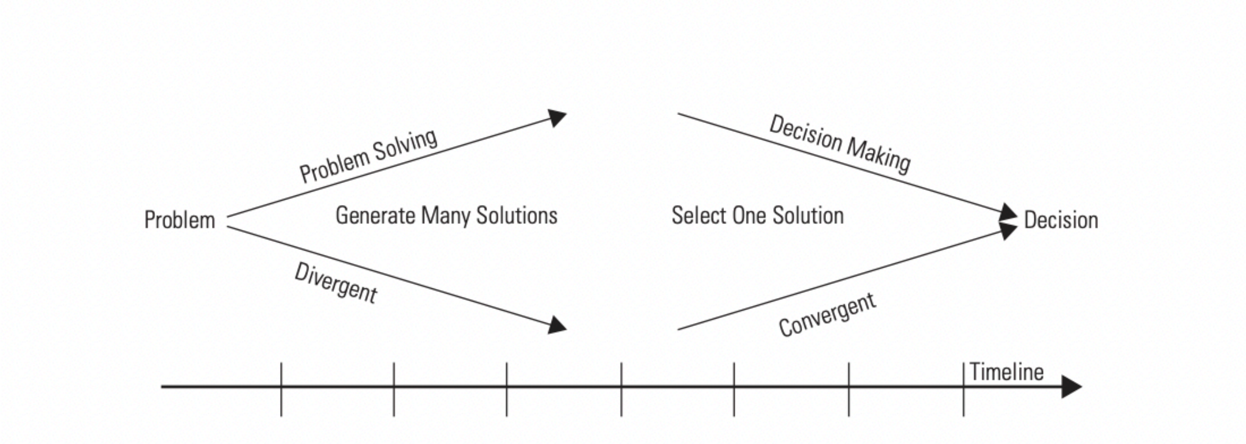


Figure 7: The problem solving and decision-making process flow

Decision making can be defined as the procedure of choosing a particular solution to address a problem among various alternative action sets. The main steps and procedures of decision making and problem solving can be demonstrated linearly in the Fig. 8, as the following schema [95].

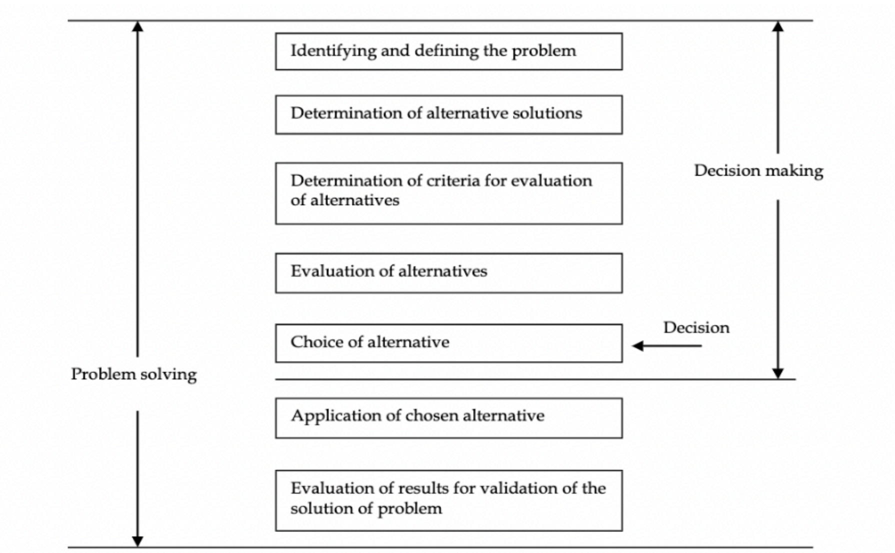


Figure 8: Relationship between problem solving and decision making

The problem can be related with any aspect of the project in which there is no one answer instead there are different and sometimes conflicting criteria to be fulfilled and distinctive success factors in other words, objectives considering stakeholder requirements, expectations, baseline plan. In this manner, not all of the criteria are in same importance and should have weighted importance scores assigned to them which is critical in order to eliminate some possible solutions that are not well suited for the decision goal. However, it must be noted that in some cases, the problem only has one or no feasible solution in the case of over-constraints problems and no selection or weighting is needed.

1.1.8. Tools and Software in PM

The project management tools and software are widely used to assist project managers to complete the project effectively and efficiently. While organizing and managing the project, these tools help to guarantee the delivery of project outputs within quality, time and budget constraints. Patanakul et al. (2010) relate PM knowledge areas and PM tools and techniques as in the Table 2 [101].

Table 2: PM Knowledge Areas and their corresponding tools and techniques

PM Knowledge Areas	PM Tools and Techniques
Integration management	Project selection, return on investment, payback period, project charter
Scope management	WBS, scope statement, quality function deployment, change request, scope change control, product review, performance measurement, lesson learned
Cost Management	Cost estimating techniques, earned value management, cost change control system, performance measurement
Quality management	Benefit/cost analysis, flowcharting, cause-and-effect diagram, cost of quality, Pareto diagram, control charts, trend analysis, quality audits, benchmarking, statistical sampling
Time management	CPM, PERT, GERT, Gantt charts, simulation, Monte Carlo analysis, buffer management, schedule crashing, milestone charts, variance analysis
Risk management	Risk matrix, Monte Carlo analysis, decision tree analysis, checklist, SWOT analysis, and Delphi, project risk audit, earned value management

Human resource management	Stakeholder analysis, responsibility matrix, team building activities, reward and recognition systems, organization charts, project team directory
Communications management	Stakeholder analysis, earned value management, information retrieval systems
Procurement management	Make-or-buy analysis, contract type selection, statement of work, contract change control system, source selection, bidder

The larger and more complex projects are likely to push the need of a software which incorporates the underlying PM tools and techniques within. According to findings of an extensive survey conducted by PwC, which has one of the largest professional service networks, in 2016, it is shown that almost all of the mature companies use project management software to automate and support their PM processes [98]. Moreover, there is a positive correlation between the usage of a project management software and high performance of the project.

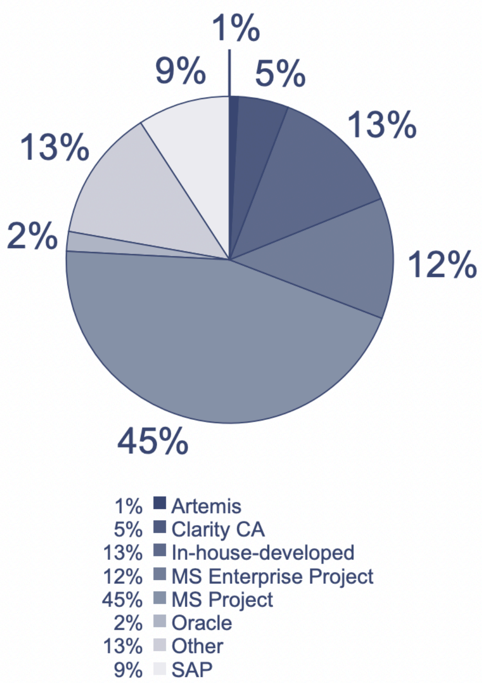


Figure 9: Types of software used by managers

The Fig. 9 represents the types of project management software used by managers by percentage. In the pie chart, it can be seen that Microsoft Project is the most widely used by far among others. In general, the software package allows the user to make scheduling, resource allocation and planning with detailed reports, graphs and tables such as cost breakdowns for activities, expenditure forecasts, Gantt charts and so on. The features are including but limited to progress updating, controlling and reporting. At this stage, it is also worth to mention Oracle's Primavera Project Planner which is a database-based software that is used for similar purposes although its contribution is not highly remarkable in the above chart because the selected software packages are strongly linked with the industry groups. By way of explanation, older sectors with a well-built PM ground such as engineering and construction tend to prefer high-end and comprehensive packages such as Primavera unlike simpler and elementary packages like MS project, in-house developed programs [99].

Depending on firms' work environment, size, industry, needs and their project complexity and size, companies can use several PM software packages. With rising cloud and software technologies, there are lots of commercial PM software in the market such as Clarizen, JIRA, Zoho project, Gemini and so on. Moreover, there are also widely used applications for risk management, workflow management, team collaboration, etc. However, these do not take a broad place in literature. Also, they are not mature and popular as the above-mentioned ones and so, for the sake of these and the thesis topic, those are not mentioned.

1.2. Artificial Intelligence

Although there is no generally accepted definition of AI, one characterization for artificial intelligence is that the capacity of a model to execute intelligent system goals derived from the information set provided as input. So, an AI machine has the characteristics as a smart intermediary taking required actions collecting data from surroundings. It can be described as everything done by machines, especially computers, that act or/and think like a human such as learning, reasoning, solving problems, optimization [20], [21].

Even in ancient times, people attempted to create objects which exhibit intelligent behaviors [14]. The fascination about making lifelike creatures from these old days has continued for a long time. Not only mathematicians and computer scientists, but also psychologists, neuroscientists, physicists, and economists concerned with the Artificial Intelligence topics. In order to have a better understanding about AI, it is vital to grasp its history and perceive how much this field evolved.

1.2.1. A Brief Early History

After the end of global war World War II, research and developments gained a lot of importance especially in the area of computer science, which can be tracked with Fig. 10 [15]. An earlier milestone in artificial intelligence created by Alan Turing who has been described as the father of the modern computer. Turing's paper "Computing Machinery and Intelligence" published in 1950 introduced the concept of Turing test which assesses whether a machine has intelligence or not [16]. With this attempt, the artificial intelligence concept is introduced. Moreover, he examined machines' ability to solve problems, make decisions and learn like human-being. However, the discipline of artificial intelligence did not formally emerge until a conference was hosted by John McCarthy and Marvin Minsky in 1956. After a couple of years later, the great duo developed one of the oldest and favored programming language LISP and it was operated for researches on AI [25].

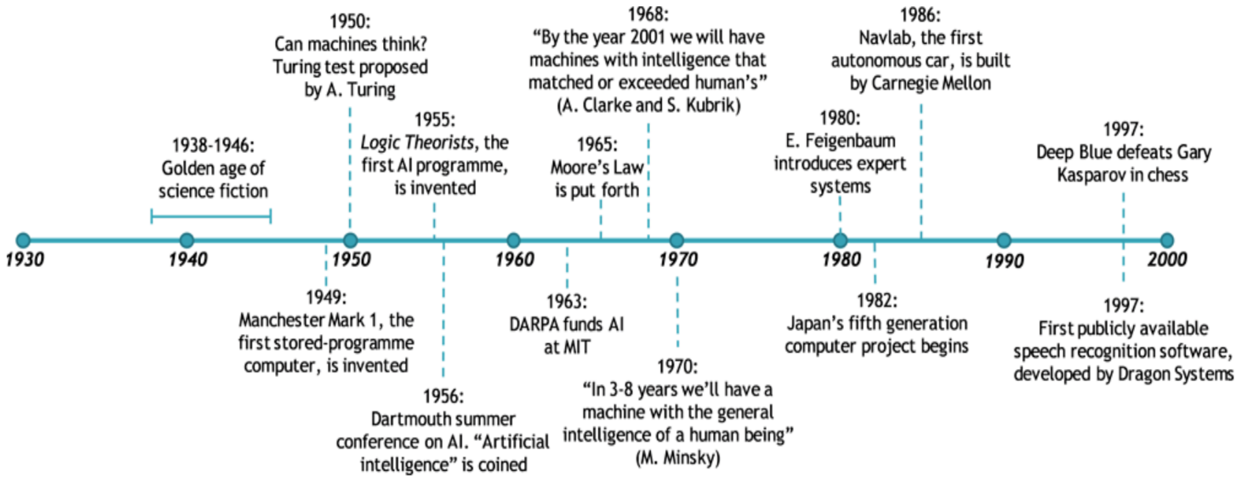


Figure 10: Timeline of early AI developments (1950s to 2000)

The idea of smart machines also created an attention on the industrial applications, the first industrial robot was invented in the 1950s and used on the General Motors assembly line in 1961. A few years later, pioneering chatbot ELIZA was created as an early natural language processing computer program. The interactive computer program was able to limitedly speak English with the user. As another milestone, in 1997, International Business Machines Corporation's (IBM) Deep Blue chess playing program defeated Garry Kasparov who is a one of the former World Chess Champion. Another great step forward was achieved by the emotionally intelligent robot Kismet which was able to recognize and display gestures and mimic movements in communication.

Despite today's AI and its success, it has gone through difficult times which are called AI winters. In these periods, the funding was reduced and the interests in research diminished. AI winters are caused by computers' inadequate computational power in terms of computer storage and processing speed. Last but not least, lack of commercial results and over expectations created disappointments about the topic. However, AI has overcome the difficulties and thrived [17], [18].

Since the 1980s there have been implementations of AI to real life cases for solving complex management problems in various domains. It is used to decrease human labor and increase productivity to save time and money. With successful commercially expert systems and available tools, AI gained momentum within the industrial applications. For example, the expert learning system XCON that structures customer orders and assists the assembly process of these orders brought remarkable financial benefits to Digital Equipment Corporation. Thanks to the configuration system, they saved $40 million annually from 1980 to 1986 [19]. The early work paved the way for applying AI to industries which boost work processes and human abilities.

1.2.2. AI Classifications

According to previous definitions done in several textbooks, Russell and Norvig [21] categorized AI along two fundamental dimensions.

Systems that think like humans	Systems that think rationally
Systems that act like humans	Systems that act rationally

The left-hand side concerns the human-likeness of AI while the right-hand side focuses on rationality that requires to reach the best possible outcome among the alternatives. At that point, in order to avoid misunderstanding, the difference must be highlighted. Humans are also rational creatures however human beings can make some systematic errors and show irrationalities that could be caused due to emotions, biases etc. For example, although people who know all the laws and rules of chess could be beaten because of their improper moves. Moreover, the table separates the ones on the top and the bottom as thought processes and behaviors, respectively.
The first quadrant tries to understand and imitate how a person thinks that is needed to develop a theory of mind and implement it to the system. This view coincides with some related disciplines. For instance, the cognitive scientists contribute the research about the topic by examining the topic on level of mind, brain, neurons. The systems that carry thinking human-like characteristics are able to solve problems like humans do.

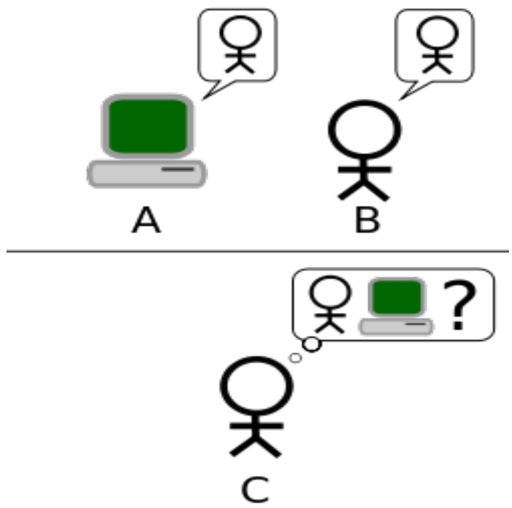


Figure 11: Graphical representation of Turing Test

The other aspect to AI is strictly related to the Turing test approach that expects systems acting like a human. This test tries to decide whether a machine is intelligent or not. As shown in the Fig. 11, it consists of an interrogator person that is represented as C and two answering bodies, a machine(A) and a person(B). The human questioner C asks the same question to the bodies and gets the answers by written messages. However, the interrogator does not know which answer is taken by whom. If the interrogator(C) cannot determine that he/she is communicating with an alive human or robot, the computer is accepted as it is acting intelligently. In order to be classified as a system that acts like a person, some capabilities are expected such as understanding the language, reasoning and some physical actions like moving the bodies etc.

According to the laws of thought approach, correct reasoning must be captured, and it must be based on an irrefutable logic process that requires making conclusions from facts. Therefore, as a result of thinking rationally, the right answer is reached. With the initial studies of Aristotle, the field of logic started in the notion of "right thinking". Consequently, the AI approach originated from the area of logic, in other words the programs that could solve logically notified problems inspired that AI could build systems which think rationally in the same manner.

In the other approach that is on the right bottom quadrant, AI is defined as systems that are able to act rationally and focus on intelligent agent's design to act so as to achieve best outcomes. It can be reached based on the level of knowledge and reasoning ability. Therefore, this approach includes the laws of thought perspective because correct inference is just one of the mechanisms that can be used to perform rational actions. Between rationality and human behavior, rationality is selected as a more ideal design objective to achieve desired goals. However, considering the uncertainties and limited operation environments, it is not possible to target perfect rationality and exact best outcome.

Other than textbook definitions, AI can be classified as narrow AI, strong AI and super AI based on the capabilities [22], [23]. Narrow AI systems designed to carry out a single or limited task. These systems represent a certain level of intelligence for an exact specific problem. Artificial Narrow Intelligence (ANI) is used in AI powered chatbots, image and facial recognition, recommendation engines or self-driving vehicles. For example, self-driving cars are not programmed to drive other different types of machines such as trucks, ships. Moreover, a recommendation engine that is powered by weak AI cannot answer each type of question. On the other hand, strong AI, also called as artificial general intelligence (AGI) could perform any problem in any domain like a human being. It potentially has an intellectual and cognitive mental ability that is indistinguishable from the human mind. Strong AI-powered machines are able to think, learn and accomplish complex problems. However, there is a lack of progress about AGI and it is a controversial concept because some think it is not possible to achieve. Even though strong AI is still a theoretical concept for now, there is a step forward. It is expected that super intelligent machines will have consciousness. Super AI can demonstrate intelligence and perform any task even better than a human being, that was mentioned by University of Oxford scholars and some AI experts [24].

Besides the classification based on capabilities of AI mentioned on the above, it can be differentiated based on various paradigms such as sub-symbolic versus symbolic, knowledge-based vs. data-driven, logic vs. neural and so on. In order to place constraint programming among other approaches, the Fig. 12 has been selected.

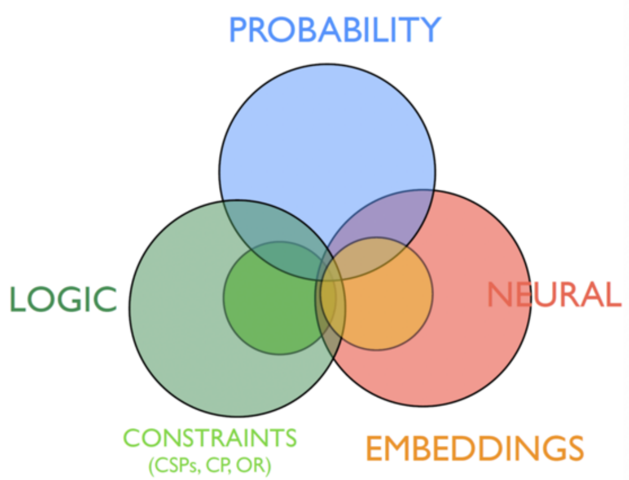


Figure 12: AI paradigms

Each paradigm has different types of strengths therefore they are used to sort out the issues according to specific kinds of tasks. As a case in point, AI neural approaches are successfully used for the tasks which require learning and perception. On the other hand, when the system is in need of an accountable reasoning, logic paradigm is focused. As another example, constraint and mathematical programming paradigms produce better results for solving combinatorial and optimization problems. Having these particular strengths of each paradigm, the AI community has worked to improve and integrate the capabilities of AI to make it more holistic and powerful. Luc De Raedt [96] claims that the combined use of different AI techniques based on probability, logic and neural help to acquire a trustworthy AI which include learning, reasoning, optimization capabilities.

Starting from early days, two major considerations are common when addressing AI; one is the practice of making machines with intelligence behaviors, this view centers the intelligent objects such as self-driving cars, smart robots. The other is the procedure to deal with problems that are complex and beyond the reach of human capabilities such as finding the right move in chess or different decision-making cases based on domain knowledge, big data, this view focalizes complex problem solving. In the thesis, the latter AI will be on the focus.

1.2.3. New Era

Artificial intelligence has become a mission critical both for industrial businesses and organizations and researchers. More than ever, AI-enhanced technologies are widely used in the industries and mentioned in academia, as indicated in the previous sections. With the advancement of AI technology in recent years, some specific sub-topics have been focused in order to imitate human behaviors and actions.

Although artificial intelligence, machine learning and deep learning are used interchangeably nowadays, they are not the same and even deliver different capabilities. The reason could be related with that AI is an umbrella term that is used for technologies which enables the machines to behave like human intelligence. According to Yuxi Li [11], Fig. 13 represents the domains and related concepts of AI. It can be seen that AI is strongly related with knowledge representation, reasoning, planning, robotics and problem solving.

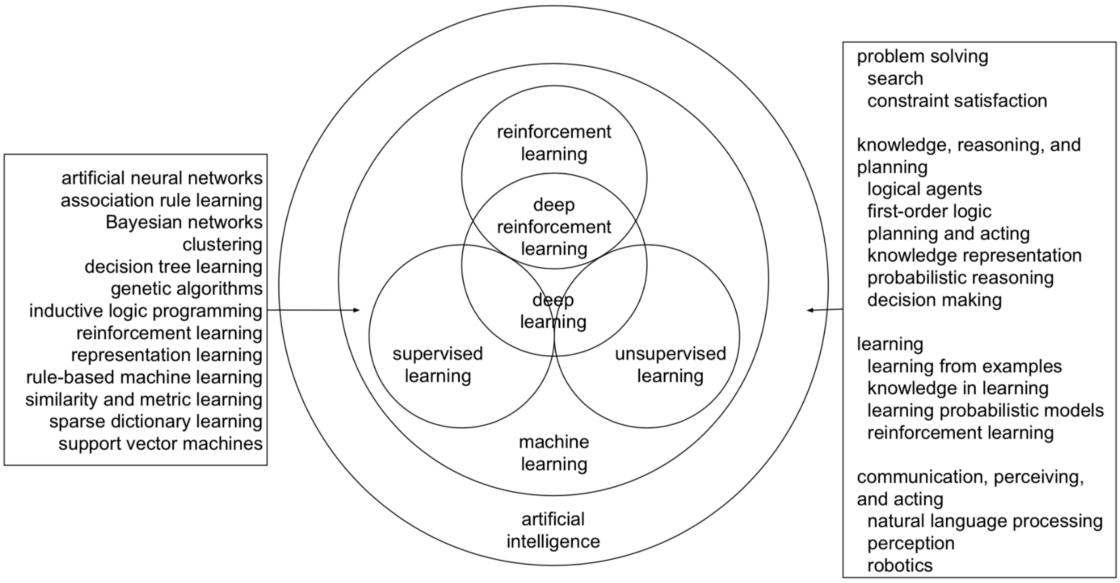


Figure 13: AI Concepts and Subsets

Some of the domains are explained in the following part:

Machine Learning is one of the most popular subsets of AI because instead of coding to task, machine learning algorithms use the data to be trained for how to perform a task, therefore these machines show the ability to automate the learning process by themselves. Moreover, the performance of accuracy and predictive power is improved in an adaptive manner with accumulated data. There are three types of training methods for ML algorithms [26]:

- Supervised Learning include well defined goals, i.e., direct outputs, with labelled data. These systems are used to predict outcomes and so they can solve classification and regression kind of problems.
- Unsupervised Learning methods are able to identify sequences, make clustering and segmentation with unlabeled and/or unclassified data. It recognizes interferences to solve the complex nature of the date.
- Reinforcement Learning target is to find the best possible action and maximize reward in case of a particular situation. It does not include a training data set; instead, it uses mistakes on its own, i.e., trial and error, with defined start and end states.

Deep Learning is a powerful subset of machine learning, and so artificial intelligence, too. The working principle based on multi-layer neuron networks which are connected to each other and in that way the data pass through from many hidden layers by self-directedly in order to be processed. It gives best performance while working with more hidden layers and big data set [27].

Neural networks allow to recognize patterns, classify and generate information. The artificial networks consist of connected node layers and starting from the input layer, the nodes send data to the next layer of the network. The system is inspired from the human brain and neurons in the way signals are transferred.

Computer vision is the ability to track patterns, generate, recognize trends and process images. These systems can be used to gather information from images, videos and visual inputs. In that manner, the machines can capture visuals on their surroundings and make interpretations in the real-life environments.

Expert Systems can be defined as computer systems which show decision making capability. The systems are able to make reasoning and solve complex problems. The systems have a knowledge base in its application domain and use an inference engine which uses logical notions in order to make deductions or choices.

Robotics focus to design, construct and apply the machines, which are called as robots, to substitute or replicate human actions. Except for computer programs, they have mechanical and electrical components such as sensors, effectors in order to operate in a real-world environment. Thanks to AI techniques, robots can understand physical inputs and plan their actions.

Natural Language Processing (NLP) relies on analyzing, understanding, and producing natural language and speeches. The research area includes not only speech to text generation and vice versa but also information extraction within the text, sentiment analysis etc. [28].

Problem Solving is a process of finding solutions especially for real-world problems from a model or data. However, reaching a solution for hard problems with direct methods i.e., using data directly, is not always possible for algorithms or humans. So, it is classified as a part of artificial intelligence according to computer science. The area focuses on a large number of techniques for algorithms and heuristics to solve problems such as genetic algorithm, fuzzy logic, neurons network and constraint satisfaction techniques.

Considering the topic of the thesis, we can clearly say that the thesis will mainly focus on problem solving. In the project management field, managers face several problems throughout the project life cycle. In such a complex and uncertain environment, it is not surprising how often they struggle among alternatives. Therefore, the success of the project and the managers can be measured with the solutions that are found as a response to the problems.

1.2.4. AI Effects

Throughout history, the introductions of new technologies have a significant influence especially on business environments. It is undeniable that AI has been one the most impactful and promising technologies for our decade. It can be confirmed with some indicators such as from an academic perspective, the number of AI journal publications increased by 34.5% from 2019 to 2020 as indicated in Fig. 14 [5] Moreover, in the business point of view, Mckinsey's survey [4] results show that 27% of the respondents increased their AI investments, while half of the respondents kept their budget same despite of the financial slowdown caused by the pandemic.

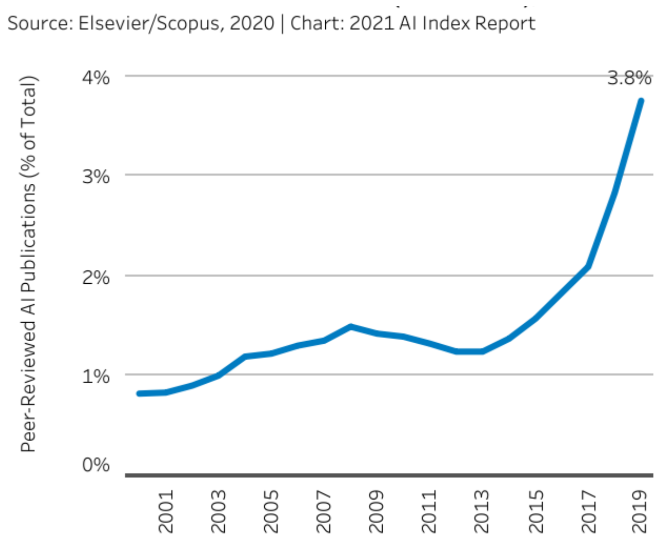


Figure 14: Peer-Reviewed Publications (% of Total) between 2000-2019

Using the right technologies in proper contexts can bring several benefits in terms of efficiency and effectiveness. As an example, according to Accenture 2019 Technology is Business report [6], research reveals that manufacturers that have proceeded to adopt AI will nearly double their competitiveness (1.8 times) in the following three years. Moreover, the effect of AI advancements on 16 industries was assessed and found that by 2035, AI could increase average profitability rates by 38%, resulting in a $14 trillion financial rise.

In Harvard Business Review Article Collaborative Intelligence: Humans and AI Are Joining Forces (2018) [72] identified five elements to improve business processes in the result of their work with hundreds of firms. These are the operational flexibility, speed, scale, decision making, and personalization of product and services. AI can boost all the characteristics by giving the right information at the right time. Over and above that AI could achieve limitless computing powers and disrupt every possible aspect of life. However, before these technologies become more prevalent, it is certain that an understanding of new technology and its impact is an essential requirement to get the most benefits. Not only technology, but also the possible application areas should be examined in terms of context and sector needs.

1.2.5. Artificial Intelligence Effect on Project Management

It is obvious that AI has been used in several different fields and adoption of artificial intelligence in recent decades is likely to continue in most of the aspects of our lives. Consequently, AI affects the methods and applications of project management and the role of project managers. According to Gartner's 'How AI will reinvent Program or Portfolio Management' report [3], by 2030, 80% of the work of today's project management discipline will be eliminated as AI takes on traditional project management functions. More specifically, the results of a 2021 survey [74] whose respondents are a group of PM experts show that among ten PM knowledge areas, project cost management, project schedule management and project risk management are the most likely to be influenced by AI in next 10 years.

AI can transform project management in several ways. At first glance, efficiency can be enhanced, and operational cost reductions can be obtained by substituting repetitive, administrative tasks with automatized and standardized AI which performs better compared to a human being. In such a way, project managers can focus more on the higher value-added and emotional tasks where AI falls short such as tuning projects according to strategic goals, developing relationships and creativity. Moreover, greater insights and problem-solving algorithms can be provided by AI for enhanced decision making. The constraint programming technology and project management relationship will be the central topic of the thesis.

1.2.6. Artificial Intelligence Applications on Project Management

Developments in technology have drastically transformed business practices. Organizations can greatly benefit from the integration of AI and PM to several applications. AI enables machine processing to make faster and more reliable decisions based on massive amounts of previously recorded data. The project insights can be presented by predictive analysis, risk-reward management assist and expert recommendations. The AI representations such as rule-based systems, logic formalisms and constraint languages can be integrated to problems of the project to guide the day-to-day or long-term strategic decisions. AI algorithms, software and devices enable to derive the best possible actions from scheduling to pattern analysis within the working team. Some areas and implementations can be explained.

Information Management: Like every interactive environment, communication between stakeholders is quite important for the success of a project. In Fig. 15 which is retrieved from PMI's Pulse of the Profession research, the factors for failure of strategic initiatives in an organization are given. The top 3 reasons are related to poor communication [64].

Figure 15: Most responsible factors for project failure

Technologies do not only include speech-to-text, natural language processing, sentiment analysis or Natural Language Processing. However also, computer-mediated communication and chat-bots assist teams and customers who could live in different time zones and speak different languages. The bots such as Howdy and PMbot enables to assign and check the status of tasks and employees without unnecessary back and forth and e-mail traffic. For healthier communications, AI can predict the rate of success of presentations and the most appropriate interactions with the target audience using various statistics and it can catch every response to process it.

Forecasting and Predictive Analytics: AI can follow the status of project and the progress and create a predictive model using analysis, statistics, and machine learning techniques. Based on historical data and the project progress, machine learning can forecast possible outcome scenarios in terms of the budget, time, and quality such as cost overruns, delayed schedules or poor outcomes. For instance, in a construction area, drone technology and cameras monitor and record the videos and photos of the activities on the site. The visuals can be interpreted with image processing to follow the progress on the area with a digital imaging-based method [63]. Another example is a performance and output tool Polydome which is an internal project assessment tool on PricewaterhouseCoopers (PwC). It identifies and connects employees based on their workload, skills, preferences, location to arrange a project team. Last but not least, potential problems during business process execution can be predicted with data mining, constraint satisfaction, machine learning [40].

Risk and Opportunity Management: PMI defines risk as an uncertain event or situation which creates positive and negative attributes on the project objectives in case of occurrence. Every project has uncertainties throughout the project lifecycle which could be related with project objectives and priorities, the relationships of stakeholders, nature of the project and so on. AI can assess the potential risks identifying the rate of occurrences and predicting the impact by analyzing previous project data, lessons learnt and status-quo of the project. Also, with the mentioned predictive capabilities, opportunity management could be used to facilitate decision making process for managers.

Decision-Making Support Systems: Managers and decision makers face with many problems and pass from several decision stages throughout the project life cycle such as supplier selections, cost-performance tradeoffs. An intelligent support can assist the problem solving and decision-making processes considering previous data, current input, multiple variables and constraints. An intelligent decision support system addresses a wide range of problems in the projects – strategic, tactical and operational. A human decision maker greatly benefits from a support system which supports evaluating and selecting the best possible alternatives using rules and logic models. The AI tools such as fuzzy logic, neural networks, genetic algorithms and expert systems have advanced and can be applicable on decision making support systems. Several real-world applications such as network, workforce or production planning, rostering, supply chain optimizations can be mentioned as examples [68].

Besides the given applications, there are lots of actual and potential AI technologies. In the survey [66] whose respondents are the project managers in diverse fields, the most potential PM aspects for the usage of AI is demonstrated. The Fig. 16 shows the results, the most prominent ones are information management, project planning and budgeting.

The role of project managers who are at the center of moving parts among several parties demands more AI applications to smoothen the process, integrate stakeholders and derive more reliable outcomes. AI enabled project management systems do not only provide the efficiency of tasks or computational improvement over human-beings performance, but also ensure concrete basis while making decisions. AI holds a key to helping to avoid human biases and unfair evaluations.

The role of project managers who are at the center of moving parts among several parties demands more AI applications to smoothen the process, integrate stakeholders and derive more reliable outcomes. AI enabled project management

systems do not only provide the efficiency of tasks or computational improvement over human-beings performance, but also ensure concrete basis while making decisions. AI holds a key to helping to avoid human biases and unfair evaluations.

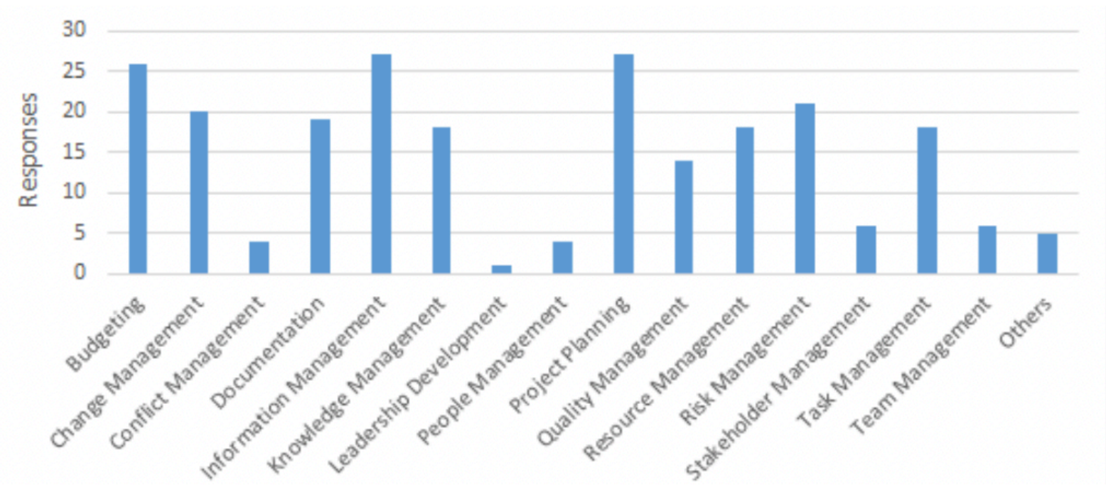


Figure 16: Aspects of PM which could use AI support

1.2.7. Challenges of Artificial Intelligence Implementation to PM

Although there are lots of opportunities of AI, there are risks and prerequisites for a successful AI implementation. In nature, each project has unique characteristics. Moreover, within a project environment, the people from different backgrounds, purposes and ideas meet. This kind of social dynamics and project complexities can be solved in the way of searching for a balance. Therefore, these can cause a fall in the performance of AI in practice because it works better in standardized and specific tasks and lacks emotional, moral and ethical capabilities. However, the use of efficient AI involves several risks such as lacking verifiability and transparency due to their opaque structure and functionality which can cause strict dependency to the technologies and unnoticed errors.

The integration of AI systems to the existing ones requires time, financial and soft resources such as a team which is open to innovation, skilled and motivated. Moreover, computational processing power, reliable massive amounts of data and compatible infrastructure are essential which are also demonstrated in Fig. 17. Therefore, before implementing AI, what is needed for raising the success rate and what can be taken from AI should be evaluated considering the maturity and readiness level of the technology and organization [65].

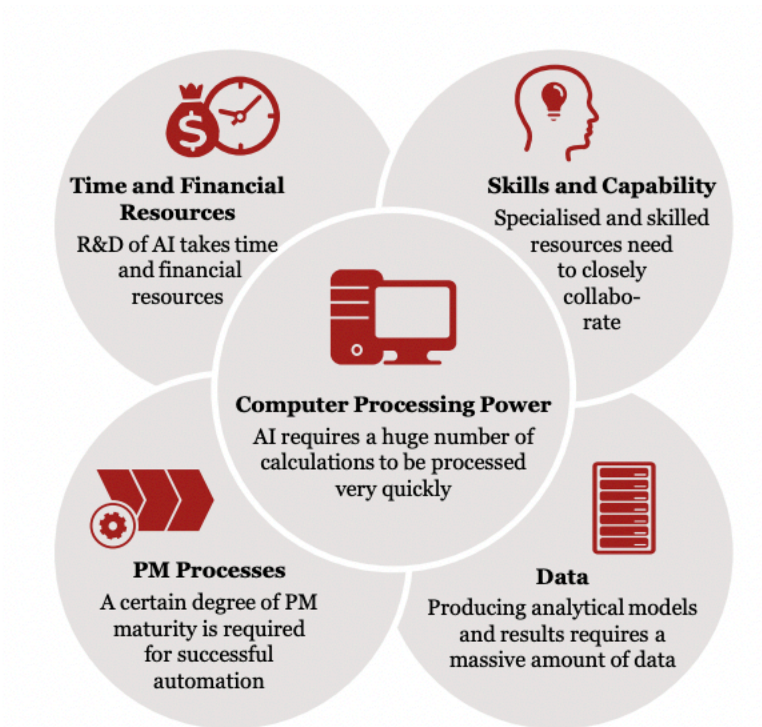


Figure 17: AI implementation needs

1.3. Operation Research

Operation research focuses on the application of mathematics, statistics and algorithms to business questions which concern the coordination and conduction of the activities, processes and operations. Kandiller [34] defines it as the development and application of quantitative models and methods to support decision making which help to make better business decisions scientifically. In a nutshell, it can be viewed as a systematic and analytical approach to decision-making and problem-solving [33].

Many organizations such as American Airlines, Google, IBM use decision modelling, which is generally referred OR, to solve complex problems and make managerial decisions to provide the best effective usage of their resources. In such a way, the process and results can be dealt without individual bias, emotions, or guesswork. Therefore, the applications of quantitative decision modelling techniques are commonly utilized in project management. The efficiency of the operations research is proven lately with the reliable results observed by managers using the OR techniques. As the scope of the operation research is wide for instance queuing theory, optimization, game theory, data science and etc. nowadays, it is a common practice business leader to utilize OR in every aspect of the managerial decisions [97, pp. 1-7].

1.3.1. Operation Research Tools and Techniques

Operation research uses several techniques and tools to solve problems. The models used can be classified into two sections as deterministic and probabilistic models. The former deals with the problem with certainty which accepts all input data as fixed and known. On the other hand, some information about the problem can be uncertain or unknown, in such a case, probabilistic models are used although they do not guarantee an accurate result. However, some problems may not be possible to solved with various kind of OR models such as linear programming, integer programming etc., in such a case heuristic method can be used to search for a solution.

The classifications are done according to the nature of the problem context. The performance and power of the techniques are problem dependent. Therefore, the approach is selected properly considering any particular problem. Moreover, new methods and tools are being devised day by day to increase the efficiency of the solutions. In such an environment, classifying all of them is not possible. Although project management consists of sides from deterministic as well as stochastic modelling, in the literature review part, the main mathematical programming techniques of problem solving were investigated furtherly because the main techniques and methods of OR are based on mathematical programming (MP) with [33], [97]. In the view of the fact that, mathematical programming seeks solutions for a wide scope of problems which could be feasible solutions that satisfy all the constraints of problem or an optimal solution as a best alternative from a set of feasible solutions.

Some main techniques can be listed as following based on the procedures:

Mathematical programming (MP) techniques:

- Linear Programming
- Non-linear Programming
- Integer Programming
- Dynamic Programming
- Goal Programming
- Network Scheduling (PERT/CPM)
- Game Theory
- Inventory Control Model

Stochastic procedure techniques:

- Simulation Methods
- Decision Theory
- Markov Process
- Stochastic Programming
- Queuing Theory

a) Linear Programming

Linear programming, which is also known as linear optimization, is one of the most known techniques used in the class of mathematical programming. It is used for selecting the best option among the set of feasible alternatives in a case in which the objective function and constraints can be given as linear functions involving variables. The constraints can be expressed as linear equations or inequalities with variables. Linear programs are commonly solved with simplex algorithm which is an iterative process that can find an optimal solution, an infeasibility or an unboundedness in which objective function is infinite.

Regards to, there are several use cases of linear programming but there are some limitations as well. These limitations consist of starting from the construction of the problem model, which can be problematic due to the possible nature of the non-linearity of a problem. In order to solve a linear programming problem, the imposed limitations, i.e., constraints, must be satisfied and the values which optimize the objective function(s) must be found. In other words, it assigns a limited number of resources such as power, time, materials to demands such as effectiveness and efficiency as follows that the objective function is optimized, and conditions are satisfied. In summary, it is a constrained optimization technique i.e., to optimize the total effectiveness such as minimizing the cost or maximizing the total profit [33, Ch. 3].

b) Non-Linear Programming

Non-Linear Programming (NLP) is used for solving optimization problems where either some of the constraints or the objective function are nonlinear which enable it to represent many aspects of real-life which do not behave in a straight-line manner. As an example, doubling the working hour of an employee does not double the output considering the inefficiencies such as tiredness, fatigue, and lack of attentiveness. In factory location determination, traffic modelling and design problems which include analyzes, trade-offs can be given as example problems. In project management, portfolio mix determination requires to maximize return and minimize risk, which can be modelled nonlinearly [33, Ch. 12].

c) Integer Programming

Integer programming (IP) is a mathematical modelling and solution algorithm to optimize, in maximization or minimization sense, a linear objective function based on equality, inequality, and constraints which take only integer values in a linear form. Therefore, in that sense integer programming can be accepted as a subset of linear programming [41]. However, it must be mentioned that the admissible domain of IP is a discrete set of points unlike linear programming. The problems can be modelled accurately because of the nature of integer programming that enables discrete optimization e.g., yes-or-no decisions such as launching a plant or buying equipment, portfolio optimization, routing etc. Bradley, Hax, and Magnanti [43] imply that IP models are one of the important models in management science.

In general, solving integer problems is harder than integer ones and consequently the effectively solvable size is much smaller than the integer programs [33, pg. 24]. The reason is that results which optimize the problem change with the constraints imposed on the integers in a way that is not viable anymore, heuristically. Moreover, optimization problems become non-convex with integer variables. Due to the fact that the memory and solution time can increase rapidly with respect to added integer variables.

The common solution techniques include:

- Branch and bound is a popular method for solving combinatorial optimization problems using a systematic process that utilizes linear programming approaches with restricted subsets of integer values and then these subsets are compared to end up with the most suitable option, to find the optimal solution. It tries to enumerate all feasible solutions.

- Cutting Planes builds on extra constraints to the model to make the search area narrower. The newly added constraints succeed to reduce the feasible region until an integer optimal solution is obtained. The Applied Mathematical Programming book [42] represents that the cutting planes method almost always falls behind the branch and bound.

- Heuristics algorithms: It searches for integer-feasible solutions. Heuristic approach aims to provide not the optimal but approximate solution within a reasonable time therefore it is used when the classical methods are slow.

Simulated annealing, tabu search, and k-optimization methods can be given as examples.

As a solution strategy, a combination of techniques can be applied to take advantage from multiple methods such as branch and cut algorithms which consist of cutting plane method for its fast pace and branch and bound algorithm for its reliability.

d) Mixed Integer Programming

Unless all the decision variables are discrete in integer programming i.e., there are integer constraints on only some of the variables, the model is a mixed-integer program (MIP). Also, MIP problems are generally solved with algorithms that are used in integer programming.

In these days, the area of operation research was mostly established around inequality-constrained mathematical programming models such as nonlinear programming (NLP), mixed integer/linear programming (MILP), and mixed integer/nonlinear programming by Hooker and Hoeve [39]. In real world problems, we usually have to overcome more complex constraints beyond the linear ones. MIP formulations are used to solve complex optimization problems and make decisions considering conflicting objectives and trade-offs with a powerful mathematical modelling approach [33].

e) Network Analysis – PERT and CPM

Network scheduling models such as Program Evaluation Review Technique (PERT) and Critical Path Method (CPM) are used for planning, scheduling, monitoring and controlling large and complex projects. In order to recognize bottlenecks and minimize delays, unproductive time and uncoordinated jobs, the critical path and critical activities which tend to cause a delay must be identified with network models. The models exhibit the flow and sequence activities with nodes and arrows considering the relations among all activities. The appropriate technique can be selected according to the project nature and certainty of project activities duration. For example, using the probabilistic model PERT is more suitable in projects which contain non repetitive events and uncertain activity time. Therefore, it uses more than one estimate of duration for an activity to determine expected time for activity duration and the variance. PERT technique allows to grasp the percentage of completion of a project by a specific time. On the other hand, CPM is a deterministic technique and so, it is used to optimize schedules where time consumed by activities are exactly known or estimated. Nevertheless, in some cases, CPM has a *crash time* which is the minimum time that the activity would take if extra expenses were used for completion of the activity. The model enables distinguishing critical and not critical activities and *crashing* which is shortening of a duration for an activity with minimum additional funds and resources [97, pp. 119-156], [42, Ch. 8].

f) Dynamic Programming

Dynamic programming is considered as a computer programming and mathematical recursive optimization method in which a discrete optimization. It solves problems by dividing the main complex problem into simpler and smaller sub-problems. More specifically, the program produces outputs for the sub-problems and then stores the outputs to optimize a bigger sub-problem. Therefore, it can be applicable to multistage decision processes [43].

g) Goal Programming

As a branch of multicriteria decision analysis, the method utilized to deal with optimizing issues containing more than one objective is called 'Goal Programming'. This approach can be considered as a variation of linear programming with wider scope. Goal Programming tries to find the best possible outcome when there are conflicting constraints. These constraints must be prioritized by some sort of a criteria, defining related ranks in order to successfully implement goal programming [97, Ch. 5].

h) Game Theory

Game theory is used to model conflict situations in which there are two or more competing actors, which are called as players. The conflict can be interpreted as that the more one of them wins means the more the opponent player loses. The players take position interdependently i.e., based on expectations about how the opponent would make a decision [33, Ch. 14]. In the business context, it helps to evaluate and select strategic choices such as pricing competition, product releases or retires decisions and so on.

i) Inventory Control Models

In a company, the decisions for purchasing, stocking, tracking, transporting must be taken carefully. Throughout whole supply chain process, the demand and supply level of different product types are considered with inventory control models. Some examples of most common inventory management algorithms are Economic Order Quantity (EOQ), Inventory Production Quantity (IPQ) and ABC Analysis. The optimal order level, size, time can be selected for orders and stocks [97, pp. 495-524].

1.3.2. Operation Research and Project Management

The knowledge and tools of OR can be applied to a wide range of areas such as finance, manufacturing, military, transportation and so on to solve diverse problems. Some known use cases in the industry are floor planning in a factory within limited space and machines or determining the most efficient product mix when lots of products with different financial margins and production specifications exist or demand forecast, resource allocation, scheduling etc. In the cases which include absence of complete information and scarce resources can dissolve with operation research because it understands the system behaviors and analyze them by developing the appropriate models and predicts the future behaviors. That is the reason why OR and management science are sometimes used interchangeably because of their common scope and the main responsibility of project managers which is making decisions. The advisory role of operation research applications helps managers to select the best alternatives with quantifiable reasoning [35].

According to Tavares's study [30], the role of optimization in project management can be categorized into 3 main areas based on the published papers between the years of 2000 and 2009, which can be seen in Fig. 18. These are the followings:

1. Modeling the network of the project activities (i.e., scheduling and sequencing)
2. Resource allocation methods
3. Project evaluation methods

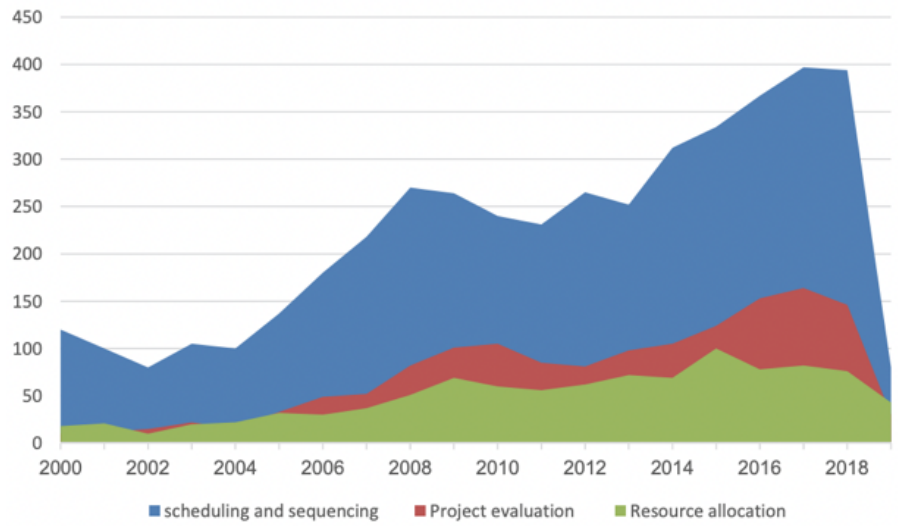


Figure 18: Optimization related papers in PM published between 2000-2019

Although optimization and problem solving can contribute to project management in other areas such as human resources, quality management, risks etc., the main focus of the thesis selected as the main promising area to optimize, scheduling.

1.3.3. Constraint Programming and Operations Research

The main purpose of constraint programming and operation research are the same which is formulating a real-world scenario into an appropriate model and finding a solution efficiently. They not only explain the phenomena but also accompany to decision making. According to Hooker (2007) [39], although the CP and OP have different techniques, they have the same theoretical and practical tasks such as optimization of scheduling, allocation, planning, logistics, product design, etc. For example, while researching the recent research on the job shop problems under Industry 4.0, the main optimization algorithms for scheduling are classified as the following Fig. 19 by Zhang, Ding, Zou, Qin and Fu (2019). It can be seen that several methods from artificial intelligence and operation research exist to optimize scheduling problems [31].

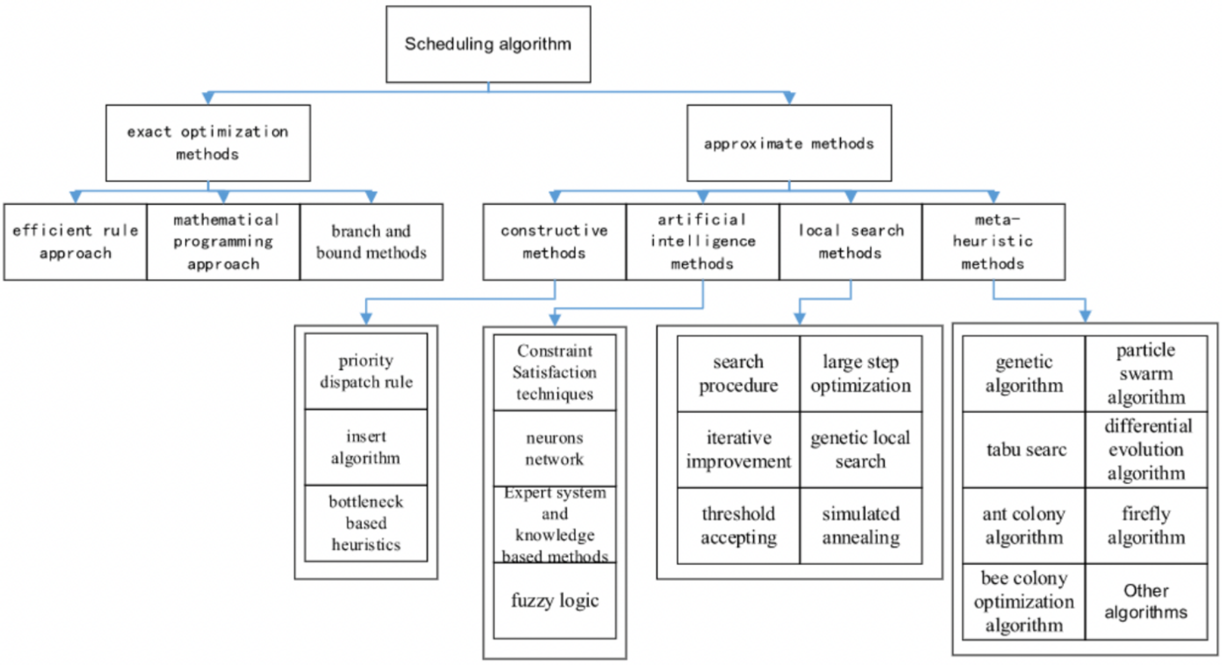


Figure 19: Optimization algorithms for scheduling

The operation researchers and constraint programming communities have lately started to realize each other. Therefore, these two empirical sciences evolved differently although they usually both work with the same structure which are models, constraints and objective functions with different terminology and formulation. However, there is a growing effort to understand each other and integrate them in the last few years such that common academic studies such as CPAIOR (Constraint Programming, Artificial Intelligence, and Operations Research) conferences and commercial applications are being conducted. As another solid example, two IBM employees Jean-François Puget and Paul Shaw [38] listed concepts of the two methods terminologically to help communities to understand each other as Table 3.

Table 3: Terminology differences between MP and CP

What CP call it	What MP people should understand
Programming	Computer Programming
Planning	Programming
Solution	Feasible Solution
Optimal Solution	Solution
Variable	Decision Variable
Variable Domain	Variable bounds or set of admissible values
Constraints	Not limited to linear, quadratic or mixed integer
Tree Search	Branch & Bound
Heuristics	Branching Strategy
Constraint Inference	Presolve
Constraint Propagation	Bound Strengthening
Global Constraints	Specialized Algorithm

Linear programming finds solutions by implementing inequality constraints and this method is the groundwork of OR. Indeed, nowadays, there are many other methods such as nonlinear programming, mixed integer/linear programming, and mixed integer/nonlinear programming derived from mathematical programming models with inequality constraints [39]. In Fig. 20 represents that these models can be seen as a sub-set of CP. For example, Berthold [58] clarifies that mixed integer programming can be accepted as a specific case of CP programming in which constraints and the objective function are linear and domains are intervals in integer or real valued. Also, the counterpart of CP is seen as MIP in MP considering the referred problems.

There are some differences between CP and MP techniques. First of all, the meanings of MP and CP are different. In OR literature, especially for MP, ‘programming’ refers to the plan of actions or activities. However, ‘programming’ in CP implies computer programming that seeks a way of solution to a particular problem. As another difference, CP has less concern to find optimal solutions than MP. In the absence of an objective function of a CSP, the main purpose is to find a feasible solution quickly that is able for large problems with thousands of individual constraints. The IBM report “Solve your Toughest Planning and Scheduling Problems: How Business Managers can use Mathematical Optimization Technology” [84] indicates that CP can solve large routing and scheduling problems much faster than MP models, also even if MP is able to solve.

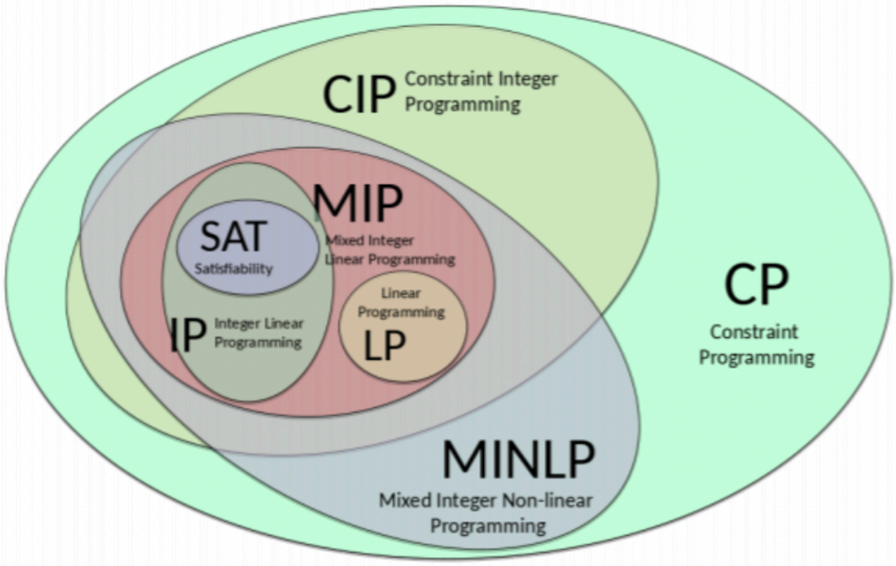


Figure 20: The scope of CP and other programming approaches

In mathematical programming, the constraints just represent the problem. Therefore, using a specialized solver or customized program in a traditional programming language are required to solve combinatorial problems. On the other hand, the constraints in CP invoke a system which repels infeasible solutions through integrated implementation of several heuristics, programmed search and constraint solvers. In addition to, CP allows defining broad range of constraints with logical such as *AND, OR* or special operators such as *alldifferent*, *forall* which are global constraints. Unlike, MP models have only some limits on it such as linear, quadratic convex, non-negative. The additional global constraints are easily formulated and provide a concise modelling power to CP. Moreover, there could be non-linear, non-differentiable, and discontinuous constraints. However, no traditional exact OR techniques such as LP, NLP, MIP, etc. can solve models including these constraints [52]. It is a remarkable advantage when considering real life, the constraints show nonlinear nature.

Baptise (2001) [51] indicates that constraint programming provides algorithms which are broadly applicable to a wide range of problems but not highly efficient for solving a problem. On the contrary, although operation research algorithms might not be fitting in practical formulations, they are more powerful in improving the solution of the problems. Therefore, some hybrid methods are applied to several problems. For example, S. M. Pour et al. [54] apply such a hybrid framework to a crew scheduling problem which consists of construction the model and initial solution with CP and then using a MIP solver to improve the initial solution furtherly. In general, CP has more modelling capabilities and provides interaction among constraints but not rich for reasoning on the objective function. Unlike, MIP models are not flexible and cannot define global constraints despite having rich solution methods and global reasoning on optimality. In a case of solving the problem directly with a standard MIP solver, it does not provide any feasible solution for more than two weeks planning horizon due to a large number of real-life aspects and constraints.

CP and OR integration such as hybrid methods which complement each other's powerful parts is still an active and promising research area to improve the methods of solution to large variety of problems. In another article, Hooker and Hoeve [39] summarize such strategies as:

- Use relaxation from OR and propagation from CP together
- Apply OR techniques for domain filtering in CP
- Separate the problem into sub-problems that suitable for OR and CP
- Implement constraint propagation to dynamic programming models.

To witness the advantages of hybrid methods, some computational examples can be reviewed that compare pure and CP&OR hybrid methods. In the paper of Jain and Grossmanny [60], a scheduling problem involves minimizing a least cost is formulated with CP, MILP and combined hybrid models and their performance are compared. As a result, the integrated method tackled better than pure ones because of the specialized propagation algorithm which is a power of CP to find feasible solutions to discrete optimization problems, and LP relaxation that is used for proving optimality as a strength of MILP. Thanks to the hybrid method, CPU time is reduced 2-3 orders of magnitude. Moreover, Hooker's Operations Research Methods in Constraint Programming [7, Ch. 15] mentions some hybrid applications with their impressive results. It is shown that not only pace improved with hybrid methods, but also more instances can be solved, further investigation can be done to the [61] and [62].

However, it must be highlighted that nor solving combinatorial problems is the only field that CP studies; neither it is not logical to limit operation research with mathematical programming. The research area of OR also includes several exact, heuristic and metaheuristic techniques for problem solving. Also, the role of dynamic programming and network theory is quite remarkable in CP.

1.4. Constraint Programming

CP is a paradigm for modelling and solving variables of constraint satisfaction and optimization problems. The approach allows to express constraints and uses deductive reasoning to narrow the search space. It is a software technology which is used to find effective solutions to combinatorial and real-life problems. More specifically, scheduling, resource allocation, production planning, routing can be considered as combinatorial real-life problems. The aim is to find solutions for a set of variables that satisfy a set of constraints which are given by the user. In other words, the process of constraint programming is to solve problems by defining requirements that impose conditions about the problem area and then, finding values that meet all the conditions i.e., constraints [13]. For instance, arranging a lecture timetable contains many constraints in terms of class unavailability, lecturer capacity, time and so on. As a mathematical example, let us say X and Y are variables that are in the domain {1, 2, 3}, and a constraint relation is $X \neq Y$. In such a case, some instantiations such as (X, Y) = (1, 1), (2, 2), (3, 3) are eliminated due to the constraint.

Constraint programming can be divided into two phases – modelling and solving. In Fig. 21 represents the phases and the related elements.

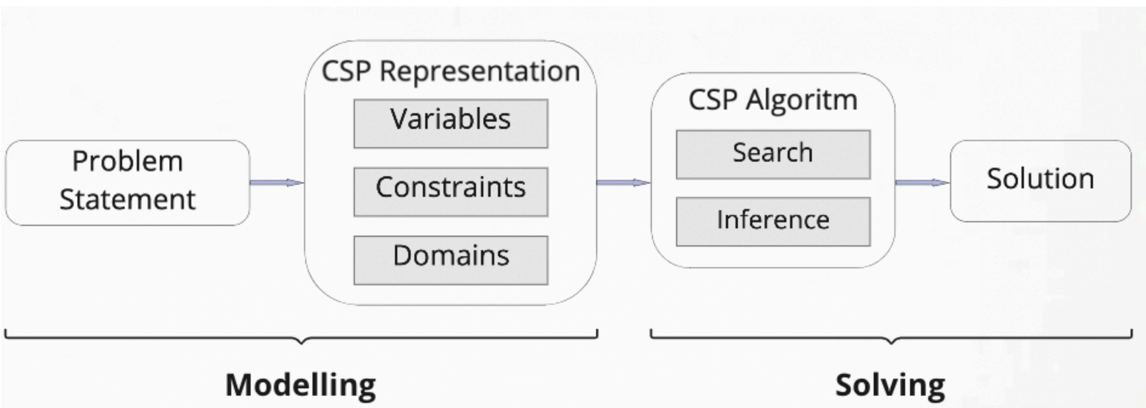


Figure 21: CP problem solving framework

In the methodology, in order to tackle a problem using constraint programming one makes three design decisions which are the constraint model, the search or inference algorithm to solve the model, and if needed, the heuristic to guide the search. After passing these stages, the solution of the problem is found. Due to the nature of solving of a CSP which contains several technicalities of mathematics and computer science disciplines, a brief literature review will be mentioned in the thesis. Moreover, a generic constraint solver and language can deal with the solving part of a problem without any burden to the programmers.

1.4.1. Characteristics of Constraint Programming

CP is accepted as quite interdisciplinary. The spread and success of CP has been emphasized in many literature sources, theoretical and practical developments. Around the topic, an active community is formed, and its applications are extended to several domains. These domains are agents, bioinformatics, design and configuration, graphics, visualization, user interfaces, human-computer interaction and decision support, robotics, machine vision and computational linguistics, scheduling, planning, resource allocation, temporal and spatial reasoning and so on. As it seen, the constraint programming paradigm can be used in several disciplines. According to an international journal “Constraints” [29], the scientific and engineering relevant subjects of CP are artificial intelligence, automated reasoning, combinatorial algorithms, databases, discrete mathematics, operations research, programming languages, satisfiability and computational logic.

The term programming can have different meanings in different areas. Lustig and Puget [1] highlights that mathematical programming implies solving optimization problems mathematically unlike computer programming that refers to coding and writing a computer program in a programming language. The term programming in the CP takes some meanings from both. Specifically, it provides efficient solutions to discrete optimization problems. Also, the solution methods are integrated into a programming language. Therefore, in a nutshell, constraint programming can be described as constraint solving embedded in a programming language to solve problems.

1.4.2. History and Origin

It is necessary to discuss about the historical evolution of the constraint programming from past to today, in an effort to form a basis for the focal point of this thesis.

Constraint Programming is a technique devised in order to solve complex problems having multi optimization criteria. In more simple words, it is the programmed models depending on constraints. The roots of the constraint programming can be tied to as back as 1960s when there were explorations about human-machine interface development, and these are the prerequisites for CP thus deemed as the origin. Then, in early 1970s with the progress declarative programming languages which means the code written in a way of describing commands verbally, led to the works in artificial intelligence. Also, at the same time, the conceptual base of some target problems of CP such as arch and path consistency started to show up. However, the major building block of constraint programming was in 1980s with the discovery of constraint logic programming (CLP), which is an ad hoc approach of constraint solving where unification in logic programming is applied. By that time to today, the main concept of CP stayed unchanged as the tech of utilizing various combinations of search methods and unique inference techniques taking constraints of the problem into account but of course many different approaches have been tried to be implemented around this core. In addition, CP found upon to deal with feasibility problems and named as Constraint Satisfaction Problem. Furthermore, there has been lots of use cases where effectiveness of CP can be observed throughout the history. Although, the most salient one is scheduling type of problems, planning, configuration, vehicle routing, networking, bioinformatics, and others can be named as the examples of problems solved by CP methods [89], [90].

1.4.3. Constraint Satisfaction Problem

Like the book title of Karl Popper who is one of the most impactful philosophers of science, "All life is problem solving." People face with some problems and try to cope with them every day. After identifying the problem, we generate potential solutions and choose one among them such as in light of our analyzes, knowledge and reasoning. Solutions can be lie in many different ways.

As indicated in Section 1.2.3, in computer science, problem solving and optimization are accepted as a domain of artificial intelligence. It uses diverse techniques like human beings. In order to find solutions, AI problem solvers can be utilized from constraint programming. Solving a problem with constraint programming requires to model it as constraint satisfaction problem [83].

The formulations of the problems are expressed as a set of variables with a range of possible values and constraints on these variables. A CSP consists of three elements, V, D, and C:

- V is a set of variables, $\{V_1, \ldots, V_n\}$.
- D is a set of domains, $\{D_1, \ldots, D_n\}$, one for each variable.
- C is a set of constraints that express allowable combinations of values.

In order to satisfy a constraint, the tuple of the values of its variables must correspond the constraint relation. Hence, the discovering a tuple on the set of variables that are simultaneously satisfying each constraint of the problem is the key for finding a solution. A solution of a constraint problem assigns a value to all the variable such a way that all the constraints are satisfied.

In CSP, state is defined in terms of values of the variables V_n from domain D_n. A state space is needed to define to solve a CSP and each state is expressed by an assignment of values to the variables such that $\{V_1 = D_1, . . ., V_n = D_n\}$. It is important to define two essential terms for a problem-solving method; goal and solution, in order to clarify the concept. The possible mix of values convenient from subsets of variables fulfill the defined constraints determines the goal test. A solution of a CSP is referred to complete and consistent set of findings. Fig. 22 represents the position of CSPs among standard search problems.

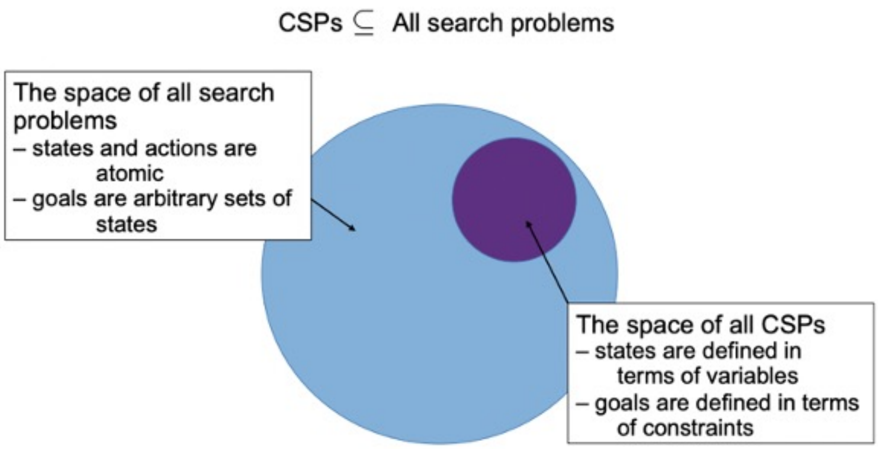


Figure 22: The position of CSPs among standard search problems

A solution must be *consistent* which means no violation of any of the constraints and *complete* which means the solution includes all variables. In solution phase after encoding the problem with a mathematical or logical modelling language, diverse range of search techniques and constraint propagation can be applied. The more detailed information about solving will be mentioned in Section 1.4.7.

The aim is to resolve the problem, so a constraint solver is used to find solution to the model defined and in that wat the values are assigned to the variables satisfying all the constraints. The solution of CSP can be:

- all solutions,
- only one solution which does not provide any preference to select among a set,

- an optimal solution regarding the given objective function which is characterized by some different factors that generate output.

An objective function can be added to the problem in order to find minimized or maximized solutions among the all the feasible solutions. These problems modelled with objective function are classified as Constraint Optimization Problems. In such a case, a solution is an assignment of values to the variables such that constraints are satisfied, and the objective is optimized [32].

Some examples of constraint satisfaction problems can be given as crossword puzzle, sudoku, scheduling problems, n-Queens problem and etc. In order to clarify, some examples are selected and the elements of the CPSs are defined as:

Sudoku: Sudoku is a logic-based game that is played all around the world. The task is filling the empty cells in a way that the numbers from 1 to 9 place once in each 3x3 squares, columns and rows which can be seen in Fig. 23. Although it is easy to solve for the smaller dimensions and no match for today's AI, some AI algorithms are used for solving sudoku problems due to its NP-hard nature [50]. For example, it can be solved using CP. To model it as a CSP, the following approach can be used.

Each cell is a variable for the problem. The variables can be represented as X = {*A1*, *A2*, …, *I8*, *I9*} which the first letter shows the row and the second one the column. As an example, for the notion, *C5* is the variable for the cell placed to Cth row and 5th column. The given values in the cells are directly set to the variables.

	7					4		
			4			5		3
	4							
	2				3			4
		4				9		
				5			6	
	3	2		9	1		7	8
6	8				2	1		
			8		6			

8	7	1	3	2	5	4	9	6
2	6	9	4	1	7	5	8	3
5	4	3	6	8	9	7	1	2
1	2	6	9	7	3	8	5	4
3	5	4	1	6	8	9	2	7
7	9	8	2	5	4	3	6	1
4	3	2	5	9	1	6	7	8
6	8	5	7	3	2	1	4	9
9	1	7	8	4	6	2	3	5

Figure 23: A Sudoku problem and its answer

The domain of each variable is the numbers {1, 2, 3, 4, 5, 6, 7, 8, 9}. Expect for the ones that are already given, theirs are the single given number. The constraints express each row, column and 3x3 square include all different numbers. In a 9 row Sudoku puzzle, there are 9 row constraints to ensure that all numbers have to different numbers in a row. The row constraints are {*alldifferent*(*A1*, *A2*, *A3*, *A4*, *A5*, *A6*, *A7*, *A8*, *A9*), …, *alldifferent*(*I1*, *I2*, *I3*, *I4*, *I5*, *I6*, *I7*, *I8*, *I9*)}.Similarly, all numbers have to different in a column which can be formulated as {*alldifferent*(*A1, B1, C1, D1, E1, F1, G1, H1, I1*), …, alldifferent(*A9, B9, C9, D9, E9, F9, G9, H9, I9*)}. Lastly, the difference of the variables placed in each square must be guaranteed such that {*alldifferent(A1, A2, A3, B1, B2, B3, C1, C2, C3) …, alldifferent(G7, G8, G9, H7, H8, H9, I7, I8, I9*)}.

Cryptarithmetic Problems: As classical cryptarithmetic problems in AI, mathematical puzzles where digits are replaced by letters, can be modelled as CSP. Each digit is substituted with an alphabet or a symbol to get the correct result. For example:

Given a cryptarithmetic problem is S E N D + M O R E = M O N E Y

	S E N D
+	M O R E
=	MONEY

The problem is assigning the digits to the letters that make the expression, adding SEND and MORE that gives to MONEY as a result, correct. Arithmetic operations can be carried out on the given problem and the constraints based on arithmetic and cryptarithmetic rules are the followings.

- A unique digit should be substituted with a unique alphabet.
- Digits should be in between 0 and 9.
- The result should obey he predefined arithmetic rules, i.e., 2+2=4.
- The first digit of three numbers cannot be zero.
- One carry forward is allowed in addition operation.
- The problem is allowed to be solved in right-hand side or left-hand side.

In order to find the distinct digits that represent the letters and satisfy the constraints, modelling the problem and defining the elements of the CSP is required which can be expressed by a triplet (V, D, C) such as:

- A set of variables V = {V1, V2..., Vn}
    - S, E, N, D, M, O, R, Y
- Domains: D
    - {0, …,9}
- A set of constraints: C = {C1, C2, ..., Ck}

- Distinct variables, S≠M, M≠S, E≠M, M≠E, …
- S≠0, M≠0
- $S*10^3 + E*10^2 + N*10^1 + D + M*10^3 + O*10^2 + R*10^1 + E = M*10^4 + O*10^3 + N*10^2 + E*10^1 + Y$

Considering that each letter can take a number from 0 to 9, the size of permutations is 10^7. However, the search tree can be narrowed to the following.

$S \in \{9\}; E \in \{4, . . . ,7\}; N \in \{5, . . ., 8\}; D \in \{2, . . ., 8\};$
$M \in \{1\}; O \in \{0\}; R \in \{2, . . .6, 8\}; Y \in \{2, . . ., 8\}$

The domains of variables are reduced, and the size of possible solutions is decreased to 5488 with using constraint propagation [12].

1.4.4. How It Works

In the publication of Baptiste [53], the model explains the behavior of a constraint programming system with the Fig. 24. As an important point, the problem definition, analysis mechanism constraint propagation and solving are divided clearly. It is one of the main features of CP because it is particularly useful for flexible and reusable systems. To clarify, firstly, the problem is characterized with variables and constraints. After, constraint propagation algorithms are stated to analyze the model. Lastly, solving phase is implemented to make a decision and such as a tree search algorithm detect all inconsistencies and specify how new constraints are built to the system. It can be exemplified with cryptarithmetic problem, which is in the previous subsection, constraint propagation, i.e., inference, limits the search space by removing some values which violate constraints.

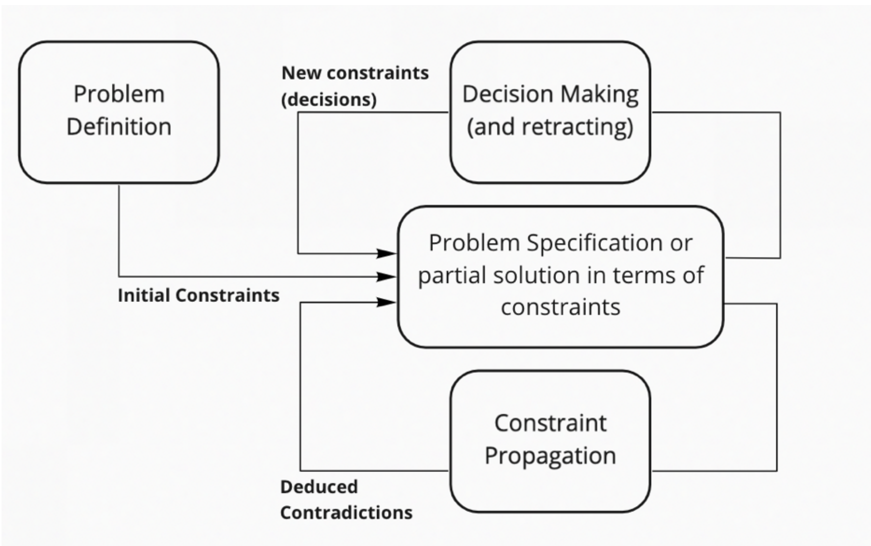


Figure 24: The behavior of a CP system

As the solutions are checked and decisions are made to decide whether modify or retract the constraints or deduce the new ones or not. The system cycle works until a solution of the initial problem is found which satisfy all the constraints. The mechanism applied to the cryptarithmetic problem, and the search tree is illustrated on the Fig. 25.

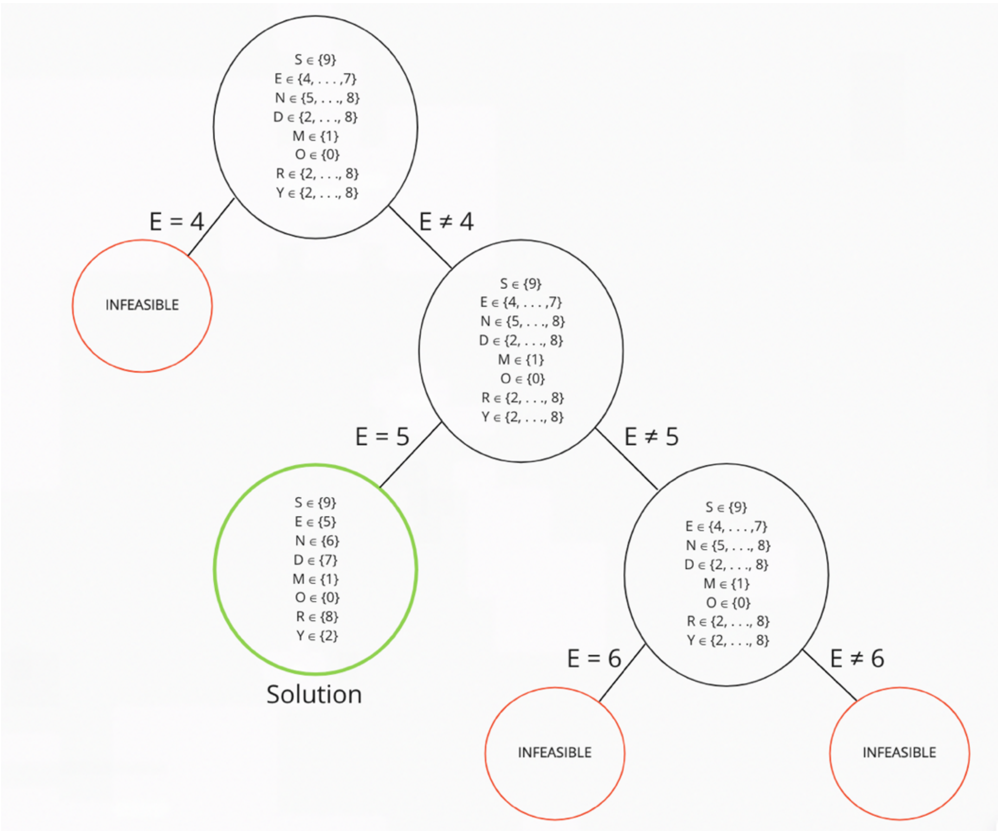


Figure 25: Search tree for Send More Money example

In order to understand how CP is utilized in an overall system, some examples are given to observe the function of CP as a support to decision support mechanism. In the following systems, we can observe the CP system for feasibility or optimization. In decision problems, the users have only to decide whether a solution exists that meet all constraints. Unlike, an objective function has to be optimized in optimization problems. However, all in all, it supports a decision maker to make a decision with solving constraints.

As an Optimization Model

In order to evaluate the influence of uncertainties belong to supply chain, a two-step analytical research was done [77], and the framework of the study is illustrated in Fig. 26. In the first phase, a CP-based scheduling optimization model is developed with respect to changeable material logistics and limited man-power availability.

The crew resource demand and availability, predetermined deadlines, precedence relations between activities, material needs, and supply are used to derive a schedule in the study.

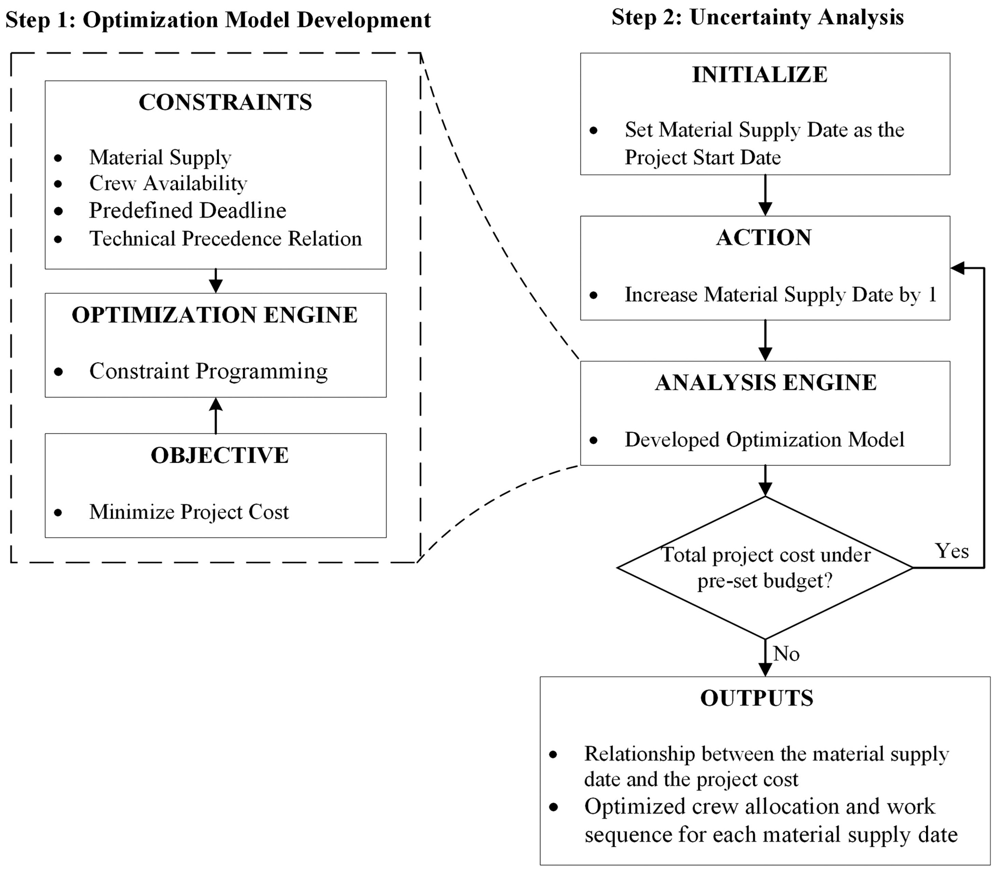
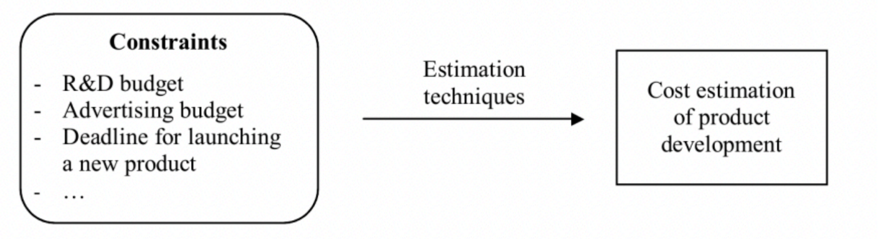


Figure 26: CP support for optimization

It can be seen that CP is used as optimization engine to find optimal schedule minimizing the total project cost in terms of crew and material inventory cost. In a general sense, optimization engines balance trade-offs and project/business constraints to determine the best possible resource allocation, which provide the most efficient resource utilization and the minimum project cost. Moreover, plans and schedules are automatically prepared, and which increases operational efficiency considering uncertain material deliveries on the project schedule. In the step two, the alternative plans are derived based on the developed mathematical model in the first step. In such a system, the alternative material delivery scenarios can help project managers to see relationships between the overall project cost and material supply dates because for each particular case subject to material delivery and team availability, an optimized schedule is presented by the proposed model. Moreover, new constraints can be added to the scheduling optimization model to reflect project holistically which can be even related with problematic weather conditions that affect progress of schedule or the availability of engineering analysis or drawings.

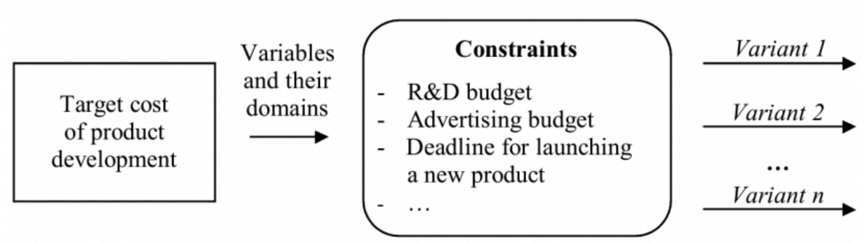
As a Feasibility Model

The article of Relich and Swic [75] is employed a prototyping problem of a product development for finding the project completion variants within the company's resources and project requirements. The problem is modeled as a constraint satisfaction problem which searched possible alternatives to fit target production cost. The reason of using CP in the proposed approach is that current decision support solutions are able to estimate the costs and compare the estimations with target, however they are insufficient to search possible variants within the cost limits.



(a) Traditional approach for product development

The drawback of traditional project evaluation is illustrated in Fig. 27a. Due to traditional approach, the projects that exceed the cost threshold is rejected as a consequence of the estimation.



(b) CP supported various product development

Figure 27: Product development approaches

On the contrary, first setting a target and specifying constraints allows to decision maker to identify alternatives which meets project objectives, budget, human resources, machines, etc. The issue solved with using CSP to formulate project prototyping problem and CP techniques to implement it.

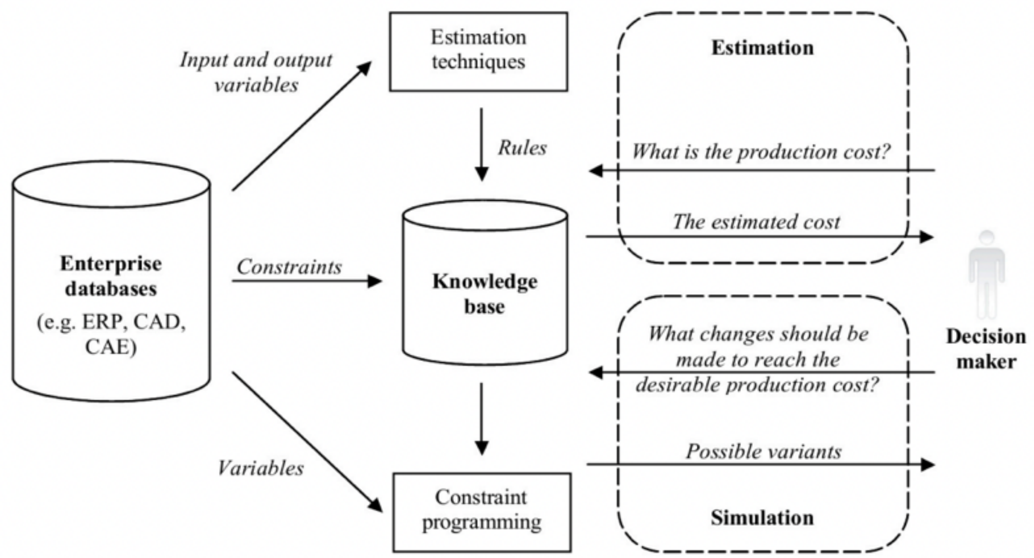


Figure 28: A framework for the proposed decision support system

Fig. 28 represents the framework of the system which consists of estimation and simulation parts. The relationships between variables are identified and used for cost estimation in the estimation part and then alternatives are searched that fit the desired limit and satisfy constraints in the simulation section. CP approaches used to reduce the search space and test the possibility of reaching the desired conditions.

Under the project constraints, an important question is asked by decision makers; which modifications and changes should be done not to exceed limits. The limits could be project objectives, budget, human resources, machines, date and so on, which are specific to projects and may vary through the life cycle of the project. The question can be answered with the results of CSP, which express the values should the variables have to reach for the desirable level of the cost. Constraint programming plays a critical role in the decision support system to find admissible solutions with effective search strategies.

1.4.5. Elements of a CSP

a) Constraints

In every aspect of the life, constraints appear in many different forms. The constraints can be conditions, properties, laws, restrictions or requirements. The work starts at 09.00 a.m. The shoes are sold in pairs. The component has a tolerance of 0.10 mm. We found ourselves within these boundaries that could be financial, social, physical etc. Although in theory of constraint, the main aim is to elevate the constraints for sake of organizations and people, the process of eliminating constraint comes with trade-offs and could be impossible depending on the context

- that is mentioned in Section 1.1.2. Project Constraints. Therefore, the art of living within the constraints must be learned in projects to keep all stakeholders satisfied.

Constraints limit the possible values of variables among the several unknowns considering the logical relations between multiple variables. For example, in a Euclidean space, *the sum of interior angles of a triangle equals the straight angle 180 degrees.* Without explicitly specifying the angle of each corner, some values are restricted in the infinite solution domain. In that sense, one can understand each angle can take a value in a domain of <0,180>. Moreover, one can limit solution space further with additional constraints as *all three sides of the triangle have the same length* that brings a clear solution and help to specify all unknowns exactly.

Moreover, they carry different features [7] [32]:

- Constraints carry *partial information* that limit the solution domain – it does not need to define the value of its variables but still guide the solution. For example, an arithmetic constraint can limit but not exactly specify the variable X such as X>1 which means X can be 2,3, etc.
- Constraints can be *added* not only in modelling phase but also to constraint propagation algorithm or even after a local inconsistency - an assignment set that is not contained in any solution. Moreover, their sequence does not create a difference. Only the conjunction of constraints matters in solution space.
- Constraints are *non-directional, t*he sequence of variables used to deduce a constraint does not matter. In other word, one constraint on several variables can be used to infer the constraint. can be used therefore the relation of variables, let us say X or Y, can be utilized first to deduce a constraint. For example, X=Y-1 can be used to declare the variable X considering the X← Y-1 relation or Y can be found using Y ← X +1 relation.
- In general, due to nature of complexity of the problems, constraints are frequently *dependent* to each other in a specific constraint store therefore, a set of constraints must be satisfied for a solution.
- Constraints can relate variables which are belong to different domains. Therefore, they can be present *heterogeneous* nature.
- Constraints have *declarative nature* which describes the desired end results without stating the control flow.

Constraints can be used in several different form to present the consistent or inconsistent value combinations. These can be extensional, arithmetic (=, >, ≠, …), logical (∧, ⇒, …) or global constraints etc. The typical algebraic constraints such as inequalities and equations which could be linear or non-linear are called as *arithmetic constraints.* Some constraints can contain direct domain restrictions on the variables. In addition to these, special symbolic constraints can be defined in constraint programming, which are called as global constraints [37, Ch. 1].

In constraint programming, constraints are not only check the feasibility, but also make pruning which excludes values from the domain unless the values come out in any solution of the constraint. Pruning can be summarized as removing impossible values from the domains. Therefore, constraints used to check the validity of a solution and deduce contradictor constraints and detect inconsistencies. The Fig. 29 represents that in traditional programming languages, constraints are only used for determining to the validity of a solution. On the contrary, constraints are able to detect inconsistent values of a variable domain quickly and infer new constraint in a constructive mode of CP [102].

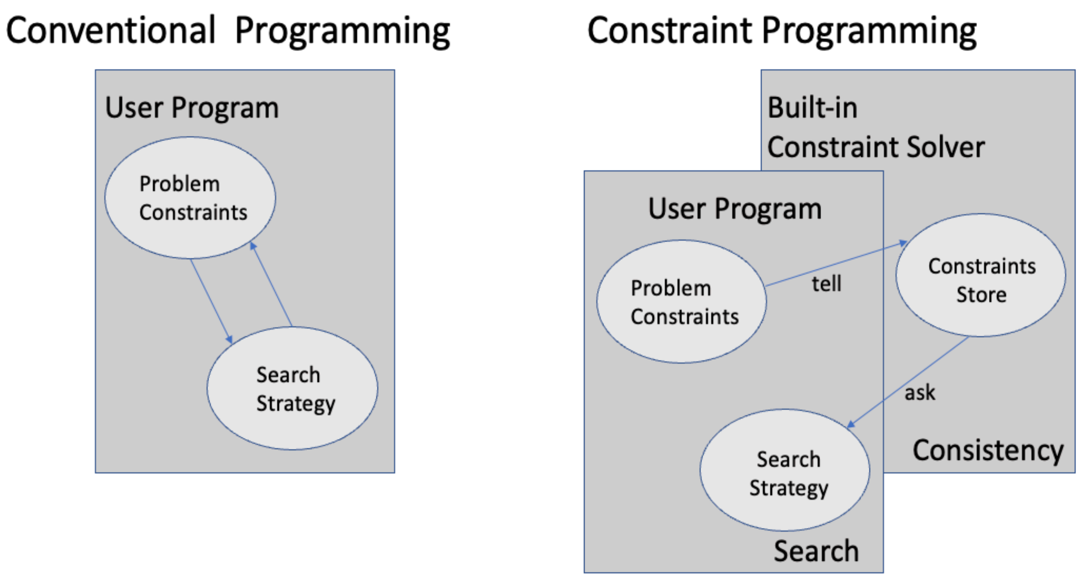


Figure 29: Constraint function for conventional programming versus CP

Global constraints

Global constraints show a relation between non-fixed number of variables $x_1, x_2, \dots x_n$. In other word, it confines a combinatorial structure over an arbitrary set of variables. The one of the main examples is *alldifferent*$[(x_1, x_2, \dots, x_n)]$ for a global constraint, which ensures each variables take the pairwise distinct values. In other words, global constraints group a system of inequations $x_i \neq x_j$, for all $1 \leq i < j \leq n$ into one *alldifferent* constraint involving all these constraints together. For instance, instead of expressing a number of simple constraints as $x_1 \neq x_2, x_2 \neq x_3, x_1 \neq x_3$, a global constraint can be defined. More global constraints can be defined according to a problem and model.

The *alldifferent* constraint is commonly used in real life problems such as rostering, resource allocation because it ensures that in the time and place, two things cannot exist such as in a basketball court two match cannot be done at the same moment. Another global constraint, *cumulative* constraint is used generally for resource constraint scheduling problems because it allows to define total amount of available resources, variables for the start and duration of tasks and the variable of the number of resources that is used by each task. In such a way that an overload for a resource capacity is prevented in these problems because it considers all the resource usage of the tasks at an instance. Likely, expressing the situation without any special constraints would require the large number of variables and constraints. Last but not least, *diff-n* constraint can be used for multi-dimensional placement problems or *cycle* constraint for complex vehicle routing problems to ensure each node is visited only once. The practical usage of global constraints can be reviewed in the applied case study which is in the further chapter. The more global constraints and the syntax documentation of the languages can be found in [36].

The global constraints improve the declarative expressivity and enhance the operational efficiency. They are considered as one of the main strengths of constraint programming as global constraints allow to the propagation algorithms to find a solution easily without creating a burden on the developer by calling already existing model to carry out the expressed function. These simple one-word constraints have exhausted and purpose-wise codes insides. According to the application and the type of problem, there are several global constraints such as for sequencing, cumulating, summing, sizing and so on. The model which will be discussed in the next chapter to solve an optimization problem have further examples for global constraints.

As another classification, constraints can be either hard or soft constraints [7, Cp. 9].

Soft Constraints

The constraints can be classified as hard and soft constraints according to their strictness to realization. The hard constraints must be satisfied necessarily, and they are not possible to negotiate on. Therefore, it can decrease the likelihood to find a solution to the problem. On the other hand, soft constraints can be violated and negotiable if necessary although it is desired that satisfy as much as possible. Any preference or wish can be specified with soft constraints such as shift or days off preferences among workers. In such a case, it causes a penalty in the objective function because the cost of violation of a constraint is given as quantifiable value. As a consequence, the violations affect the quality of the solution. Moreover, a function of the decision variables can include both hard and soft constraint in the same model. For instance, an employee can work at max, hard constraint upper limit, 4 hours for overtime although the ideally can be determined as 2 hours with a goal.

In general CSP, feasible solutions are sought for a given problem. Therefore, it is not possible to put in CSP directly to the over-constrained problems, in such a case a solution cannot be found. To overcome the problem, there are some techniques which use soft constraints. In that manner, the soft constraints are not obligatory and can be violated.

b) Variables

The decision variables are the unknowns that are expected to be found as CSP solution. Each variable has a set of possible values, which is domain. The decision variables can be either discrete or continuous in theory. However, in practice, CP models can work with continuous expressions but not with decision variables. The discrete variables could have infinite domains and finite domains. However, finite domain CP has only discrete variables (but a larger variety of constraints). For instance, the starting date is of each task is a discrete variable in a scheduling problem. However, the temperature of a room is a continuous variable. More examples can be given as binary, integers, objects, strings, Boolean (True/False) etc. based on the problem context which are not necessarily defined explicitly because those are declared by the domain-function [7], [51].

c) Domain

The domain of a decision variable is a set of possible values which can be taken by the variable. They are typically a finite set or an interval of real numbers. Some common domains that are *generally* used in CP can be separated as discrete and finite domains.

First of all, a finite set is even could be any type of symbol, so a domain is not limited with numbers. This fact provides an advantage to CP over other programming paradigms in terms of modelling alternatives. As another finite domain type, CP can have a Boolean domain in such a case the variables can be true or false. The Boolean satisfiability problems, which is known as 3SAT problems, is NP-complete. Unlike the infinite domains, the discrete domains can have one state for multiple variables such as the set of integers, strings.

The continuous domains, which is knowns as a set of values that includes all numbers in an interval have continuous states expressing one domain for one particular variable. They are very common especially for cases that require precise timing and to describe real world problems. Linear programming problems are the best know class of continuous-domain CSPs [7], [51].

1.4.6. Modelling with Constraint Programming

In the first phase, the problem is modelled with decision variables and their constraints. Without concerning about the steps of the solution, the user defines the problem by a set of variables, domains which is set of possible values for variables and constraints for restricting combination of values. In constraint programming domain, combinatorial problems modelled as constraint satisfaction problems.

The models which are designed based on problem is an important step for the solution because some models can affect the performance of the solution. For example, adding some constraints to the model formulation which could be redundant and create no change on the set of solution of the problem help to the solution procedure which particularly improving constraint propagation. Moreover, the reformulation of problem depends on the available constraints in the language or even in some cases, it could be not possible to model the problem.

In constraint programming, the user states constraints declaratively. Declarative approach is a programming paradigm which describes the desired end results without stating the control flow. It implements algorithms without explicit intermediate instructions like commands or steps that must be followed. In a nutshell, the user specifies what needs to be done, and not how to do it. On the contrary, the imperative programming approach, that is also known as command-based paradigm, is concerned with solution path, steps, and its sequences to execute in order to reach the result.

1.4.7. Solving of a CSP

There is a number of different approaches to solve constraint satisfaction problems [7]. The overall solution techniques are to enumerate potential solutions of constraint store to identify feasible solution to the problem. Always, a CSP can be solved with standard *backtracking* algorithms that is a general search strategy which has been commonly used for problem solving - in some cases, with poor performance. It systematically tests and directly searches the possibilities for finding the solution. In other words, different possible combinations are tries until the foundation of a complete solution. However, not learning from the failure of different nodes leads to inefficiencies. In regards, it requires to support it with other techniques like Branch and Bound (B&B) in order to solve larger problems. Other than systematic search method B&B, local search algorithms can be used such as hill climbing, simulated annealing, genetic algorithms, beam search which are also known as metaheuristics techniques. In other to select the proper algorithms subject to CSP characteristics, the table can be checked from Annex - A.2 [78]. To increase the performance of solutions to constraint satisfaction issues, heuristics can be applied because it can be computed with less effort during the search and guide the search algorithm.

Some of the approaches use *constraint propagation* which is an inference method for removing the inconsistent variable values to make the original problem simpler and search space smaller. By way of explanation, reducing the search space and trying to mark partial solutions by continuously assign unlike values to different subsets of CSP variables is the main working principle of constraint propagation which is an effective way of implementing constraint technology. The idea behind the constraint propagation is based on *local consistency* such as node, arc, path, k-consistency.

There are two primary types of use cases of constraint propagation; first is as a preliminary task; before attempting to find solutions for instance arc or path consistency preprocessing and the second is as buried inside the standard code of search for example *forward checking* or maintaining arc-consistency algorithms. These two are important filtering techniques which prune the domain of unassigned variables in advance to avoid unnecessary backtracking. Either use case increases performance dramatically with the successful elimination of inconsistent values. In other word, forward checking prevents assignments which fail guarantee in later stages. Furthermore, constraint propagation is straight methodology in order to automate every possible outcome of a variation in the original model. Also, even, depending on how much constraint propagation done and the problem, solutions *may* be reached without a need of search [37].

It is critical to be able to link CSP solution strategies to problem-specific attributes as this helps selecting the best appropriate techniques for a given problem. The solving techniques have some strengths and weaknesses compared to each other therefore the selections of which algorithm or heuristic to be used are important decisions. However, nowadays, the off-the-shelf CP solvers provide an effective implementation for the required solution techniques without creating any burden to the user for solving the model. Therefore, for the sake of simplicity and the topic of the thesis, the points that belongs to the solution part of the CP are only mentioned.

1.4.8. Constraint Programming Systems and Languages

This section presents some popular CP systems and languages are employed by constraint community for tackling complex problems. As a one of the significant advantages of constraint solvers is that provide several built-in strategies and algorithms to solve constraint models. They offer constraint propagation, search, and a range of additional approaches.

As a market leader role among commercial constraint programming software, IBM ILOG CP Optimizer provides a CP engine targeting constraint satisfaction problems (especially for scheduling) as well as optimization problems. The model and run system which is illustrated in Fig. 30 can operate without a need of writing and keeping a search strategy by a user [7]. It is possible to define new constraints and new search algorithms. However, it is rarely applied in an industrial situation thanks to the efficiency, robustness and flexibility of the automatic search and the expressiveness of the modeling concepts. In that way, it is quite productive and riskless with respect to hand-crafted constraints and search techniques in the industrial context. For example, CP Optimizer model which is solved with automatic search won the Industrial Modeling Competition at CP 2015 [85].

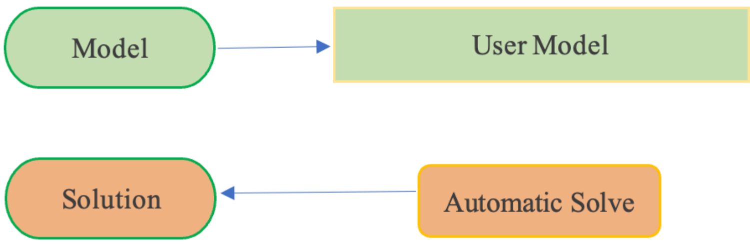


Figure 30: Model and run approach

Also, one of the biggest competitor Google comes up with open-sourced OR-Tools solvers for CP such as CP-SAT solver and Original CP solver which was built by OR team on Google. The tools provide software libraries and application programming interface which is a connection between computers or programs via data transmission. The library use can use Python, Java, C++ and NET library. Similarly, MiniZinc is an open source and free constraint modeling language to model CSP and optimization problems independent from the solver which means that the user can run the model with different solvers such as Google's and Gecode (Generic Constraint Development Environment) which is another software library to solve CSP and it is programmed in C++ [82]. Last but not least, LINGO is an integrated tool to modelling and solving problems which provide a powerful modeling language, suitable environment to build and edit problems and built-in solvers for linear, non-linear, quadratic, constraint, integer optimizations etc. [81].

For more, the Table 4 lists some CP systems with their corresponding information which are gathered by Bockmayr and Hooker [37, Ch. 5].

Table 4: CP Systems and their aspects

System	Availability	Constraints	Language	Web site
B-prolog	Commercial	Finite domain	Prolog	www.probp.com
CHIP	Commercial	Finite domain, Boolean, Linear rational, Hybrid	Prolog, C, C++	www.cosytec.com
Choco	Free	Finite domain	Claire	www.choco-constraints.net
Eclipse	Free for nonprofit	Finite domain, Hybrid	Prolog	www.icparc.ic.ac.uk/eclipse/
GNU Prolog	Free	Finite domain	Prolog	gnu-prolog.inria.fr
IF/Prolog	Commercial	Finite domain Boolean, Linear arithmetic	Prolog	www.ifcomputer.co.jp
ILOG	Commercial	Finite domain, Hybrid	C++, Java	www.ilog.com
NCL	Commercial	Finite domain		www.enginest.com
Mozart	Free	Finite domain	Oz	www.mozart-oz.org
Prolog IV	Commercial	Finite domain, Linear/nonlinear interval arithmetic	Prolog	prologianet.univ-mrs.fr
Sicstus	Commercial	Finite domain, Boolean, linear real/rational	Prolog	www.sics.se/sicstus/

1.4.9. Constraint Programming and Artificial Intelligence

A large variety of problems in Artificial Intelligence can be viewed as a special case of the constraint satisfaction problems which can be represented in the form of searching in a state graph or can be modelled as CSP according to Nadel [2]. Also, Eugene Freuder says "The user states the problem, the computer solves it." for CP in her Constraints book. In the light of these words, CP is considered as AI. A very important point is that it supports and automates decisions with providing methods for complex solving problems that cannot be tackled without the application of intelligence.

Search is one of the mostly used technique in AI and CP to solve a problem. Exact search techniques can be used to find an optimal solution in a guaranteed way by investigating the complete problem space. CP systems are mainly based on enumeration methods such as depth-first search (DFS) which explore the search tree in one branch downly [7]. As an example, DFS can be seen on the solving of the cryptarithmetic example on Fig. 25. On the contrary, heuristic search methods are used for problem solving within a reasonable time limit or memory space. The solution is not necessarily the optimal one, but approximate among the available options. Several AI techniques such as A*, IDA*, Alpha-Beta pruning, constraint satisfaction is belonged to heuristic search category. Both AI and CP can use heuristics to find satisfactory solutions that is illustrated in Fig. 31.

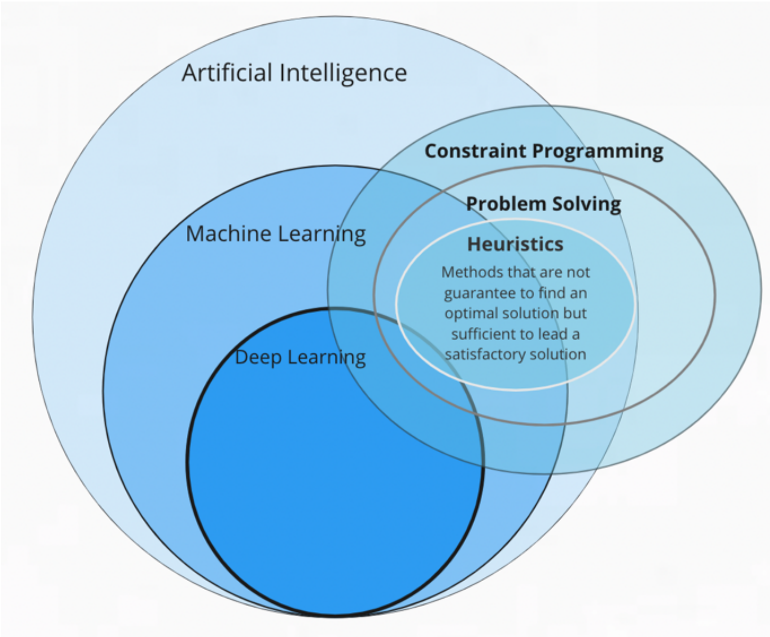


Figure 31: AI and CP relationships for problem solving

Lee et al [67] declare that CP integrates the complementary power of domain-specific knowledge representation of artificial intelligence and structured mathematical computation of operations research. Constraint programming does not only use a broad variety of methods from artificial intelligence, databases, programming languages but also enriches them. Even, for example, the study "Constraint Programming meets Data Mining and Machine Learning" [80] represents how CP and ML both get and bring benefits to each other such as how CP techniques are applied to learning and pattern mining problems and how to improve the performance of the constraints solvers with ML or can modelling part of CP be automatized.

1.4.10. Constraint Programming and Project Management

The problem solving with constraint are not only concerned on project management area but also computational systems and so on. For the industrial point of view, combinatorial problems are frequently faced in various different application areas. CSP is used for representing and solving these problems. In the computational perspective, the problems are belonged to NP-hard class which indicate there is not generally known and efficient algorithm. These NP-hard problems include feasibility problems and also could be optimization problems. There are several motivations to use CP to solve problems that a project manager has to deal with like sequencing, scheduling, allocation, planning, routing, packing [53].

In some projects, project managers may be asked a feasible solution in a reasonable time because even finding a feasible alternative can be difficult considering the complexity and limitations of the project. Although it can be seen as a superficial motivation to use CP, it is based on feasibility which means finding a feasible solution that satisfy all the constraint of the problem instead of finding an optimal solution.

Also, unexpected and unpredicted events could happen in a project considering uncertainties on the project such as unforeseen machine failures, sudden team unavailability, delayed material deliveries. For example, such kind of events can cause a change on the schedule of the project if the scheduling system is not adapted to these unexpected situations.

Moreover, the size of the project is not stable which can enlarge. Due to scope creep, new constraints and objectives can evolve during the project life cycle that makes exceeds the capacity of manual human problem solving. When a dynamic condition of a project taken into consideration, the use of CP which enables to add new constraints and find potential solutions for the iterated version of the problem without affecting other parts of the constraint system. At that point, the credits can also be given to the declarative programming approach for the flexibility because it allows to modify the CP system.

Besides, Simonis [79] implies that user-friendly interfaces and systems that can cooperate with the user are required to solve problems and make decisions can be achieved with the use of constraint logic programming which is a class of constraint programming.

There are lots of parameters such as activities, their precedence, allocations, constraints and so on that affect the project and in return, the project's problems such as scheduling, resource allocations, routing. In such a problem size, the complexity of the problem exceeds the human ability of problem solving. For example, crashing an activity network could be deficient and hard to apply to large and complex projects. The calculation time and effort increase steeply which means the solving of this problem requires beyond the capacity of a human. Therefore, an intelligent mechanism such as the ones with constraint programming which seeks solutions for the variables that satisfy all the existing constraints in a reasonable effort can play a central role for a project.

Constraint satisfaction is one of the AI technologies that opened a way to apply it into industrial applications [7]. In the following section, some industrial application examples are given to witness real world applications of CP.

1.4.11. Common Problems in Constraint Programming and PM

Optimization

The term "optimization" can be applied to a wide range of situations. Some of the common applications are listed in the Table 5 under their application domains [84]. It can be seen that most of the optimization applications build best possible plan or schedule using mathematics and logic. The planning and scheduling problems will be also mentioned in the following titles.

Table 5: Optimization applications domain

Manufacturing	Transportation and Logistics	Financial Services	Utilities and Natural Resources	Telecom	Multiple and Other
•Inventory optimization •Supply chain network design •Production planning •Detailed scheduling •Shipment planning •Truck loading •Maintenance scheduling	•Depot/warehouse location •Fleet assignment •Network design •Vehicle and container loading •Vehicle routing and delivery scheduling •Yard, crew, driver and maintenance scheduling •Inventory optimization	•Portfolio optimization and rebalancing •Portfolio in-kinding •Trade crossing •Loan pooling •Product/price recommendations	•Supply portfolio planning •Power generation scheduling •Distribution planning •Water reservoir management •Mine operations •Timber harvesting	•Network capacity planning •Routing •Adaptive network configuration •Antenna and concentrator location •Equipment and service configuration	•Workforce scheduling •Advertising scheduling •Marketing campaign optimization •Revenue/Yield management •Appointment and field service scheduling •Combinatorial auctions for procurement

A project problem can be solved with an optimization model that carries a bundle of inputs such as needs, costs, preferences, constraints, resources, assumptions, objectives. In the model, these can be specified with equations, matrices and logic statements and these are executed by the optimization engine(s) that can employ mathematical programming, CP or both. Considering a project environment, the elements of model can include the followings:

Available resources: Project resources are defined as machines, tools, personnel with required skills and expertise, customers, inventories and so forth. It can be summarized as anything needed to complete the project.

Operating constraints: These constraints can be separated as globally and individually. For example, as a customer specific constraint, firm A can accept deliveries only on Monday. Or staff X wants to take off after 2:00 p.m. for a week.

On the other hand, global constraints can be related with the policies about working hour limit of an employee, the construction site is 500 kilometers away from the production facility.

Goals: Each project has goals, and it is better to express them analytically to plan and achieve the objectives. Maximizing revenue, increasing key performance indicators, improving customer service/satisfaction, minimizing costs can be given as set of goals of the project.

Project needs: It implies the needs to be fulfilled and services to be carried out. In particular, products to be built, museums to be renovated, processes to be implemented.

Costs: Project costs are the expenses incurred for conducting project activities to complete it. The costs can be categorized as direct, indirect, fixed, variable and sunk. The examples are supplied material or service costs, hourly labor costs which may be extra for outsourcing, inventory holding costs, etc.

Assumptions: The model is formulated based on some inputs that are accepted as true and taken for granted. For instance, the marketing campaign will benefit to sales of a product, the environmental conditions will be suitable for construction works. These are also considered as project uncertainties and risks.

It must be pointed out this stage that when the problem-solving concept is evaluated in the project context, it is not always possible to have many solutions to be selected as the best one. The reason is that there are several operating constraints and interdependencies between project activities. In such a case, reaching to feasible solution rather than the optimal one is very valuable enough for the project. CP technology was developed to deal with such intricate problems. As a priority, it finds feasibility and then tries to refine the solution until it gets optimal. Therefore, it can help project managers to satisfy project team, meet customer demands, and increase profitability.

Planning

Planning deals with finding a correct sequence of operations from initial state into goal state. In management point of view, it has a broader meaning because it includes defining and refining the scope, objectives, risk, procurements, communications, resources and all aspects of the project and actions to be tracked to achieve the desired objectives. In artificial intelligence, since its beginning, it is studied as the way of accomplishing certain predefined objectives by identifying and eventually performing required tasks and to do so that can be performed by robots or computer programs [83]. Moreover, it is a critical aspect of sequential decision making and rational behaviors in a way that plan specify the AI what to do.

In general, the formulation of a planning problem can be described with three elements which are initial state and anticipated changes, agent's goal and possible events that can be realized. A feasible chain of actions is introduced as plan considering the initial state, expected changes and a course of actions in order to reach desired state like Fig. 32.

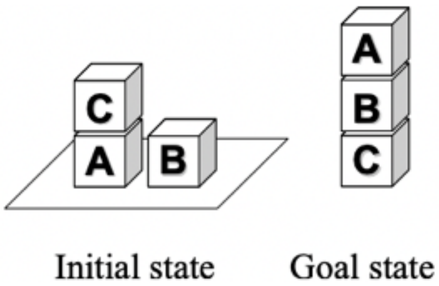


Figure 32: A planning problem

The discipline of constraint-based planning investigates how to address planning challenges using CP. The set of activities to be scheduled and constraints are decided in the planning part of the project, which cause an additional complexity for planning. Baptiste [7, Ch. 22] imply that CP in planning is less than scheduling. However, the strengths of CP are valid both in planning and scheduling that having specific heuristics with tree search, defining global constraints, propagation of temporal and resource constraints, a natural and flexible modelling of problems.

Planning problems carry several constraints. These can be;

- Resource constraints referred to some amount of resource that is needed to realize that activities such as machine, space, personnel, energy.
- Temporal constraints express precedence relations between activities. They can be qualitative such that between two activities A_1 and A_2 stating that A_1 is to be started before A_2 or quantitative A_1 starts in a specified time internal.

There can be some domain-dependent constraints. For example, if one activity uses a specific resource, none of the activities can use this specific resource that express interdependencies of resource allocations. Another commonly found case in a project is priority among resources like if a product is available from a specific supplier, it should be procured from there. The constraints can be extended according to project requirements.

Resource Allocation and Staff Assignment

Resource management is a process that is highly iterative and complicated to implement. To execute a work which requires time and effort, it is needed to assign resources to the work packages. The aim is to assign a variety of resources to the activities which belong to restricted domains such as by a set of rules and restrictions. Project managers are able to adjust a resource allocation model or schedule with resource optimization techniques. For example, resource smoothing ensure that the project do not go beyond the limit of redefined resources base on timely constraints in a way that activities can only be postponed within their free and total float time. Alternatively, based on resource constraints, resource levelling can be carried out to visualize where the team has flexibilities in order to move the resources when needed and where we can delay the activity as long as they are not on the critical path.

One of the most important type of allocations in the project management is staff assignment. Finding correct activities for the appropriate members of the project team is important decision in project management. A good match between personnel and task requirements has an impact on the performance of the project and organization. The role and responsibilities of the team members must be clearly specified by project managers considering their different competencies such as skills and seniorities, preferences and availabilities. Moreover, some constraints can be emerged from regulations, contracts or technical limits.

Scheduling

The one of the most important concepts of PM, scheduling is a branch of AI and OR both. As NP-complete problem, it has been studied by AI community since 80s and the problem of scheduling activities is a discrete optimization problem [56], [83]. In a scheduling problem, the variables are the project activities that are needed to be scheduled. According to the situation, it could be courses in a university, or a job in a job-shop. The domains are the various combinations of times and places for each activity such as start/end times of courses, machines in a job-shop for each job. Moreover, the constraints must imply the activities that use the same limited resource cannot be overlapped, i.e., arranged at the same time in the same location. There can be more constraints depending on the case such as some particular activities cannot be scheduled in some location or time. Or another example, some activities cannot start before than others considering the precedence relations like in general manufacturing jobs must start earlier than assembly activities in a job shop.

Scheduling is also defined as the allocation of scarce resources over time. It is used timely completion of a project. It answers several critical project questions like which resource will be assigned, which activity will be realized, what is the most efficient time for them. There are various techniques for project scheduling. The CPM and PERT, also known as network models, are most commonly used mathematical methods by project managers. Specifically, using probabilistic model PERT is more suitable in projects which contain non repetitive events and uncertain activity time. In contrast, CPM is used to optimize schedule where time consumed by activities are exactly known. Moreover, critical chain method, what-if analysis, and resource optimization techniques are also used to develop a schedule. While network diagrams are useful to illustrate the general representation of the timeframe and logical interdependencies of the project, it can become complicated to develop and monitor constantly. Therefore, Gantt Charts are generally employed, as they present much clearer portrait of the project activities. Gantt chart refers to a well-known method of visualizing project tasks in a full illustration in order to show and highlight the timeline of the jobs required to be done with available resources for a predefined time interval in other words it is the big picture of 'where are we at now in the project' kind of tool.

Although AI community imply that planning focus on finding a sequence of actions from some initial state to a goal state whilst scheduling deals with on an optimal allocation considering time and resources, in project management perspective, these cannot be exactly separated. According PMBOK [8], planning schedule management and developing schedule is belong to project time management knowledge area and planning process group, that the further details can be found under the project schedule management title.

Constraint programming practices are helpful to deal with scheduling problem solving and this demonstrates a good example about the possible application areas of CP. Constraint based scheduling (CBS) has become one of the most prominent application fields of CP throughout the years. Its expressiveness enables handling complex scheduling applications which could consist non-mandatory activities, over constraints, non-regular objectives, etc. The one of the key sources of the success is the interest to scheduling both by AI and OR communities. AI approach tries to focus more general scheduling models in order to solve with general problem-solving paradigms therefore its performance on specific cases can be low. On the contrary, OR tends to get high level of efficiency in its algorithms. They use classical statistical and mathematical models that might neglect degrees of freedom and side constraints which turn the original problem to the simpler form. It can cause impractical or straight-forward solutions. In short, neither pure approached of AI nor OR can deal with a real project scheduling problem effectively.

Fortunately, the integration of efficient OR algorithms and general modelling and problem-solving paradigm of CP contribute the performance for constraint-based problems.

As an additional consideration for this part, it is important to mention about the common solutions methods of different project scheduling problems mentioned in the previous sections of literature.

Scheduling: The base scheduling problems are mainly solved by CPM procedures which depict projects by "network diagrams where jobs or activities are represented by arcs, events are represented by nodes, and inter-relations between the jobs or activities are defined by network structure." CPM can be implemented by hand or with the help of a LP software developed from the CPM concept, tools like Microsoft Excel or others can also do the same, for more complicated diagrams. Another useful technique to deal with these kinds of problems is the Gantt chart which demonstrates a clear representation of the overall project timeline yet is not decent enough to visualize the interrelationships between project tasks. Furthermore, PERT is another less frequently practiced method to deal with scheduling which is for stochastic problems rather than directly finding the minimum time needed to complete the project, it gives an interval of time to finalize the project within a defined probability [97].

Scheduling with Crashing: When crashing is introduced to the scheduling problems, Gantt charts become more or less useless to come up with a solution but instead the common practices for time/cost trade off problems are CPM, LP and CP. Similar to the base problem, CPM provides a network diagram and additionally crashes equations in order to find an optimal solution manually. LP is also likewise formulated from CPM logic and can be beneficial to solve problems in complex settings. CP is another method, not regularly, to tackle these kinds of problems with built-in functions.

Scheduling with Crashing under Resource Constraint: For scheduling with crashing under resource constraint, CPM and LP are not sufficient to get an optimal answer since the extra resource constraint from the former adds up to the complexity of the problem and turns the problem into NP-hard [56] and hand calculation by CPM is not feasible and LP is not constructed in that way. MIP is an approach to manage the resource constraint since it can be modelled with a binary system and use branch-and-bound type methods. Similarly, CP is employed in order to find an optimal solution as well [103], [58].

As mentioned in the sub-section tools and software in PM, there are several software packages especially for scheduling. However, the software packages which keep several rules and algorithms inside can produce remarkably different outputs, i.e., schedules, using the same data. Not only among the packages but also existing theoretical techniques give different estimations for the completion of project due dates. Trautmann and Baumann (2009) [59] analyzed 1560 projects from internationally recognized benchmark Project Scheduling Problem Library PSPLIB and recognized that some commonly used scheduling packages such as MS Project, Primavera P6, Turbo Project Professional which generally use priority rule based heuristic algorithms are not able to give the best-known feasible schedules. In other words, the analysis showed that the software packages compute the project duration significantly longer for resource-constrained project scheduling problems. Therefore, project managers should not trust only the results they get and try other rules or even software.

It should be noted that there are other solution methods to scheduling problems too but in the scope of this thesis the ones mentioned above are the most relevant.

1.4.12. Industrial Applications with Constraint Programming

Nowadays, CP has a lot of application area in industry and businesses such as manufacturing, transportation, health care, advertising, telecommunications, financial services, energy and utilities, as well as marketing and sales. There are several firms that use constraint programming technologies to optimize their processes such as American Express, BMW, Danone, eBay, General Electric, Porsche, Shell, Visa, Wal-Mart, Xerox, Yves Rocher, Zurich Insurance and so on [7]. In this section, some of the examples are given to emphasis the application areas of constraint technology which project managers can face with in a project context.

ORBIS-Dienstplan - Personnel allocation

Personnel planning problems such as university timetabling, rostering for hospitals and long-term staff planning are the ones of the common constraint programming applications. As an application of a timetabling of staff, a nurse scheduling system is exemplified in this section which is illustrated on Fig. 33 (Meyer auf'm Hofe, 1997) [86]. Nurse rostering which means the allocation of nurses to stages of work over time has drawn attention of expertise from OR and AI for many years. The size of the problems tends to be large and complex. The resource, logical and legal restrictions regulate the working pattern. For instance, after night shift of a nurse, a morning shift should not be assigned. Or, the maximum working time should obey the legal laws. In addition to, the several preferences of team members create conflicts which are expressed as soft constraints and may be violated although it causes a displeasure for the owner of the preference. In such a context, arranging the best workforce schedules are quite fruitful for the employees' performance and satisfaction, and thus improve the operational performance of the project. The ORBIS-Dienstplan system tackles complex problems in hospitals and fire departments. It has declared that it was using in sixty hospitals in Germany for nurse rostering system in the beginning of 20's. The systems problems are represented as (partial) constraint satisfaction or constraint optimization problems because considering the realistic situations, it is almost impossible to satisfy all the constraints in a rostering problem. The software is quite interactive, and the user can change the result of the algorithms. The automated nurse rostering system use a heuristic local search with CP techniques.

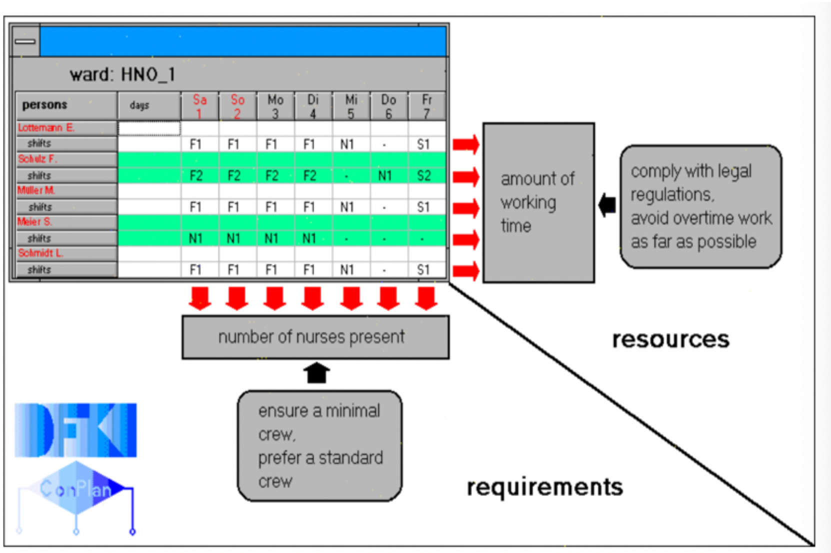


Figure 33: The schema of the requirements in nurse scheduling

With progress and requirements, the system enabled to define fuzzy constraints which allow to partial satisfying and violation of constraints as well as hierarchy levels and constraint weights to give priority for specific constraints for reflecting their importance. For example, the system can give priority to legal regulations, personnel cost, employee's requests, guarantee to minimum crew size, etc. Moreover, staffing levels (overstaffing, understaffing in crew size) can be defined as fuzzy constraints and the penalties can be applied to assign priorities to the system. In addition to, constraint processing improved the flexibility of the automatic rostering system so it can be adapted to different situations and requirements which can vary for different hospitals [87].

Tact- A transport planning and scheduling system

Even in early days (1999), constraint programming is used to develop a decision support system to solve transportation and scheduling problem in the food industry which is illustrated in Fig. 34 [79].

The TACT (Transport Application for Chickens and Turkeys) is designed for the specific needs of a conglomerate operating in food industry in the United Kingdom. The system deal with an integrated problem of logistics of raw goods to various processing factories using several resources such as drivers, lorries, other physical resources and material preparation team. The problem contains some further constraints. For example, the chicken and turkeys should be collected in a certain age and delivered to the processing factories.

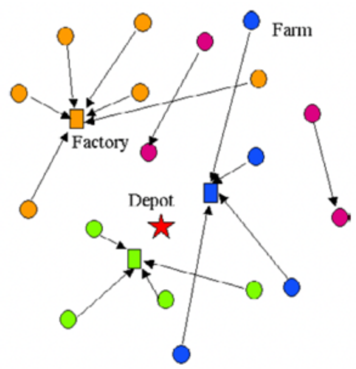


Figure 34: The simplified version of the problem

When the replaced manual system and TACT are compared, an operational efficiency can be seen such that the solutions are found in a quarter-hour instead of eight hours of work. Therefore, last minute changes can be avoided or compensated flexibly thanks to secured reaction time.

The TACT procedure can be utilized as a supporting mechanism to decision making instead of a direct problem-solving tool for scheduling type of managerial tasks. The power of using TACT is that the scheduling process accelerates considerably while complying with the constraints and instructions. In addition, it is also possible to try varied combination of constraints or choosing alternative heuristics, coming up with different solutions in order to find the suitable one.

Also, the company had face with new restrictions and some constraints were revised as a consequence of changes in the five years of operation. These updates were introduced to the system and implemented quickly to the constraint model without replacing the main architecture of the system.

Hybrid (CP&MP) Optimization Tools

IBM ILOG Studio offers some hybrid systems for optimization tools to the organizations commercially. For example, Advanced Production and Optimization (APO) tool by SAP is basically a comprehensive supply system planning application indeed it is a quite popular and prominent way of creating an effective and efficient supply chain operations benefiting from IBM ILOG CPLEX which use constraint programming and mathematical programming techniques. The application tries to find the best optimal solutions for supply chain activities in between producers, vendors, distributors, and customers. The model is optimized with respect to costs, constraints and goals which can be summarized as Table 6.

Up to a point, the packed optimization applications with a generic model are used effectively. However, if a project has prioritization rules, additional specific constraints, costs and goals, a custom application can be needed. It is possible to use tailored decision support applications based on custom models for a company, which are also using mathematical and constraint technology [84].

Table 6: Common costs, constraints and goals in an optimization model

Cost	Constraints	Goals
Production	Production capacity	Minimize costs
Inventory carrying	Transportation capacity	Maximize customer services
Transportation	Safety stock requirements	
Labor overtime	Product shelf life	
Non-delivery penalties	Lot sizes	
	Customer category priorities	
	Product Priorities	

2 Methodology

In the methodology part, the methods and ways utilized to perform the literature search, conduct the analysis and achieve the findings which are the fundamentals of this study have been presented, in detail. In this manner, what is done in the scope of this thesis from the very first steps of the motivations to start around the topic of AI and PM to the last where there are the obtained results from these can be found in this part. To do so, the main point explained is the research methodologies which discuss the breakdown of the investigations carried out on a case study to answer the research questions and validate the hypothesis.

2.1. Motivations

Although, several potential benefits and superiorities over the other methods, depending on the situation, have been mentioned in the literature review part. Some of the motivations to focus on this thesis topic have been highlighted with findings from academia and industry.

a) AI is Highly Used on Optimization and Decision Making

According to Mckinsey's "The State of AI in 2020" report [4], AI adopted highly in the optimization areas within the functions as seen on Fig. 35. Considering how artificial intelligence can support project management processes, the field of optimization comes to the stage because optimization is used and researched by AI and operation research communities. Many efficient optimization algorithms have been employed in the Artificial Intelligence area, which will be mentioned in previous sections. Moreover, optimizing project processes and resources are considered as important duties for management of a project. Especially, the use of optimization in scheduling and sequencing activities were found as very promising and applicable [30]. Moreover, in Harvard Business Review Article Collaborative Intelligence: Humans and AI Are Joining Forces (2018) identified five elements to improve business processes in the result of their work with hundreds of firms.

Figure 35: AI use cases most commonly adopted within business functions, %

These are the operational flexibility, speed, scale, decision making, and personalization of product and services. AI can boost all the characteristics by giving the right information on the right time [72]. Enhancing these promising topics which are closely related with finding the feasible solutions for the problems and deciding the optimal ones among alternatives have been examined.

b) Expectations From Project Managers

According to Scott-Young and Samson (2009) [44], the three-element represented in Fig. 36. are key for achieving project objectives successfully. Therefore, a project manager must have good knowledge to collaborate and implement the key dimensions. One of them requires the knowledge and skill to apply appropriate tools which enable an efficient organization and management to the project both for day-to-day decision and strategic direction. Not only Gantt charts, timesheets, Kanban boards but also some optimization and problem-solving software such as

CPLEX, Python, LINDO etc. are key tools that are expected from a project manager to know.

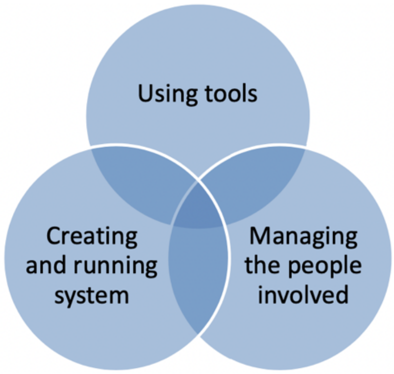


Figure 36: Dimensions for the successful project management

Moreover, in the study of Kwak and Anbari (2009) [47] which is also funded by the Project Management Institute, the evolution and trend of PM studies are analyzed based on top management journal articles (on AOM, IEEE, INFORMS, HBR and so on). This study reveals that Operation Research (OR) and Decision Science (DS) were the second primary and important research topic among allied disciplines of PM in the last 50 years. In other word, scholars and practioners have a strong interest to the concept of managing resources, maximizing profits, minimizing cost and optimizing the key variables such as schedule, allocations, risks of project performance.

c) Optimum Efficiency and Readability

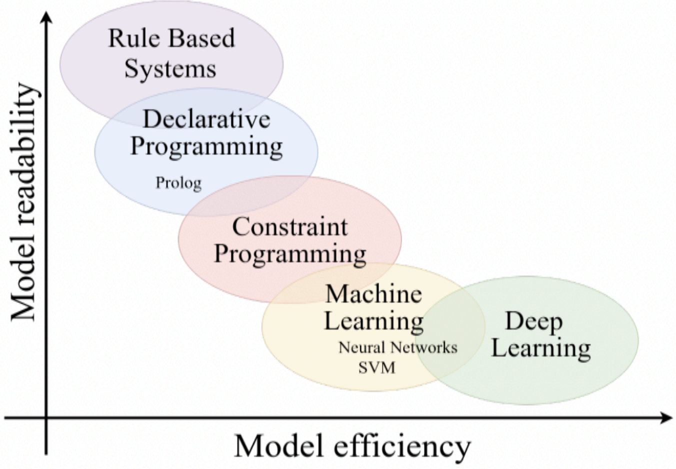


Figure 37: The efficiency and readability of AI technologies

Understandability, transparency, and interpretability are key concept in explainable AI. Constraint programming is placed on the optimum level in terms of efficiency and readability which represents performance and explainability, respectively, which can be seen on Fig. 37. The goal of explainable AI research is to meet the demand for reliable AI systems that justify their reasoning in a way that humans can understand and interpret. However, the lack of transparency on AI processes are rising and it is getting difficult, or could be impossible in black box systems, to understand why certain decisions are made because these systems utilize more advanced reasoning techniques and computational power [57].

For a project manager that utilized from AI, recognizing and knowing the process of the decisions made is critical for both verifying the system's reliability and monitoring biased or unfair decisions which could be taken by the system Moreover, considering the writer of the thesis who is a management engineering Master of Science student and prospective project manager, these features of the models are vital.

d) Less Challenging Implementation

Like indicated in Section 1.2.7., having a successful AI implementation in an organization can be difficult and inconvenient. Unlike, constraint programming is an easy-to-use paradigm with respect to machine learning, deep learning applications. Even, modelling the problems is enough for a CP system which has the automatic solving capabilities. Moreover, the programmer can modify the model of the problem according to the requirements which could change during the project lifecycle. Considering the elements of the problem, this comes to mean to alter the constraints and variables which can be accepted effortlessly. In addition to, CP has a declarative nature therefore it causes less errors without dealing with technical details and is easier to develop for non-programmers (which is important because the target group is project managers in the thesis) with respect to imperative languages.

The compilation 'When you should use Constraint Solvers instead of Machine Learning' by Antoine Champion [76] exhibits a pandemic situation which needs an algorithm to match infected people and hospitals. The solution of the case can be searched in deep learning and neural networks. However, the need of having large historical data also necessitates cleaning data, testing different architectures and training them. In other words, it takes lots of time and money to apply machine learning approaches. On the contrary, the problem can be solved with constraint programming, which whole process, modelling and solving, takes a couple of hours.

The readiness level of CP can also be seen as an advantage. The programming languages and solvers are ready to model problems such as MiniZinc, ECLIPSe, IBM ILOG CP. Moreover, it is not limited to the algorithm itself because the language and compatible solvers can be integrated with other technologies both software and hardware to create a complete solution. In short, decision makers should have knowledge and be conscious about technological tools and solutions. Instead of going around the horn, it is better to select the basics in some cases.

e) Analytical Problem-Solving Need

Apart from the hardness of the project problems to solve them with traditional ways, another discussion is about the perception. In some context, decision making and problem solving are considered as soft skills in project management. Nowadays, the several non-analytical ways of solving problems such as Plan-Do-Check-Act cycle, 8D methodology are examined and focused. The methods are well-developed and could provide the best answers to the teams. However, it is undeniable that the decisions taken by human's intuitive approach to the problem may have some issues in it. The heuristics and biases cause a person to view and evaluate a situation inaccurately. For example, sometimes, a project can struggle such as starts to fail delivering satisfactory results and continuously lose efforts and money without a progress. Even in such a case, a project manager could tend to cling to the project with some reasons such as denying the failure, being dwelled on the suck costs, or still expecting the good results too optimistically although freezing or closing of the project could be more profitable. Therefore, an analytical and logical problem-solving system can compete better than an emotional individual.

Regarding these concerns, constraint programming and the CPSs that can be solved with the constraint paradigm are selected as a topic of the thesis. These are the reasons why this thesis will examine the relationship between constraint programming and project management.

2.2. Aim and Objectives

After the literature review has been walked through, it is essential to state about the objectives of the study in the first section of the methodology. A well-developed and powerful concept is taken into consideration deliberately considering the nature of the project environment which includes dynamicity, limited resources, conditions, and etc. Considering that constraint programming is one of the artificial intelligence technologies that effectively applied into industrial applications. Constraint satisfaction or optimization is a way of automated reasoning and decision making where the problem is modelled with constraints and a set of variables. Further motivations that are clarified in previous section, constraint programming on project management is selected as the main theme of the study and related research questions are drafted accordingly.

The main research question which all the parts revolve around the first research question; '*How constraint programming can support to project management especially in limited resource and time frame?*' The research questions try to understand project management processes or problems which can be carried out effectively with CP. Although constraint programming can be successfully applicable to several domains, this study aims at understanding the existing and potential applications of CP for project management applications such as scheduling, planning, resource allocation which involve combinatorial problems. The constraint programming techniques are examined for modelling and solving a wide variety of constraint-based project's decisions. The real-life models and systems are particularly demonstrating the applications. The Fig. 38 represents the research questions and the structure of respective concepts.

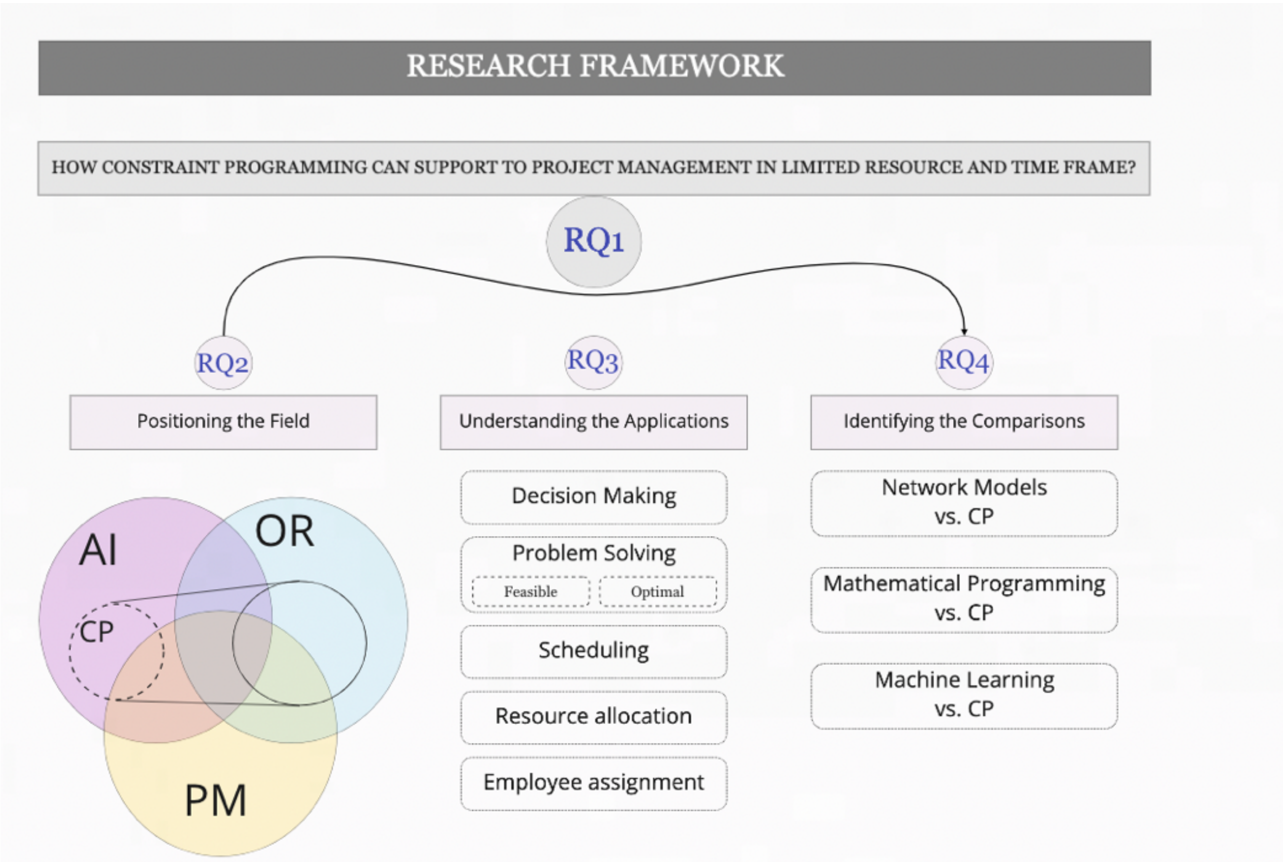


Figure 38: Overall research framework

The side studies are set as follows:

RQ.2: What are the relationships between OR, AI, CP and PM?

In the literature, there are several domains that work with CP. However, the relationships and contributions are quite interwoven. At one point, all of the mentioned disciplines try to solve questions and to find the feasible or if it is possible, optimal solution. Even, there are lots of controversies in the literature regarding relationships and definitions. Therefore, the research question research plan to highlight the connections among the OR, PM, AI and CP.

RQ.3: What kind of problems are solved, and decisions are made in the project context with constraint programming? Why is it promising?

Constraint programming can be utilized to solve several problems to find feasible and optimal options for various areas such as portfolio management, machine vision belief maintenance, new product development, DNA sequencing, etc. The thesis focuses on the problems that can arise in a project which has time and resource constraints. Therefore, scheduling, allocation and assignment problems are taken into the main consideration. The research question searches for the benefits, comparisons and promising features.

RQ.4: What are the advantages of Constraint programming over other techniques such as machine learning and mathematical programming?

There are several technologies such as process automations, deep learning, predictive analytics that can be applied to organizations for project management processes. However, it is highly crucial to understand the needs of the organization when a promising technology is introduced and implementation of it is planned. Moreover, the readiness level of the technology and the parties which make use of it should be considered. The other wondering of the research question is about the comparison between the practical examples on the current level.

2.3. Concept and Relations

The topic of CP is commonly related to various areas which are illustrated in the Fig. 39, inspired by Mortenson et. [92] to map the topic. Specifically, decision making which is a vital part of project management and artificial intelligence are getting integrated with a subject. Moreover, although operation research is mainly associated with mathematics and statistics, it has lots of similar traits and purposes with CP. These can be seen as competing tools for the same types of optimization problems. Also, the closeness of the formulation of MP and CP make the topics related. In the literature, there is also studies that classify the CP under the operation research. These interconnected disciplines and their relationships have been explained in literature review.

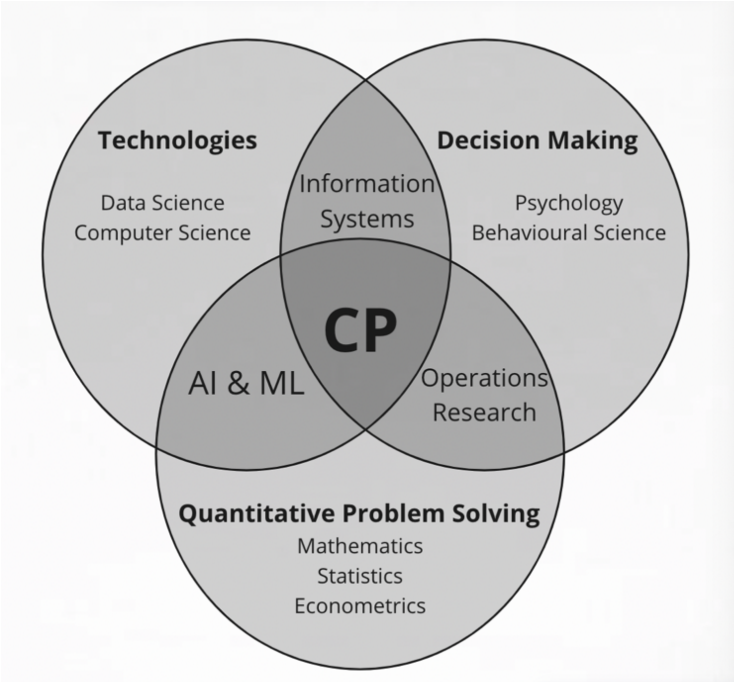


Figure 39: Relations of CP with related concepts

At this stage, the relations between the concepts of the thesis must be clarified with a frame. The thesis proposes the potential and efficient use of CP to solve project problems and support decision making especially within projects' time and resource limits. Therefore, in order to understand how CP can be used as a decision support and problem-solving mechanism, its role has been illustrated with the decision modelling flowchart on Fig. 40. Also, the chart has been developed to position of CP in the decision-making process which also responds to RQ.1 and RQ.3.

As introduced in [97], the decision-making process consists of three stages which are formulation of the problem, solution of the problem and interpretation of the results. Namely, the first step is defining the decision problem as a constraint satisfaction or optimization problem (CSP/COP). It is important to note that it is necessary to understand and express the problem with related parameters. In real life, it could be hard to draft a clear and concise statement of a problem, therefore, an intensive observation of the problem and formulation is important to remaining part of the process. The process continues with developing a model. Except for using a CP solver, the ease of formulating a model with respect to mathematical programming techniques provides an advantage to the user.

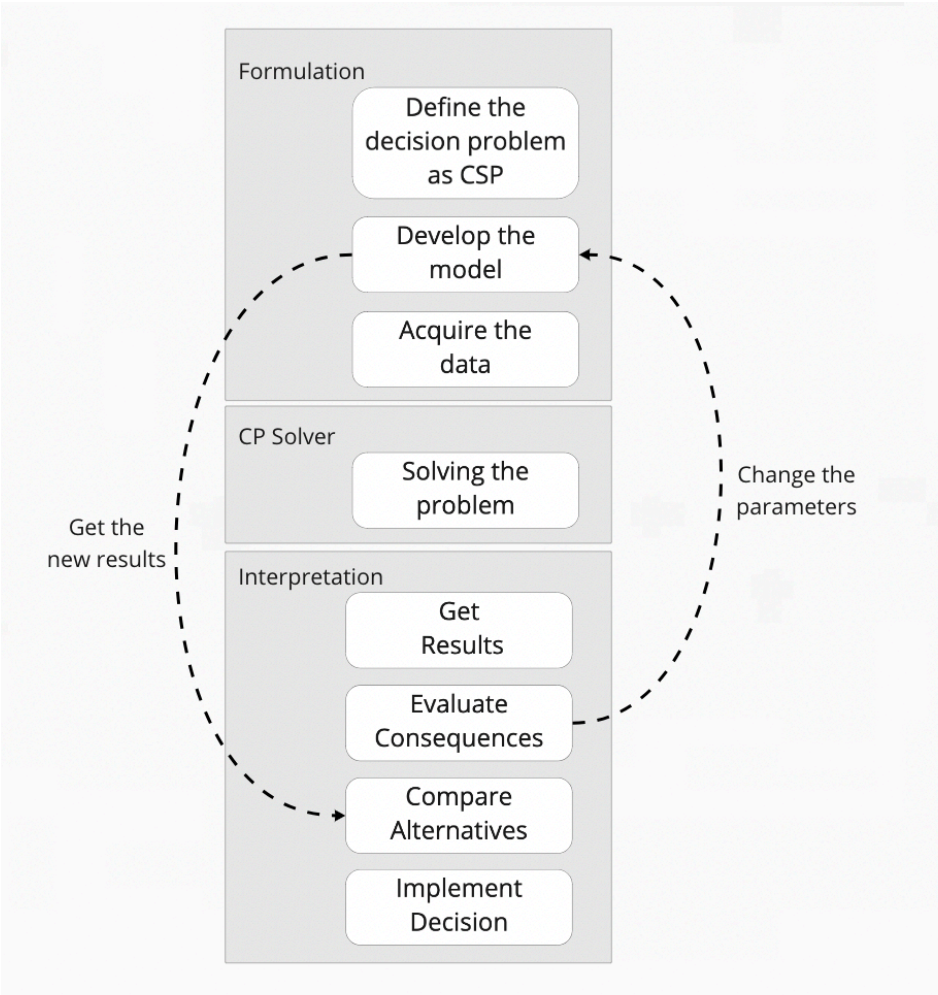


Figure 40: Proposed decision-making flowchart

For example, it is not mandatory to express the values numerically, as mentioned in the literature review, nevertheless using specific and quantifiable variables and parameters are required in an analytic problem solving. In the following step, an input data is introduced to a model which can be drafted from several sources such as reports, documents, observations, interviews etc. In the project context, the parameters can be related with due time, defined budget, resource capacity and so on.

In the solution phase, take form with CP which is used to solve combinatorial problems. The strength and motivation for using a problem that is defined with variables and constraints can be solved with a search procedure of CP to find the solution. Moreover, several successful and well-developed solvers are available to solve the models with various characteristics which are mentioned in Section 1.4.8. therefore, a project manager or organization can select depending on their needs.

The following phase is the interpretation which involves getting the results, evaluating the results in real-life context, and if necessary, changing the parameters and introducing them to the model to get new results, then comparing the alternatives and selecting the best alternative to implement the selection. CP optimization engine enables the user to run the model and get results multiple times with very limited effort and time. With the support of CP solver, project managers can observe the alternatives and take possible different solutions for different parameters. As an example, different completion time can be defined to understand how this modification affects the resource usage or project budget. Alternatively, converting a soft constraint to a hard constraint can create a difference on the number of solutions found. According to input which is given by the user to the model, the alternatives can be received. Consequently, CP supports the project manager to solve the problems and make decisions.

Differently from the models discussed in the literature, in the schematic overview of the decision-making process that is proposed by this thesis, the modelling stage is effectively carried out and the solution stage is effectively passed with constraint programming solvers. Moreover, the stages are adjusted according to the potential benefits that can be brought with CP which are developing the model easily by user and getting new results for possible different scenarios.

In order to highlight the scope of this thesis and clarify the umbrella terms it is important to note that some explanations should be defined. According to Burke and Barron, problem solving and decision making can be seen as the two sides of the same coin. In project management, these are used to come up with feasible or optimal solutions. The problems and their solutions can be related with all elements of the project such as schedules, allocations, planning etc. The thesis considers problem solving topics as an analytical and cognitive approach rather than interpersonal and social. In other words, it can be interchangeably used as conflict and communication management in many project contexts, however the problem solving has been come up as a systematic manner in this study.

As another justification, the topics such as machine learning, deep learning which are interchangeably used as artificial intelligence are not the main focus of this thesis but instead, the purpose of this work is to review and understand how to use the constraint paradigm, a sub-topic of artificial intelligence, as a support for the project management fields. Moreover, the role and responsibilities of a project manager can vary among organizations. The job descriptions do not have a single and certainly defined form. However, it is obvious that taking critical decisions and solving problems are regular tasks for the managers. In this study, the resolutions on the operational level such as optimization, scheduling, resource allocation, etc. are seen as a part of their job.

2.4. Research Methodologies

Throughout the study, peculiar techniques have been used for collecting knowledge and gathering insights to answer the research questions and shape the course of the thesis. In order to provide a relationship between applied research methodologies, generated outputs and research questions, Table 7. has been illustrated.

Table 7: Research Methodologies adopted

Research Methodologies		
Method	**Insights**	**Answers to RQ**
Literature Review	The insights and knowledge have been gathered about the CP and PM. The potential benefits and promising areas have been identified.	**RQ. 1, 2, 3, 4**
Case Study & Analysis Framework	The possible problem solving techniques have been used and in the lights of the findings from the results and literature review, CP have been analyzed.	**RQ. 1, 3, 4**

In the literature review, the insights and expertise have been acquired related to the CP and PM. The possible advantages and favorable application fields have been specified to answer all the research questions. Then, with a practical case step by step comparison between different viable problem-solving methods and procedures have been performed and the findings have been examined in relation with the literature analysis. Not represented but some other techniques have also been chosen to work systematically. For example, in terms of specifying the core topic of the thesis and selecting a case to be presented to manifest the hypothesis, basic funnel structure inverse pyramid and comparison matrix have been utilized, which have been demonstrated in the related sections.

2.4.1. Methodology of Literature Review

In order to define the ground and set the boundaries, a literature search has been carried out. Large databases such as ScienceDirect, Mendeley, Scopus have been used to collect academic literature. Most of the topics underresearched, constraint programming and project management, are well-developed and mature fields.

Therefore, apart from academic documents found on these platforms, related books and internationally used standards have been used. At the same time, considering the novelty of the AI topic and digital transformation potential of CP on PM, the studies from various trusted advisor and counselor firms such as McKinsey, Accenture, PwC have been used. In such a way, scholastic research has been augmented with their reasonable insights and real-life studies.

As the first step of this research, the subjects of Project Management and Artificial Intelligence have been examined as preliminary research to specify the research questions and the core topic of the thesis. The study has been conducted in different dimensions that are practically and theoretically. In practical perspective, real-world examples, industrial applications and promising areas to thrive with AI have been reviewed. Several potential and existing AI applications and areas in the project management context have been reviewed. Some of the potential topics can be listed as:

- Communication Management (Information distribution, performance reporting)
- Risk Management (predicting the impact from risks and issues)
- Progress Evaluation and Resource Management
- Scope/Time Management (especially for repetitive task)
- Budget/Cost Estimations
- Data/Fact Driven Decision Making for Uncertain/Risky Cases
- Forecasting Project Scenarios and Outcomes
- Lessons Learned
- Personal Motivation and Behavior Support and KPI (thanks to empathy ability and personalization and personal evaluation)

From the theoretical point of view, as a Management Engineering Master of Science student, initially, fundamental academic resources are checked about AI. For that reason, one of the cult books 'Artificial intelligence: A modern approach' [21] has been gone through. It caught the attention that Chapter II presents problem solving with constraint satisfaction problems because one of the critical roles and responsibilities of a project manager is problem solving. Then, the related topics have been canalized by the author. The frequently encountered subjects on the topic of constraint programming are constraint satisfaction problems, AI problems, constraint optimization problems, search mechanism, constraint propagation, exact and heuristic methods to find a solution. At this stage, it has been recognized that some of the problems which belong to classical AI problems such as cryptarithmetic, sudoku, eight queens, map coloring etc. Over and above that lots of real-world scenarios can be expressed as CSP such as vehicle routing, planning, timetabling, assignment, job scheduling. In advance of discovering the topic of the study, the examples faced in the literature have grasped an attention to constraint programming. Moreover, the nature of projects which include various limitations and constraints have been found up-and-coming to apply constraint programming technology. Having insights gathered from literature in mind, problems that can arise in a project environment and the benefits of constraint programming to the problems – and as a matter of course to project managers – have been decided to focus on the thesis.

Moreover, CP paradigm handles with similar obstacles which are optimization, problem solving, decision making with mathematical programming. Therefore, mixed integer programming, linear programming, goal programming and so forth are investigated. Subsequently, hybrid methods that integrate MP and CP to take advantage of the strengths of the each of them.

In the phase of preliminary research, an examination about project management had been kept on basic thanks to background information and knowledge of the author. However, when the literature review has been limited to 'business, management and accounting' and 'decision science' subject area for CP topic, many analytic and complex problems that are possible to occur in a project came out such as resource limited project scheduling, floor planning, production scheduling, supplier selection problems, inventory management, crew assignment etc. The Fig. 41 illustrates the canalization of the scope of the study throughout the literature review.

Over and above that the study has continuously and iteratively been backed up by literature. In order to answer research questions, the existing examples and applications are detailly examined and potential areas of improvements have been searched. With the applied case study, the theory and practice have been combined and the minor problems, unexpected issues, points that require technical knowledge have been solved with the reference studies from literature.

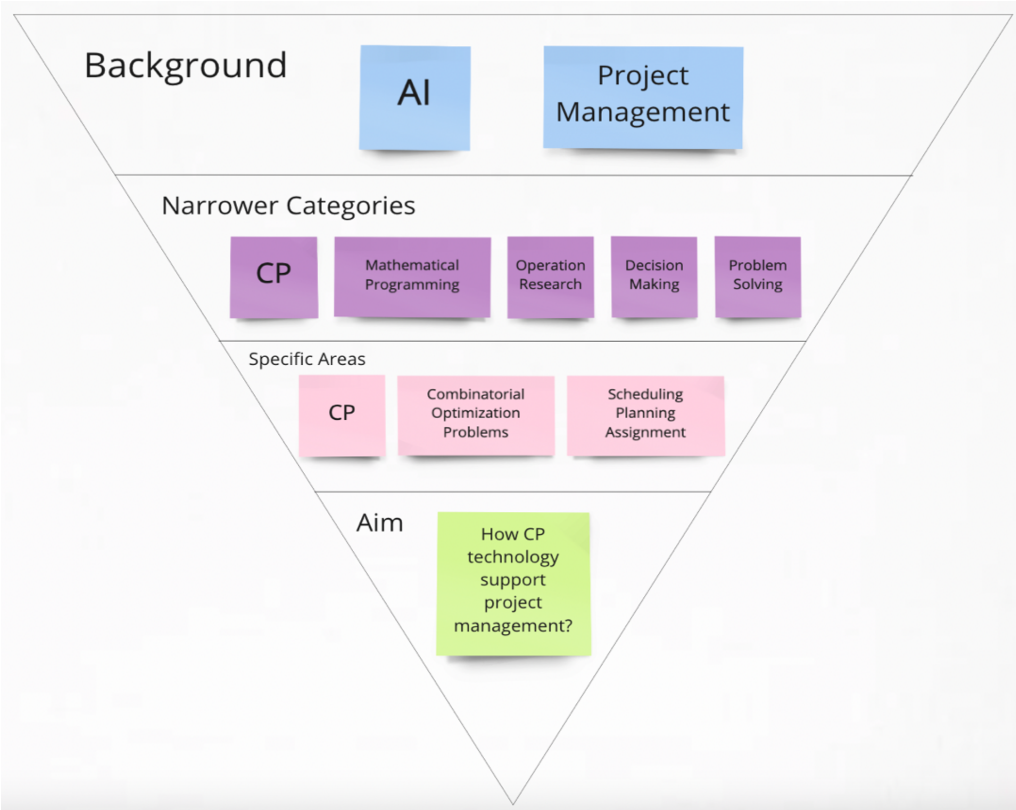


Figure 41: Inverse pyramid for thesis topic specification

2.4.2. Practical Case Study Analysis Methodology

In real life businesses regularly tackle different complex scheduling problems after the planning phase in the field of project management and there are various solution techniques. In this manner, one of the major analyses performed in this thesis is related to the comparison of these solution methods applied to diverse scheduling problems similar to real life cases. One illustration to these business settings can be the following; a project is commissioned by a customer to a company operating in the aviation industry and the company has a team of 25 people available to be assigned to this project. The customer wants the project to be delivered at most in 180 days. So, the project team starts to break down the tasks needed to be done for completion to sub-activities and assign the required time and resources to each. Since the resources of the company are limited and not infinite, on average it is not possible to deliver the project with the available resources and at this point the company has some options such as hiring more staff, acquiring extra machinery or outsourcing in order to balance the so-called demands of the customer and supply of the company possessions. In this thesis, an equivalent business case was tried to be modelled by defining a number of constraints and variables then, to be addressed with procedures such as Gantt chart, CPM by hand, LP and CP. These solution modes have particular characteristics and not all of them are applicable to every problem model.

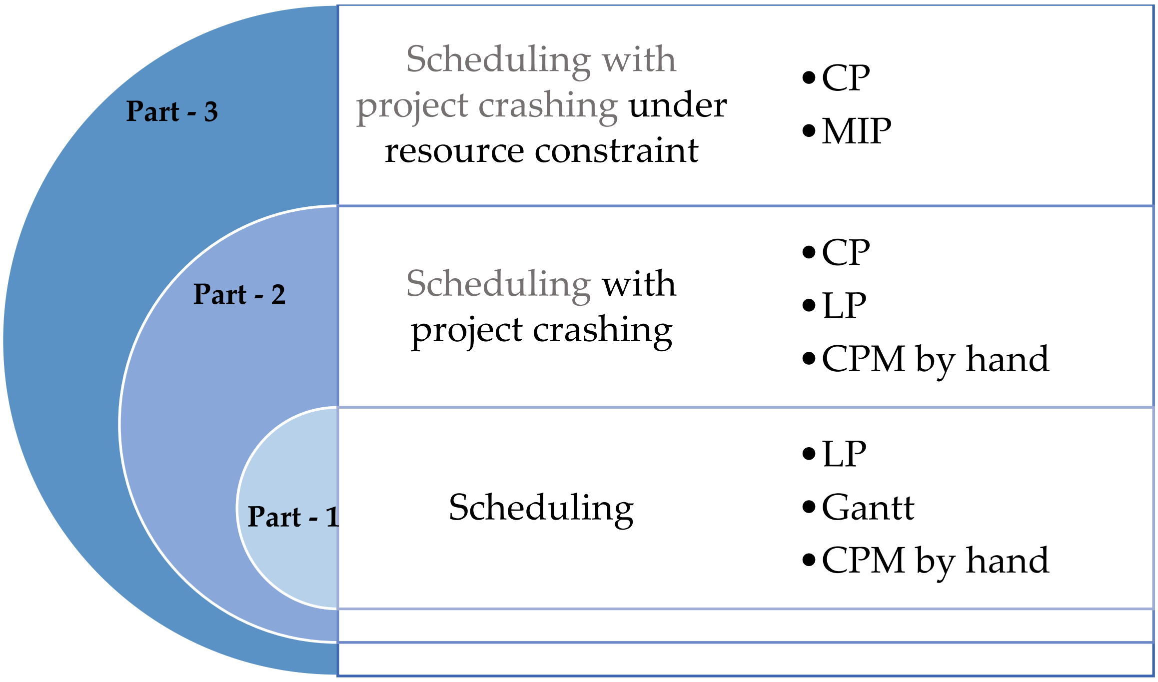


Figure 42: Incremental Progress of problem and corresponding models

As discussed in the literature and can be seen from the Fig. 42, the problem models are scheduling, scheduling with crashing and scheduling with crashing under resource constraints. Naturally, there are some assumptions made in order to simplify the problems from real life. Furthermore, in the following sections, the developed practical case models containing a problem description, selection of parameters about the problem and techniques, assumptions made and step by step explanations of the modelling are discussed in detail. The ultimate goal of the comparison analysis between the solutions methods of Gantt chart, CPM by hand, LP and CP when practiced to different scheduling models is to identify the problem-solving capabilities of these techniques in terms of both qualitative such as natural programming language affinity, ease of modelling, the existence of Gantt chart as a support to solution and adjusted resource levels, and quantitative like the time and memory usage to find the solution.

Selection of the Problem

To represent the adoption of the thesis topic and to illustrate RQ.1 with a practical case, a project problem has been solved using different approaches. As one of the main application areas of CP and project management, a project scheduling problem has been selected which has been classified as NP-hard problems. In the literature, academicians and practitioners are commonly working on these problems and several types of scheduling problems exist such as single and multiple machine scheduling, job shop scheduling, etc. [100] As one of the most comprehensive and interdisciplinary databases, Scopus has been reviewed to be able to ensure that scheduling is a problem worth investigating furtherly. As a result, it can be seen that more than half of the sources which indicate AI and PM in the subject area of business, management, accounting and decision sciences mention scheduling, which can be viewed on Fig. 43a.

Name	Query	Documents
CP AI Scheduling	TITLE-ABS-KEY ("CONSTRAINT PROGRAMMING" AND "ARTIFICIAL IN TELLIGENCE" AND scheduling) AND (LIMIT-TO (SUBJAREA , "DEC I") OR LIMIT-TO (SUBJAREA , "BUSI")) View Less Edit query	30
CP AI	TITLE-ABS-KEY ("CONSTRAINT PROGRAMMING" AND "ARTIFICIAL IN TELLIGENCE") AND (LIMIT-TO (SUBJAREA , "DECI") OR LIMIT-TO (S UBJAREA , "BUSI")) View Less Edit query	53

(a) Scopus results for CP, AI, Scheduling

Scheduling problems draw common attention from PM and CP, and naturally AI communities. In order to make a holistic search the search has been carried out within not only in abstract title, keywords and abstracts but also in all fields. As can be seen from the Figure 43b, there is little or no analysis revolving around the CP techniques applied to scheduling with crashing especially under resource constraint thus this reasoning implies that it could bring value and be innovative to study this topic in the scope of this thesis.

Name	Query	Documents
AI CP PM Sc. Crashing Resource	ALL ((ai OR "Artificial Intelligence") AND ("Project Management" AN D project) AND "Constraint Programming" AND scheduling AND res ouce AND (crash OR crashing OR "project crashing")) View Less Edit query	0
AI CP PM Scheduling Crashing	ALL ((ai OR "Artificial Intelligence") AND "Project Management" AND "Constraint Programming" AND scheduling AND project AND (crash OR crashing OR "Project Crashing")) View Less Edit query	11
AI CP PM Scheduling	ALL ((ai OR "Artificial Intelligence") AND "Constraint Programming" A ND scheduling AND project AND "Project Management")	288

(b) Scopus results for AI, CP, PM, Scheduling with crashing and resource

Figure 43: Scopus results

Moreover, there is also a broad problem type which is Resource Constrained Project Scheduling Problem (RCPSP) that comprehends many other classical scheduling problems. Similar to most of the projects ultimate goal, the objective of the scheduling problem is the meeting of the project deliverables which is completion of all the project activities with project resources and keeping the project duration and project cost minimum. Therefore, the trade-offs which could be between key constraints such as time and cost should be evaluated by the project managers. Considering these practical needs, a case is developed for project crashing.

Division of the Problem

The case has been separated into three sub-parts which are a basic scheduling problem, a project scheduling problem with project crashing and a resource constrained project scheduling problem with project crashing which can be seen on Fig. 42.

In the first part of the cases, a basic scheduling problem has been tackled which aims to become proficient in the earliest finalization time of the project. The only decision to be made is finding the starting and end dates of each activity. At this stage, the problem has been described with a set of activities, the precedence relations between activities and the durations of the activities which can be seen in Fig. 44 for the sample solution case which is for the project with 5 activities. For this base part, a simple problem with 5 activities and another more complicated version with 25 activities have been designed in order to observe the increasing exertion in relation with increasing complexity for the Gantt chart technique as well as hand calculation for CPM. Increasing the activity number further has been deemed as no necessary since the CPM by hand technique is already really hard to be implemented for 25 activities.

Activity	Precedence	Duration (hour)
A	-	56
B	A	88
C	-	72
D	C	40
E	B,D	140

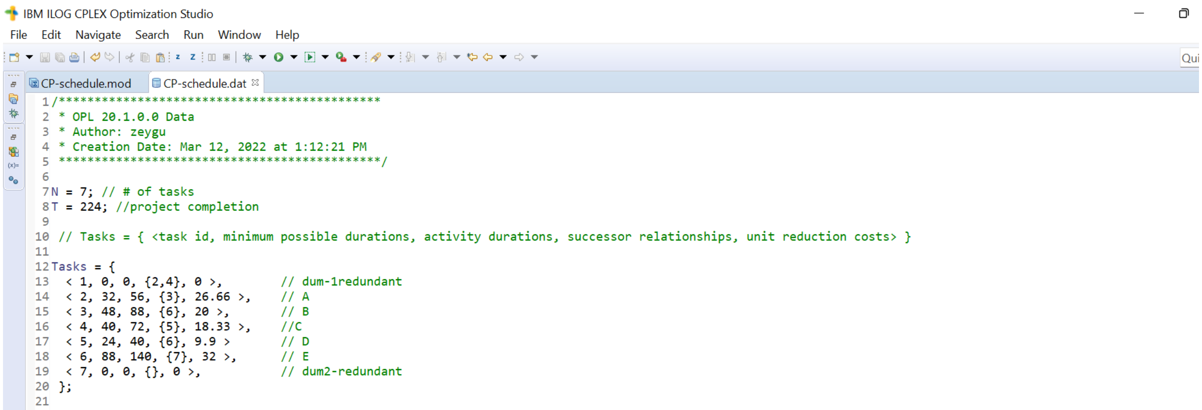
Figure 44: Given problem for scheduling 5 activities project

In the part – 2, the cases have been extended with an optimization function for minimizing the cost of project crashing. To do so, the unit reduction costs for tasks have been put on the problem data as inputs to determine how much crash can be done with keeping the cost minimum. Moreover, the minimum allowable activity durations with maximum crashing and activity durations without any crash has been provided. The problem turns into a combinatorial problem because the use of optimization function is required to find the minimum cost of project crashing and satisfy the completion of the project on the defined duration which has been taken as 224 hours for the sample solution. For this part, there are different variations of the problem that at first the problem consists of 5 activities with a required project completion time varying between 232 to 164 with 8 intervals while all the other inputs remain same. Secondly, with the same inputs of precedence relationships, durations and cost of crash, the activity number is increased to 25 and the project completion duration is varied between 450 to 390 with 10 intervals. Thirdly, with the same logic the number of activities is modified to 50 including two additional dummies and the project completion time is altered in between 70 and 45 with 5 intervals. Indeed, adjusting the problem characteristics like number of activities designates the level of complexity of the problem. In this manner, the main goal to construct these problems described is to understand the strengths and weaknesses of some solution techniques under changing circumstances. The logic behind these choices of problem building is that to monitor and investigate the crashing cost fluctuation when the customer or management demands a shorter or longer project completion time. Within this direction, as a side goal, the analysis in the big picture wants to detect the mechanisms of problem solving to help the managerial decision-making process.

Activity	Precedence	Successor	Min. Duration (hour)	Duration (hour)	Cost of Crash (€/hour)
dum1	-	A,C	0	0	0
A	-	B	32	56	26.66
B	A	E	48	88	20
C	-	D	40	72	18.33
D	C	E	24	40	9.9
E	B,D	dum2	88	140	32
dum2	E	-	0	0	0

(a) Given information about the problem

The parameters and input data for the sample case problem has shown on Fig. 45a and 45b for solving the crashing problem without any resource constraint.



(b) Input data in software

Figure 45: Given problem for crashing 5 activities project

In the last part, the actual, more realistic, case is developed with adding required resources for each activity and overall resource capacity for the project allocated by the project management as inputs. The Fig. 46 indicates all the given data about the case, for sample solution in this part, of project scheduling with crashing under resource constraint. There are temporal and resource constraints, one for the required workers and another for the required workers, between activities and an objective function about keeping the budget minimum. It focuses on finding an optimal solution that ensures the project to be completed on the defined timeline, which is taken as 224 hours again for the sample solution, with finalization of each task satisfying the available resources and other constraints. In order to compare the solution methods to the scheduling with crashing under resource constraint case, the problem has been extended that there are 5 activities initially, then it is increased to 25 activities and after that to 50 activities which can be checked on Annex - B.1 with details. For these three activity numbers set, at first the project completion duration condition is varied between 232 and 224 with 8 intervals for 5 activities, between 450 and 420 with 10 intervals for 25 activities and between 70 and 45 with 5 intervals while setting a capacity less than the minimum needed to complete the project without crossing the line such that there is no resource constraint at all. After that, secondly, again for all number of activities defined, in the beginning, the machine resource is assigned as unlimited, 99, and the worker constraint is altered in between unlimited and not enough to finalize the project obeying the case data thirdly and lastly the other way around is implemented as the worker resource is fixed as unlimited, 99, and the machine constraint is modified in between unlimited and not enough. As mentioned in the previous part, the main goal is same which is making the comparison between different solvers. The side goal also remains as it is which is trying to find a concept that is beneficial for the managerial decision making within a project scope.

Activity	Precedence	Successor	Min. Duration (hour)	Duration (hour)	Resource Req. (worker)	Resource Req. (machine)	Cost of Crash (€/hour)
dum1	-	A,C	0	0	0	0	0
A	-	B	32	56	4	3	26.66
B	A	E	48	88	8	2	20
C	-	D	40	72	3	4	18.33
D	C	E	24	40	11	5	9.9
E	B,D	dum2	88	140	9	1	32
dum2	E	-	0	0	0	0	0

(a) Given information about the problem

The Fig. 46b has been introduced to the system with the input data for the sample solution case as the following:

```
N = 7;                              // #of tasks
T = 224;                            // project completion
Capacity = 13;                      // #of worker
Capacity2 = 6;                      // #of machine

//Tasks={ <task id, minimum possible durations, activity durations, worker resource req.,machine resource req., successor relationships, unit reduction costs> }

Tasks = {
 < 1, 0, 0, 0, 0, {2,4}, 0 >,         // dum1 - redundant
 < 2, 32, 56, 4, 3, {3}, 26.66 >,     // A
 < 3, 48, 88, 8, 2, {6}, 20 >,        // B
 < 4, 40, 72, 3, 4, {5}, 18.33 >,     // C
 < 5, 24, 40, 11, 5, {6}, 9.9 >,      // D
 < 6, 88, 140, 9, 1, {7}, 32 >,       // E
 < 7, 0, 0, 0, 0, {}, 0 >             // dum2 - redundant
};
```

(b) Input data in software

Figure 46: The problem for crashing under resource constraints for 5 activities project

Selection of Parameters

The activities and the related data such as the durations, the precedence relationships, resource needs and unit cut costs are generated hypothetically for 3 data sets. To compare the performance of the methods, the projects with 5, 25 and 50 activities has been produced. However, considering the effect of input data on the performance of the techniques the parameters has been selected in a systematic manner such that the amount of resource needed for a project without any crash and any resource constraint has been calculated and kept constant for all project sizes. In other words, it is important to note that for all the parts, the parameters for 50 activities, 25 activities are designed such that being consistent with the ones for 5 activities. The Table 8. shows the resource need for workers and machines, C1 and C2, respectively and selected parameters while evaluating the cost under the crashing. One can think that a project needs to have 25 workers and 15 machines to finalize the work without any crash in an ideal case, however a case has been examined as if there is a scarcity for the resources such that a project has 17 workers and 12 machines with an obligation to finalize it in a shorter duration. The rate of scarcity has been kept constant for three of the projects.

Table 8: Constraint valuation with constant rate

	DATA1		DATA2		DATA3	
	C1	C2	C1	C2	C1	C2
No crash no resource constraint	19	7	25	15	41	54
Input data with crash	13	6	17	12	28	46

The problem instances are expressed by the indexes such that (T_{TARGET}/C1/C2) which index T_{TARGET} denotes target the project duration, C1 and C2 denote the number of workers and machines, respectively. For instance, the project completion duration requirement is 224 hours for 5 activities which is nearly %20 of the project completion duration without crashing availability and indeed the same percentage is preserved for 25 activities which corresponds to the project completion duration requirement of 440 while it is 559 for non-crashing case.

Assumptions Done

In this case, some reasonable assumptions are made to simplify the model and to formulate a real-life problem to some extent. It is certain that in real life, there are lots of uncertainties and risks on the projects. However, some of them are not limited to theoretical studies or models, also they can be applicable even in the real projects. Although, in project management these unexpected and possible risks or opportunities should be contemplated, it is not practical to define every aspect of the project in a model that solves a fundamental project problem to support decision making of project managers. Therefore, some presumes have been taken for granted and the model formulated accordingly.

Part - 1 assumes that if an activity starts, it cannot be interrupted or discontinued which means scheduling non-preemptive activities. Also, it assumes the availability of the resources. In terms of scheduling, the activities can be executed in parallel manner. Furthermore, to the second part, the assumption is added for the continuity of the linear time-cost function which is shown on Fig. 47. In other words, the more cost allocated means the shorter time for the project completion.

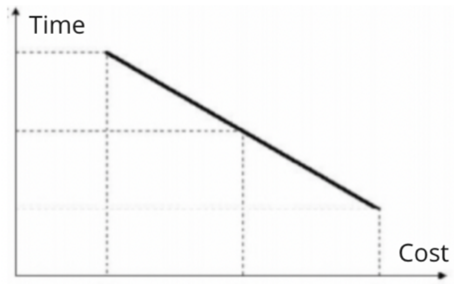


Figure 47: Linear time-cost function assumption

Besides all the mentioned ones, lastly, for the resource constraint which implies the workers, additional assumptions are also made:

- Activities share two types of resources which are workers and machines.
- All resources are identical and give full performance continuously.
- A resource can be used for each activity without any skill, proficiency or performance constraint.
- A resource can only be used for one activity at the same time (disjunctive scheduling).
- There is a bounded capacity for the resource and cannot be exceeded.
- The activities are non-preemptive, no interruption is possible.

Moreover, considering the nature of the techniques such as MIP and CP which are used for discrete optimization problems, non-negative, which is necessary for MIP and integer valued parameters are assumed, and the model adjusted. In addition, one of the data sets which has been shown in detail in the chapter is generated for the sake of simplicity and clarity. Moreover, some data sets to compare the results of the models are generated by the author with assumptions. Also, task and activity terms have been used interchangeably.

Selection of the Solving Paradigms

There is no model or technique exist to solve every scheduling problem, each project has a unique nature and faces different challenges. Therefore, depending on the characteristics of the problem such as its size and complexity, availability of the information which may require fuzzy and stochastic methods, the required accuracy and so on, the models and techniques to be utilized are selected. Apart from the purpose of reflecting academical and practical common sense for proper model selection for the problem, the sake of the logic of the analysis has also been taken into consideration in the selection of the techniques to be used. Making a fair and meaningful comparison between methods and analyzing their outputs is aimed to verify that CP can have superiority in project management problems and support the decision making especially for scheduling. In the light of these realities, different solution techniques are used depending on the part of the problem.

Firstly, the first part of the scheduling problem has been solved with Gantt chart, CPM and linear programming. As discussed in the literature review part, Gantt charts and CPM techniques are commonly used in project management areas for scheduling. To also represent various and commonly used alternatives, the linear programming (LP) technique which is applicable to CPM has been preferred to tackle the problem part - 1 and part - 2. Also, in the second part, considering a long history, constraint programming has been a success for optimization problems especially for scheduling, CP has been selected. The same model except for the syntax adjustments is used for solving the problem with CP to be able to make a proper analysis among the results. In the next step, which is the whole version of the case (part - 3), CP and its counterpart in mathematical programming MIP models are developed in view of the fact that CP is not able to solve problems with continuous decision variables efficiently. Therefore, it is more meaningful to compare it with MIP, as like the common applications in the literature [61], [62], [7, Ch. 15], the comparisons and analyses have been done between MIP and CP. In the third part of the problem, binary variables, which deals with binary integers "1" or "0" to express the resource usage, or absence, at an instance that cannot be defined with linear programming [51].

Developing the Models

The problem should be modelled realistically to reflect the problem with appropriate amounts of detail. In other words, although the more detail defined provides a better reflection about the decision problem, it can make the problem hardly solvable. Therefore, these trade-offs and the variables whose value is searched should be determined wisely. Moreover, understanding and modifying should be uncomplicated and readable because CP enables introducing new parameters and variables which can be subject to change in a project environment.

In this stage, the problem has been expressed as a constraint problem to solve it with constraint programming. The model of the problem consists of relationships as equations and inequalities. In this case, the problem has been expressed as Constraint Optimization Problem (COP) which is composed of variables, constraints, and objective function. The model tries to find an answer for the following questions. Which activity or activities should be selected to shorten while keeping the costs minimum and how much reduction should be done to satisfy the completion target.

The variables are measurable quantities that were searched as decisions to be made in the problem which was start and end times of activities and hourly crash amounts in the case. The objective function has been defined as the minimization function of the overall cost of project crashing. Some of the constraints have been expressed with special global constraints which are specific to constraint programming and some of them have been declared with mathematical balance. The fundamental components of the model have been tabulated on Table 9.

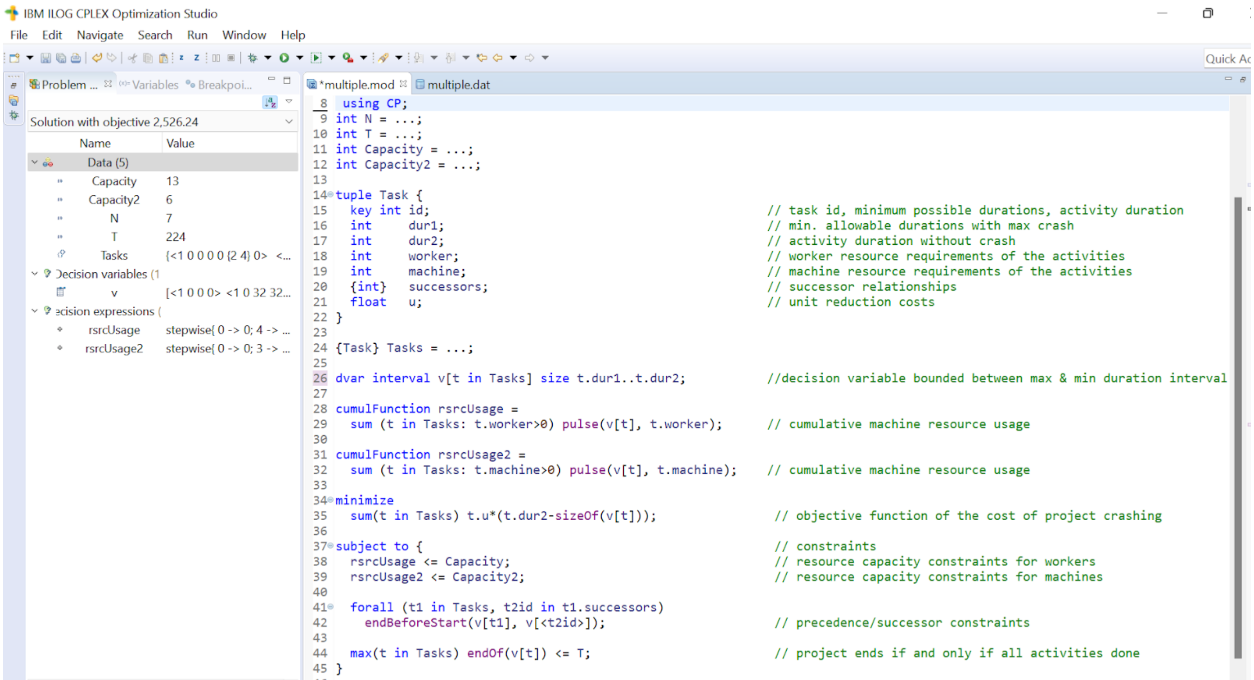
Table 9: Objectives, variables and constraints for the problem parts

	Objective	**Decisions to be made**	**Constraints**
Part -1	- Minimize the completion date	- Start time - End time	- Precedence relations
Part - 2	- Satisfy desired duration - Keep the cost minimum	- Start time - End time - Crash amount	- Precedence relations - Max. crash amount from an activity
Part - 3	- Satisfy desired duration - Keep the cost minimum	- Start time - End time - Crash Amount	- Precedence relations - Max. crash amount from an activity - Resource capacity constraint

Except for the above-mentioned constraints, some default constraints are added to express the problem such that:

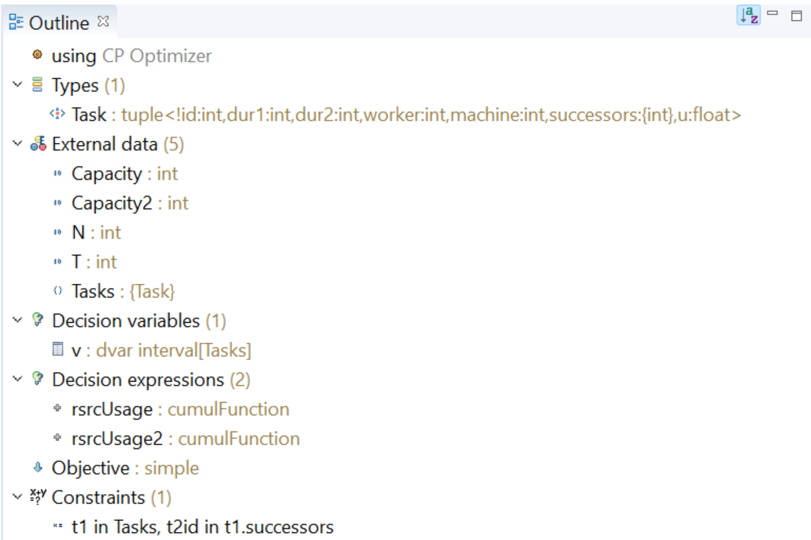
- The largest end time among the activities has to be smaller than the completion time of the project.
- The end time of an activity has to be larger than the summation of starting time and duration of the activity.

Considering the problem's incremental nature, which implies part -1 and part - 2 include all the information in on part - 3, only the model and outline of part - 3 has been represented for the sake of readability of the thesis. The models for the two parts can be seen on Annex - B as text. The designated CP model can be found on Fig. 48a for project crashing problem for scheduling under resource constraints.



(a) CP model on the software

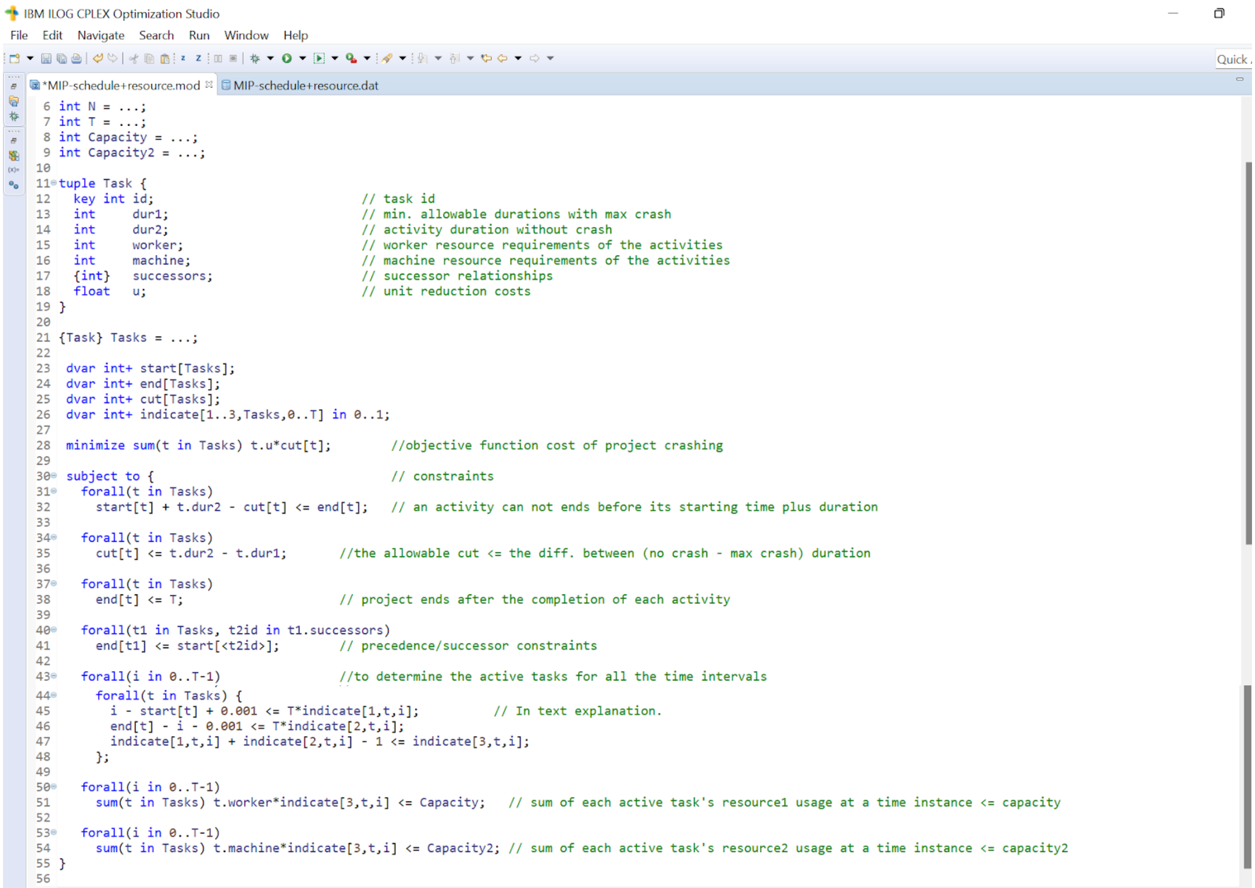
Its outline can be seen on Fig. 48b, which reflects the structure of the model.



(b) Outline of the CP model

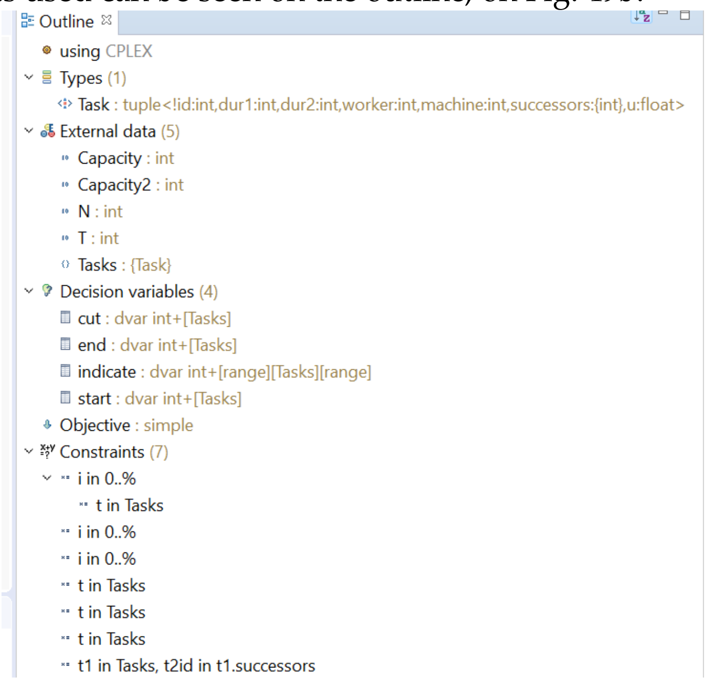
Figure 48: CP model for crashing problem with resources

The MIP model to formulate the same problem can be viewed on Fig. 49. Referring to Line 45 on the MIP model, an exhausted explanation should be done to the reader. To satisfy the capacity constraints which are written on line 51 and 54, the overall resource usages of the active tasks on an instance should be tested.



(a) MIP model on the software

Also, all elements used can be seen on the outline, on Fig. 49b.



(b) The outline of MIP model

Figure 49: MIP model for crashing problem with resources

indicate variable is used to decide the activeness of a task which acts as a binary variable. There could be 3 alternatives:

- The task t is over on time i. - done task, no resource usage
- The task t is on the time interval i. - ongoing task, resource usage
- The task t is pending on the time i. - pending task, no resource usage

To decide the activity is over or not, *indicate[1,i,t]* compares the start time of activity and *i*. *indicate[2,i,t]* compares the end time of activity and *i* to determine the task is just pending and not using any resource on the instance *i*. Lastly, *indicate[3,i,t]* shows that the individual task t is ongoing and it is using the resource on time *i*. As an illustration Fig. 50 has been presented, say task 3 takes 10 days and starts from *i* = 130 ends to *i* = 140. The function *indicate[1, (0…130),t]* can take either 1 and 0 up to *i* = 130, then the result of the mathematical inequality on the Line 45 gives 1 for the value of the function. For the remaining time instances from 140 to 224, which implies the task is already started and *indicate[1,(140 ...224), t]* gives the value 1. Then the Line 46 runs and similar procedure is carried out. Using the results of the *indicate[1,i,t]* and *indicate[2,i,t]*, *indicate[3,i,t]* function on the Line 47 gives 1 if the activity is active and using resources, as like in the demonstration presented below.

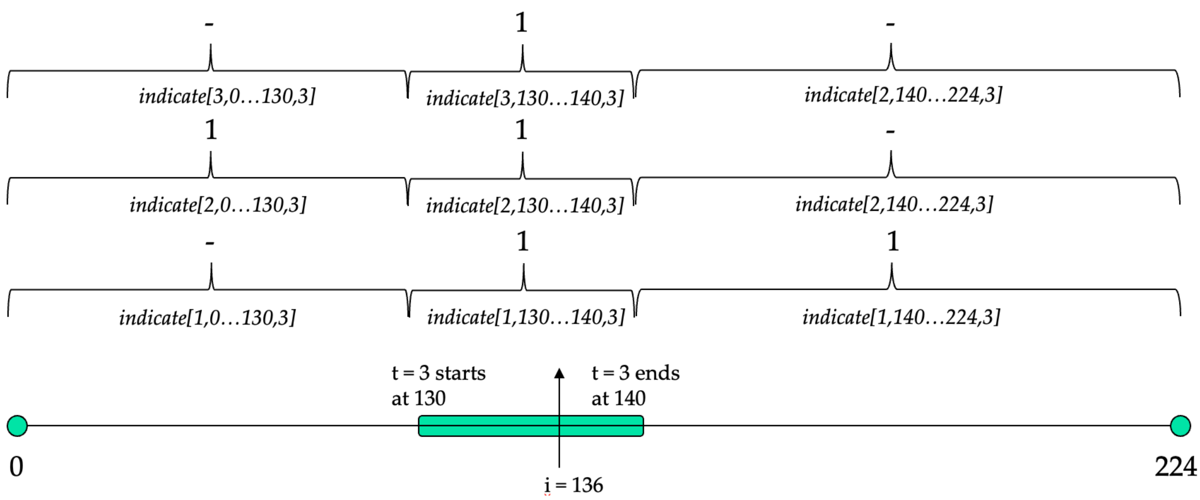


Figure 50: MIP Line 43 working principle to determine a presence of a task

In order to increase the understandability of the CP model. Some explanations about the keywords, functions, global constraints and so on have been listed.

- All of the variables to *int+* and *float+* has been set to declare a non-negative integer for each decision variable.
- Tuple sets to express the subsets of valid combinations. Generally, it consists of a combination of elements such as integer and string data.

- The *sizeof* keyword is used in the express objective function over interval variable which minimizes the overall crashing cost to calculate the duration of the activities when crashing is applied.
- The interval variable *size* is used for limiting the activity from the minimum duration to maximum duration of each task.
- The integer expression *endOf* is used for accessing the end time of an interval.
- *pulse* is a cumulative function expression which relates the interval variable that defines the task and characteristics of the tuple without a need to make a summation of each contribution on an interval mathematically.
- The precedence constraints *endBeforestart is used* on interval variables to specify precedence relationships between precedence and successor activities.
- A cumulative function for resource capacity, which is expressed as *cumulFunction* in the model for discrete capacity resource usage. In general, to formulate an element which changes over time and the value of the element is also related with other decision variables on the problem, the function is used.

The further details can be found on the comprehensive documentation site of the IBM ILOG CPLEX Studio for version 20.1.0.

Selection of Tools

Several products are available that use mathematical and constraint programming. Among them, bearing in mind the strengths of IBM ILOG CP Optimizer discussed in the Section 1.4.8, it has been decided to use it to model and solve the project scheduling problem with CP. The optimizer engine is included within IBM ILOG CPLEX Studio that has an integrated development environment (IDE) which allows users to develop, run, analyze and try the models. Optimization Programming Language (OPL) is used as a flexible modelling language to build models which is easier to do with respect to general-purpose programming languages like Java or C++. [104]

As another motivation, IBM provides free usage for researching and teaching of students and academics besides the software that allows to model and solve the problem efficiently. It is known as a next-generation CP system for complex real-world applications. Moreover, there are lots of features available for IBM ILOG CP Optimizer scheduling models. These are illustrated in Fig. 51.

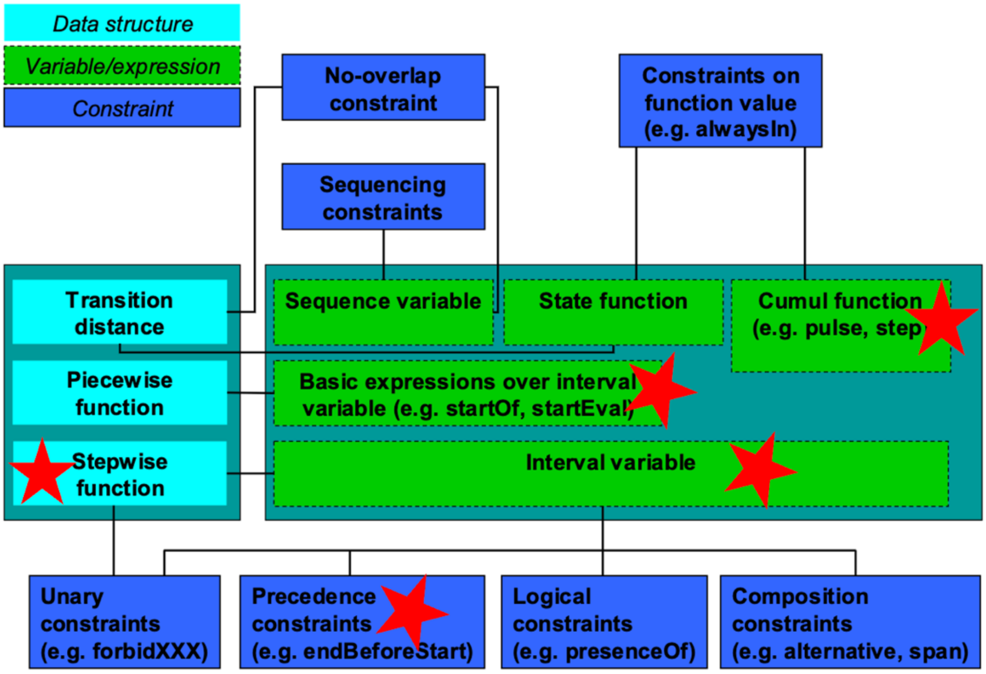


Figure 51: Used features on the CP Model

The ones that have been used for the case of the thesis can be seen with star sign. It must be underlined that IBM ILOG CPLEX is not only software package to solve problems. The feature for the separation of the model and the data enables comparing different alternatives and support to make decisions, unlike some packages such as LINGO discussed in the literature which does not allow to do so.

During the study, a basic personal computer is used to model and run the system which has the following properties:

- Processor: Intel(R) Core (TM) i7-8565U CPU @ 1.80GHz 1.99 GHz
- Installed RAM 16'0 GB (15'8 GB usable)
- System type 64-bit operating system, x64-based processor

Selection of Performance Metrics and Design of Comparison Framework

The analysis, performed on the case explained above, has been focused on the hypothesis that CP can benefit project management practices. In line with this target, comparison between in terms of both qualitative and quantitative aspects for different techniques, which are used with different frequencies by project management teams, has been made including CP. One of the quantitative performance measures has been decided on the computational runtime or time to find an optimal solution that is obviously critical by nature since a solver should not try to come up with results for hours or days. The other quantitative performance metric is the memory usage of the solver that means the load a computer has to bear to deal with a task and this determines the serviceability and functionality of a system with specific memory and computational power for a specific case hence important. In the literature, there are lots of study that solve a pure scheduling crashing problem which takes the availability of resources for granted with LP [97, pp. 253-257] and CP. Moreover, some works compare the performance of MIP and CP for RCPSP. Wari and Zhu (2019) analyze the run time and optimality with respect to batch size in terms of memory, time and number of variables [46]. Ham and Cakici (2016) evaluate the performance of models MIP and CP with a problem for flexible job shop scheduling considering computational time and the number of binary variables which is correlated with the memory usage [69], [106]. Both computational runtime and memory usage are correlated with the total number of the constraints and variables and the complexity of the case problems are by increasing/decreasing the number of activities in order to investigate this correlation as well as how various solution techniques are affected by it.

On the other hand, from a qualitative point of view, the amount of effort for modelling, which is crucial since in real life the problems are always complex and even with some reasonable assumptions to model the situation is the first step and key to end up with a solution, and level of closeness to natural programming language, which brings readability and self-explanatory characteristic to the code that eventually ends up with better user experience are two influential performance measures. Similar soft criteria are highlighted by Brailsford et al. (1999) for the proper selection of a method for problem solving [78]. Moreover, the third qualitative performance metric that the approaches have been differentiated also according to the existence of supplementary material provided such as Gantt chart, resource profile graph representation which is quite practical for decision makers since these materials visualize the situation and even sometimes when the solver could not find an exact solution, this visualization assist the project team for discovering a right direction as an analytical decision support mechanism.

So, that adds up to a total of 5 performance metrics, 2 quantitative and 3 qualitative, which are computational runtime or time to find an optimal solution, memory usage, the amount of effort for modelling, level of closeness to natural programming language and existence of supplementary materials. At this point, it is essential to state that under normal circumstances in the analysis the time to find an optimal solution metric is measured with computational runtime and memory usage metric is measured with peak memory but for some instances from the case problem, the solver could not mark an optimal result after 15 minutes, that leads to the absence of computational runtime and peak memory figures due to this issue instead of these figures the complementary ones are deemed which are time to find an optimal solution and the memory usage up to that. Furthermore, the ease of modelling metric is identified by a number of sub- criteria such as number of lines of codes, built-in functions and high-level syntax for mathematical models, etc.

A comparison framework is built with the aim of properly making use of these performance measures thus, marking the advantages and disadvantages of the project management problem solving techniques and selecting the proper one. In this manner, an assessment matrix is created with the 5-performance metrics mentioned above in its rows and the 3 solution techniques within the primary scope of this thesis which are CP and LP for part - 2 scheduling with crashing and CP and MIP for part – 3 RCPSP, in its columns. In this manner, the three solution techniques have been evaluated according to the 5-performance metrics and the related scores are recorded on the matrix. There are three possible scores that a solution technique can get as a result of the assessment procedure; √ which means the solution technique is a winner and the dominant one among the peers as for quantitative metrics it averages the best of all and for qualitative metrics either it provides the necessary materials or performs considerably well in terms of the criteria in question or ? which means even if the solution technique is not doing outstanding or not superior to every other one but still functions good enough to finalize a given task or X which means it is not feasible in any sense to use that method for that particular case. In addition, in order to generate scores numerically, all these three have numerical representations as; √ is 1 point, ? is 0.5 points and X is 0 points since this scoring is based on the reasoning that a solution technique either can solve a problem in a viable manner or not and also sometimes but not often it can be in between that the solution method operates just enough.

The empty version of the framework designed can be seen from the Table 10. In the findings and discussion part, this framework will be filled out and investigated detailly.

Table 10: Comparison framework for used approaches

	Part - 2		Part - 3	
Performance Metrics / Solution Techniques	**LP**	**CP**	**CP**	**MIP**
Time				
Memory Usage				
Ease of Modelling				
Closeness to Natural Language				
Complementary Materials				
Total Score				

In short, √ is 1 point, ? is 0.5 points & X is 0 points.

Data Collection Criteria

The quantitative performance metrics is decided according to the tabulated results of the comparison analysis between the solution techniques. At part - 2 and part - 3 of the case problems, for every parameter and constraint defined in the scope of different activity sets of 5, 25 and 50, the performance results of solution techniques capable to solve that particular problem are noted in the results table in findings part. These performance results are mainly time (T) and memory (M) requirement as indicated for performance metrics. At this juncture, it is critical to state some important methodology choices to make this result collection process consistently. So, there are 2 solution methods in the domain of interest for both parts (CP vs LP or CP vs MIP) and if both solvers are able to show a solution before the cut off time which is 15 minutes the CPU time and peak memory usage are recorded. However, if one of the methods cannot and exposed to the time off, the time to optimal solution is written down which is observable on the engine log section of the program. However, if both solvers could not finalize the search and show an exact solution, then this specific instance is not taken into account, but they will be evaluated with words in terms of the closeness of the last solution found on the engine log and the optimal solution. 15 minutes which is the cut off time is marked when there is no solution found by the solver. The CPU given by the software is in unit minute, seconds and centiseconds. It is converted to second for simplicity.

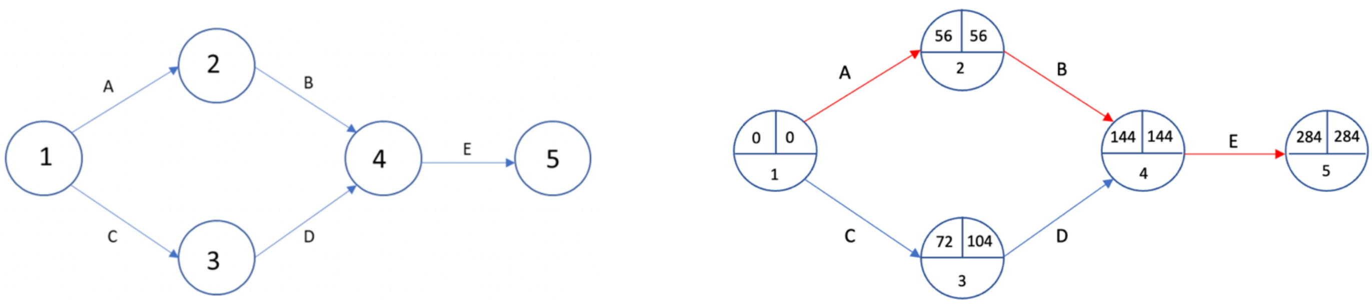
A similar logic is applied for memory requirement data collection as well if there is no cut off, the peak memory used by the solver up to finding the solution is used from the profiler section. But there is cut off, search tree space and memory usage values are considered which are available on the engine log. Furthermore, to compare and contrast the quantitative performance metrics of the techniques, the different activity sized 3 projects are considered as basis and the average T and M values found while calculating the solutions of all indicated instances for each 3 projects are compared.

Solution of the Problem with Sample Data

The solutions which have been reached with different techniques for the three parts of the problem have been represented for the sample data.

1. Part - 1: Scheduling

Critical Path Method: CPM technique, as its nature, is implemented manually by hand. First, the network diagram of the problem has been drawn according to the precedence relations between activities on Fig. 52a. The project activity network has 5 nodes, 5 arcs (i.e., activities) and 5 estimated duration time.



(a) Network diagram (b) with critical path

Figure 52: Critical path method solution

Then, the critical path is shown on Fig. 52b with red arrows. It shows the longest path to completion of the project. It has been identified by using forward and backward pass with respect to the activity durations by formulating the diagram mathematically.

Gantt Chart: The Gantt Chart has been designated on the Excel, which can be seen on Fig. 53. It can be seen that project is completed on 284 hours.

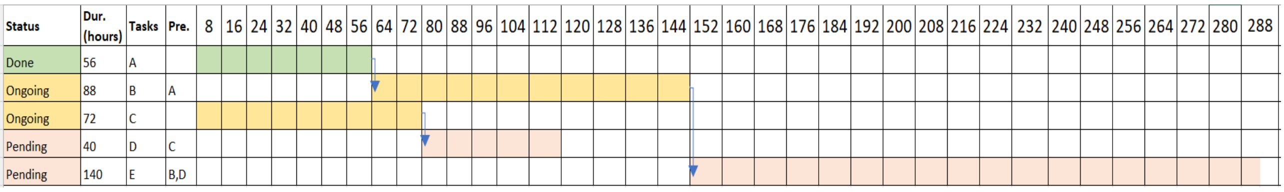


Figure 53: Gantt Chart Schedule

Linear Programming approach to CPM: The LP expresses the problem with inequalities and the variable X shows the earliest start time of the subscripted node which have been used for formulating precedence constraints with equations (1.2), (1.3), (1.4), (1.5), (1.6). Also, the objective function is to minimize the project completion time. The LP solution can be expressed as following:

min X_5
such that

$$X_1 + 56 \leq X_2, \tag{1.2}$$

$$X_1 + 72 \leq X_3, \tag{1.3}$$

$$X_2 + 88 \leq X_4, \tag{1.4}$$

$$X_3 + 40 \leq X_4, \tag{1.5}$$

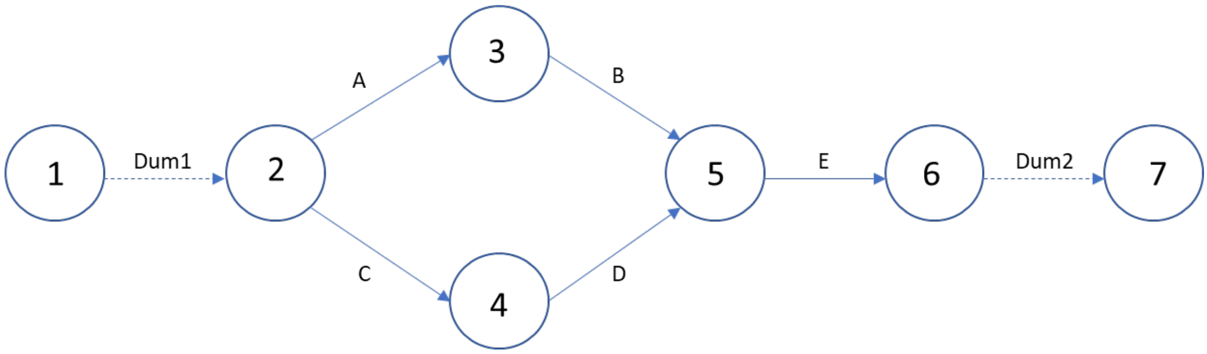
$$X_4 + 140 \leq X_5. \tag{1.6}$$

The equations are quite straightforward, and the solution set of these equations is $X_5 \geq 284$ thus the project duration is taken as 284.

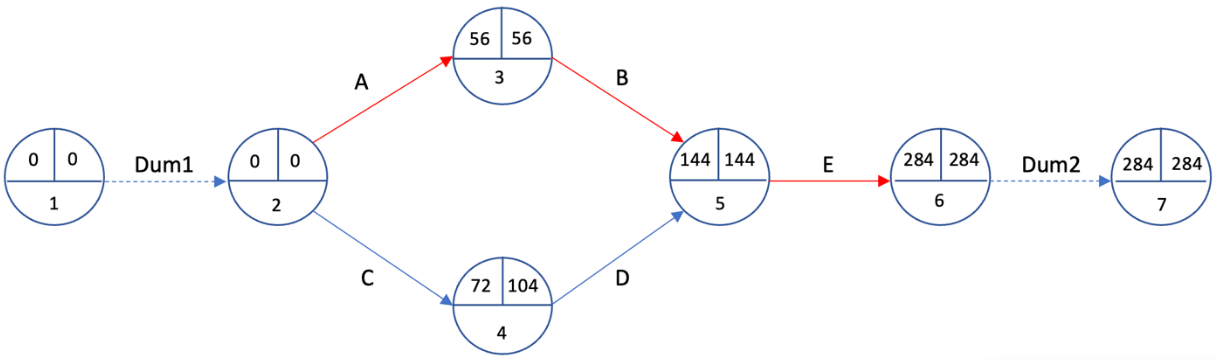
2. Part – 2: Scheduling with Crashing

CPM: The steps to solve scheduling with crashing by CPM is almost the same with the base scheduling problem as drawing the network diagram, identifying the critical path and formulating the diagram mathematically with the addition of the total project completion duration constraint and crashing cost equations.

With the introduction of dummy tasks, the network diagram becomes as the Figure. 54. and the critical path remains same for this part.



(a) Network diagram



(b) Network diagram

Figure 54: Critical path method solution for crashing

Now, the objective function (1.1) is to minimize the amount spend for crashing the tasks up to the extent that the project completion duration, 224 hours, is met. While A, B, C, D and E variables represents the crash amounts for each task. The equations (1.2), (1.3), (1.4), (1.5), (1.6) gives the precedence relations.

$$\min 26.66A + 20B + 18.33C + 9.9D + 32E \tag{1.1}$$

such that

$$X_1 + 56 \leq X_2, \tag{1.2}$$

$$X_1 + 72 \leq X_3, \tag{1.3}$$

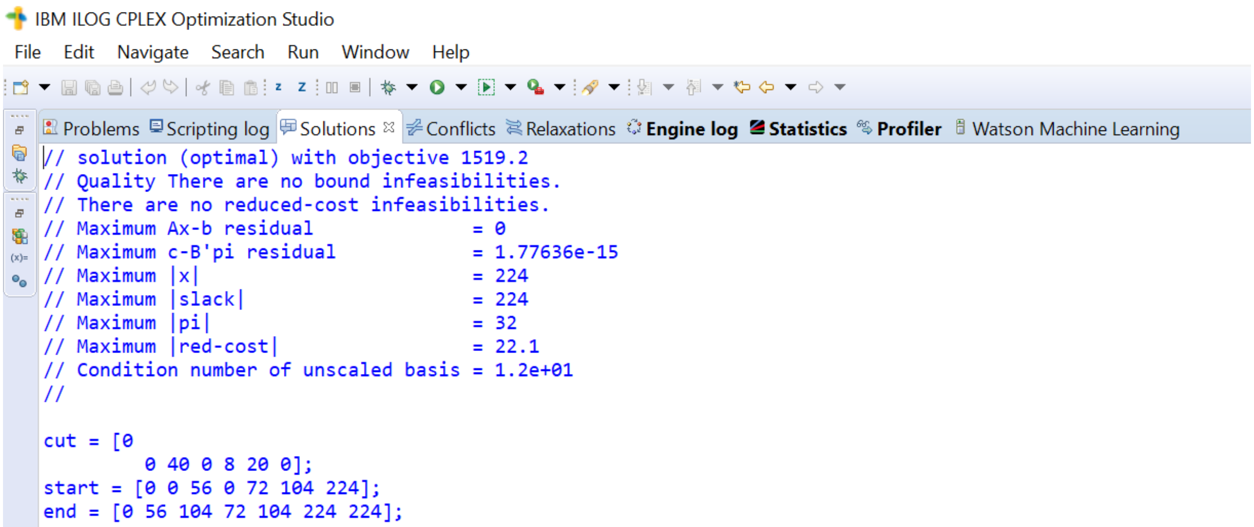
$$X_2 + 88 \leq X_4, \tag{1.4}$$

$$X_3 + 40 \leq X_4, \tag{1.5}$$

$$X_4 + 140 \leq X_5. \tag{1.6}$$

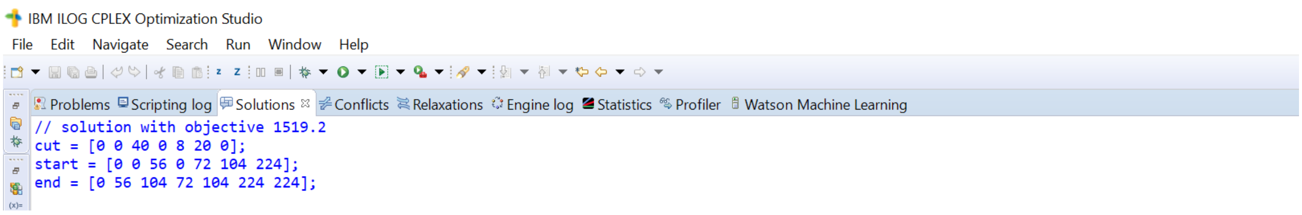
The solution procedure is more complex than the base one but still to the point that the process starts with the reduction of the rates of the task with the cheapest cut off price in the critical path and once this has been done, a status check has been performed to understand whether the critical path is changed or not. Then, the same steps are repeated until the project completion duration, 224 hours, target have been met. In short, the solution set of these equations is A = 0, B = 40, C= 0, D = 8 and E = 20. In addition, the corresponding total crashing cost to meet the target completion time can be calculated as 26.66*A + 20*B + 18.33*C + 9.9*D + 32*E = 1519.2 €.

LP: The linear programming solution can be seen on Fig. 55a for crashing amounts from each activity. Also, start and end times of the activities have been represented.



(a) with LP

CP: The solutions have been demonstrated on Fig. 55b for interval decision variables which were cut amounts, start and times.

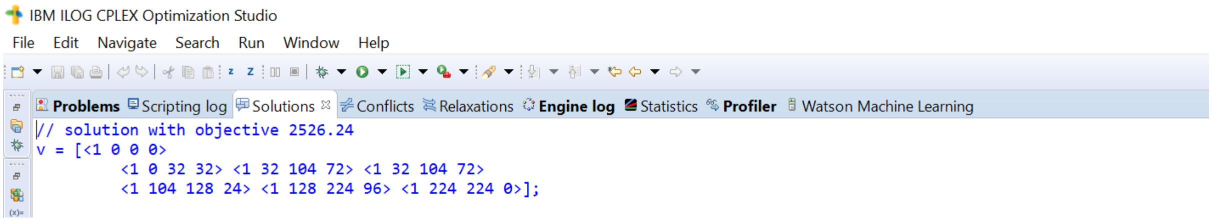


(b) with CP

Figure 55: The solution for part – 2

3. Part – 3: Scheduling with Crashing under Resource Constraint

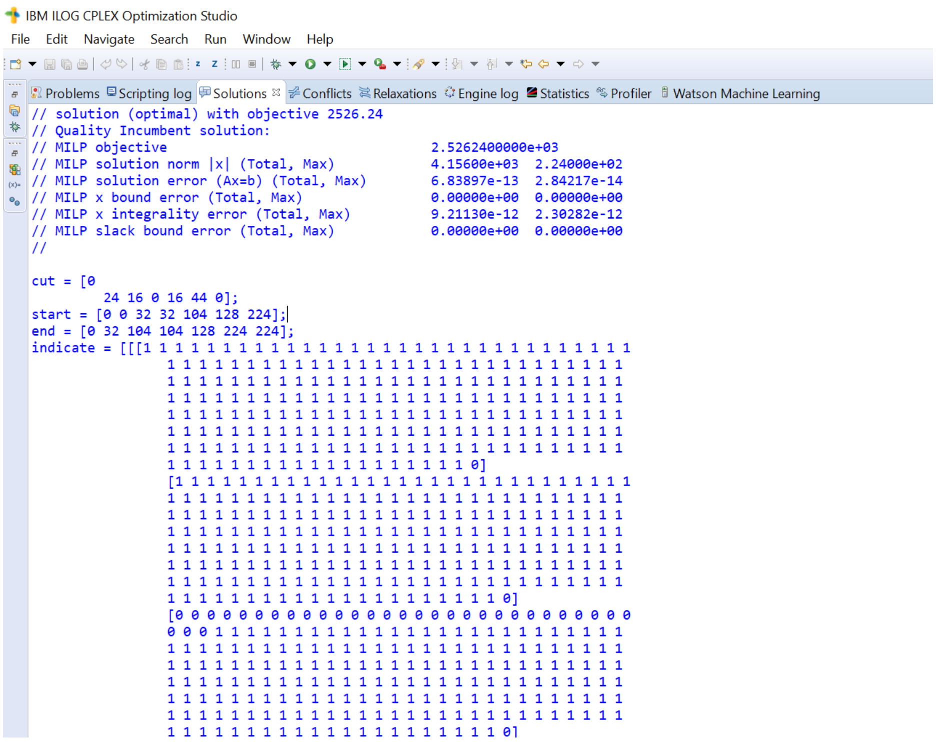
CP: The solution of the CP model has been illustrated on Fig. 56a. The result of the objective function has been found as 2526.24€ which is the monetary amount to be spent to finalize the project on the given time.



```
// solution with objective 2526.24
v = [<1 0 0 0>
        <1 0 32 32> <1 32 104 72> <1 32 104 72>
        <1 104 128 24> <1 128 224 96> <1 224 224 0>];
```

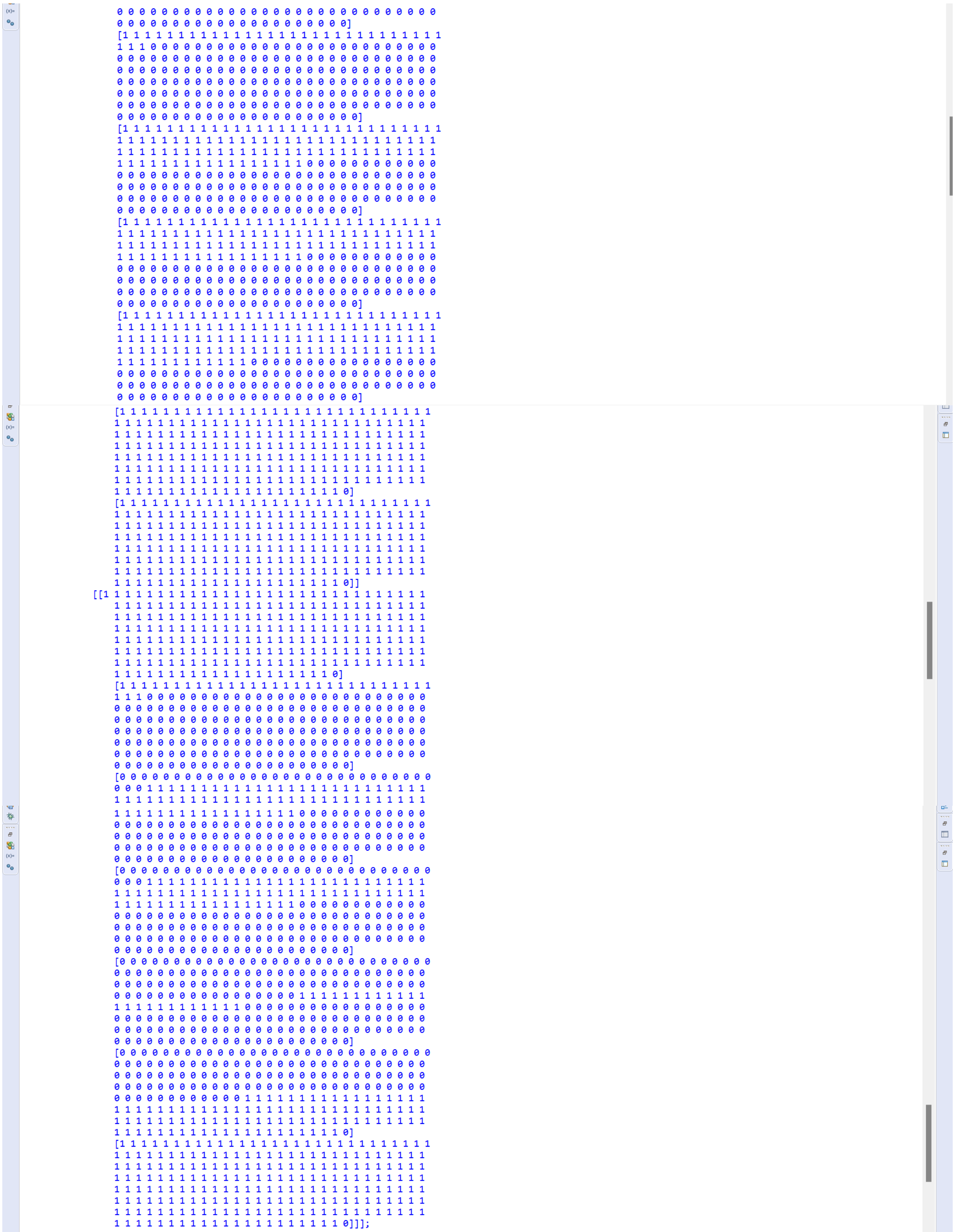
(a) with CP

MIP: The problem on the part – 3 is modelled for MIP and solved with CPLEX solver. Differently, the tasks' status about their occurrences on each instance (i.e. hour) have been found with the solution of the indicate variable that is just used as an enabler to problem solving; however it has been kept to reflect the whole solution as exist, on the Fig. 56b.



```
// solution (optimal) with objective 2526.24
// Quality Incumbent solution:
// MILP objective                              2.5262400000e+03
// MILP solution norm |x| (Total, Max)         4.15600e+03  2.24000e+02
// MILP solution error (Ax=b) (Total, Max)     6.83897e-13  2.84217e-14
// MILP x bound error (Total, Max)             0.00000e+00  0.00000e+00
// MILP x integrality error (Total, Max)       9.21130e-12  2.30282e-12
// MILP slack bound error (Total, Max)         0.00000e+00  0.00000e+00
//

cut = [0
         24 16 0 16 44 0];
start = [0 0 32 32 104 128 224];
end = [0 32 104 104 128 224 224];
indicate = [[[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0]
                 [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0]
                 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0]
```



(b) with MIP

Figure 56: The solution for part - 3

3 Findings and Discussion

Project management can be summarized as problem solving and decision making under the project constraints such as cost, time and quality. It requires to evaluate trade-offs and find a balance to satisfy stakeholders and meet requirements within the boundaries of the project. In pursuit of project success, proper decisions must be applied considering the project limits and possible shortages. However, high number of constraints and variables, which are also interconnected, and the dynamic nature of the projects can becloud to select correct strategies. In such an environment, it is very hard to compute the best or feasible scenario among several cases. The complexity of the problem exceeds the ability of problem solving of a human being. Therefore, an intelligent support mechanism is vital to make reliable decisions considering the rising importance of decision sciences in project management [47] and the expected abilities from project managers [44], [88].

At this stage, AI come to the stage to solve these hard problems with grounded approach because it is highly adopted for the applications of optimization and decision making [30]. There are various intelligent techniques that able to find the possible alternatives such as search algorithms, ML techniques, fuzzy logic etc. Among these alternatives, proper tools and applications should be developed to meet the need of analytical decision-making support system. The literature shows that time and financial requirements, technical readiness levels, the maturity of technologies can create problems in the implementation of AI systems [65]. Moreover, considering the readability and efficiency, there could be more reasonable alternatives than DL and ML as constraint programming. [57] Also, depending on the application, the performance can even be superior [76], [65]. Therefore, the instead of going with the flow, the expectations and the needs to solve the problems should be evaluated and a proper selection should be done because there can be better and simpler alternatives like CP that answer all these issues.

Constraint Programming is a technique born from the artificial intelligence domain, initially conceived to solve decision problems where a feasible solution is needed.

In similar sense, many applications of AI, OR and PM have a common point to find solutions for a set of variables which satisfy a defined set of constraints. The multi-disciplined technology CP gathers, unifies and improves ideas shared by all these domains to solve decision support problems such as scheduling, allocation, rostering, optimization, routing etc. Therefore, there is a symbiosis between AI, OR, PM and CP. These kinds of problems frequently prevail in project management considering the limited resourced nature of the projects. However, considering these problems can be difficult to compute or need technical and modelling expertise. In such a case, the critical role of a project manager is making decisions to achieve the best possible alternatives that can be supported with constraint programming.

CP consists of the modelling of a problem as a CSP and solving the CSP with constraints propagation, search algorithms and heuristics. It permits to model and solve combinatorial optimization problems. Its main modeling concepts are not too different from the ones in mathematical programming. A set of variables represents decisions, and a set of constraints relate those variables by limiting certain combinations of values. Nevertheless, the similarities stop here. Each variable is defined by a set of discrete finite possible values, called a domain. Constraints are not limited to linear equations: they apply any function to a subset of variables that limits the product of the domains of those variables. In other word, the ease of modelling is allowed with constraints that can be logical, arithmetic, or functional with expressing the problem naturally. Also, specialized global constraints can be defined for certain requirements. In such manner, these allow to reduce the variable domain with some powerful propagation algorithms.

There are also several commonly used techniques for combinatorial project problems especially for, scheduling – such as CP, mathematical programming, metaheuristics, evolutionary algorithms such as simulated annealing and genetic algorithms, machine learning algorithms. However, there is not one generally accepted method that dominate others for all problems including the most advanced ones and the simplest one. For instance, the network techniques such as PERT/CPM mentioned assume the availability of the resources. However, a case of scarcity or limit for a resource can occur in a project and that requires other remedies such as resource levelling to find an optimal schedule of a resource constrained project. Alternatively, although LP is very powerful, the approach can be used for limited type of optimization problems and are not possibly linearized so they are not suitable for discrete combinatorial problems. In mathematical programming, these problems are commonly dealt with MIP. However, an artificial intelligence paradigm CP can compete with them, too.

The insights and findings related with CP utilization in project management domain compared to other problem-solving techniques from the literature have been examined in depth through a case study in order to discover the parallelisms and divergences. All the results for the different datasets have been shared at https://drive.google.com/drive/folders/1S7awWyMw2-JGPkh6TJ5kHdVEiJnAAWj1?usp=sharing

For the first part of the problem which is the base scheduling, Gantt chart and CPM techniques provide a fine visualization, bar chart and network diagram, respectively, of the overall project task flow that is beneficial for the managerial decision-making process. While Gantt was implemented via Microsoft Excel, CPM network diagram was created by hand which is a significant drawback especially for more complex project breakdowns that is mostly the case in real life but in this sample case the assumptions and simplifications made allowed CPM to succeed relatively easily. Both of these techniques were able to come up with the time needed for the project completion. In fact, luckily the project completion time was directly visible on the Gantt chart. On the other hand, the CPM network needed to be mathematically formulated in order to obtain the project completion time, which is again not much feasible by hand calculation, still works alright for basic cases. Although, in this calculation no software was employed, the CPM formulation logic can be considered as a variation of LP and if LP were practiced, the computational time and effort would be way less. There is not much to discuss about the base scheduling problem as it has too many simplifications and is far from a real-life business case but still added to the analysis for the sake of wholeness.

For the second part of the problem, which is scheduling with crashing, the Gantt chart technique is not capable of finding the results that must be in the form of the time cut from each activity since it is just a visualization mode rather than a calculation procedure. CPM method still works fine and secures the solution nevertheless the same struggle exists that solving a more complex and realistic scheduling with crashing problem with CPM by hand would be very much time consuming, it took hours to solve the case problem with 25 activities, and even for some instances not possible to be successful without a computational solver support. LP technique addresses this issue quite efficiently, the problematic nature of hand calculation, that with the help of the IBM ILOG CPLEX Studio tool, the computation time decreases substantially. However, now with LP, the significant effort lies on the modelling and eventually coding the solver which is not so difficult for unsophisticated cases but would be puzzling for real life ones. One of the main reasons is that objective function and constraints have to be linear which can be found in the literature review part and observed in the LP model that is designated for the case. It is a quite critical aspect in terms of modelling real life situations. In addition, linear programming is able to deal with one single objective function. In such a case, nonlinear programming techniques, quadratic programming, genetic algorithm, goal programming, and other nonlinear optimization techniques can be selected to express the constraints or objective functions. As seen on Fig. 20, constraint programming is also able to solve such nonlinear problems. However, taking into consideration the aim of comparing LP and CP, the practical case that is discussed has no nonlinear element for part - 2.

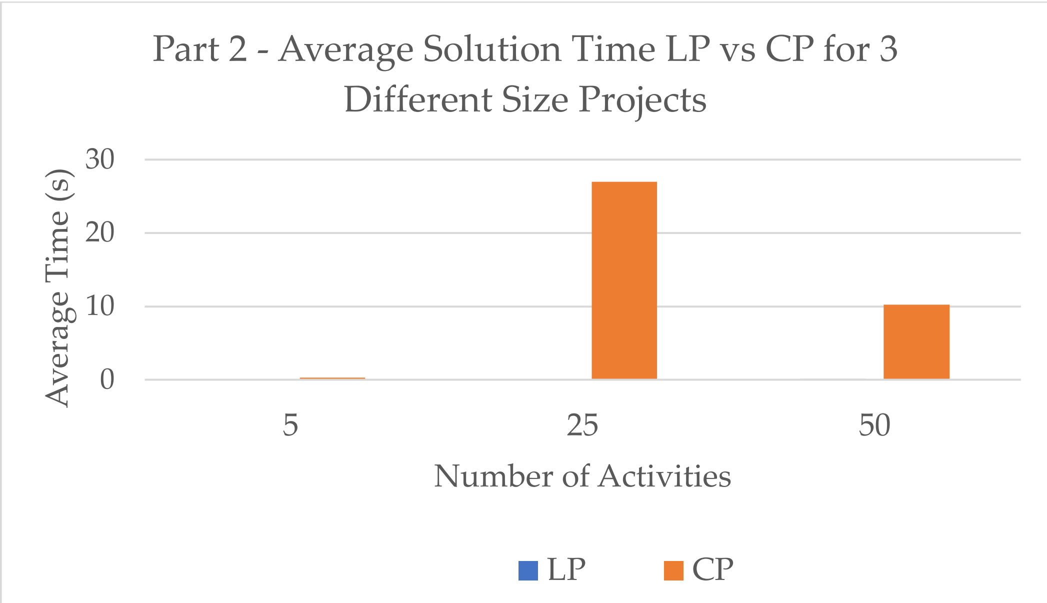
In order to shorten the duration, an additional budget is required with an aim of keeping it minimum considering the fundamental project constraints, which has been explained in methodology. The minimized cost values to satisfy the corresponding target completion time of the project and the performance metrics of the techniques applied to the problem have been illustrated in Table 11, Annex – C.2. for 5, 25 and 50 activities, respectively. With changing the target time parameter, the cost-time tradeoff can be seen by a project manager to assess the best condition for the project as proposed on the decision modelling framework on Fig. 3. As discussed in the methodology part, the parameters have been alternated until the infeasible value.

Table 11: Part - 2 solution and performance outputs

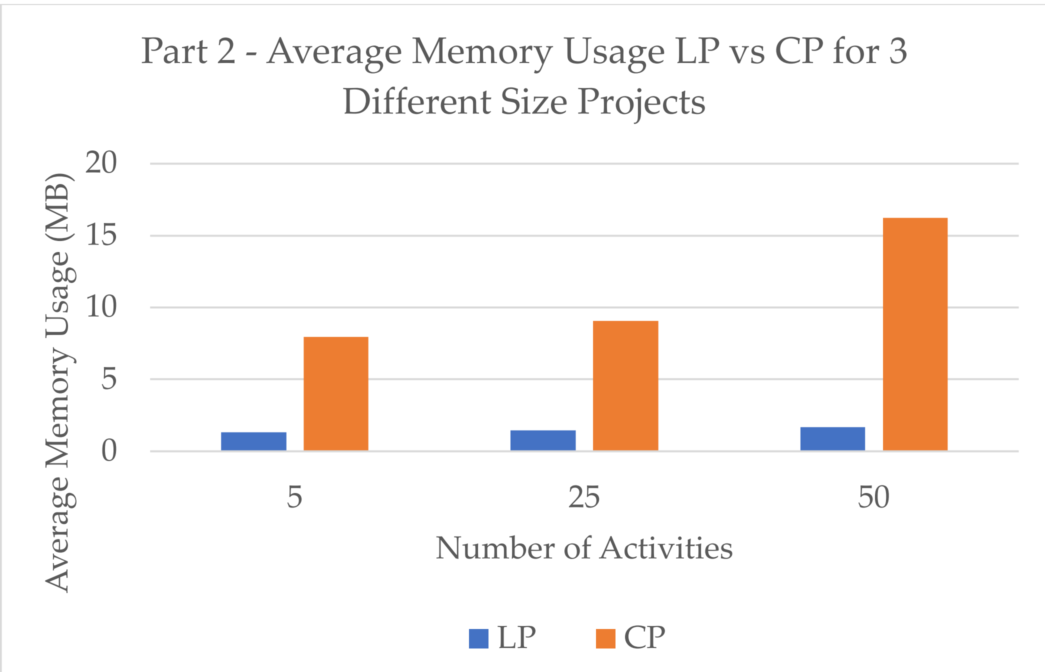
T_actual = 284								LP				CP			
	Target time T	Cost (€)	Cut_A	Cut_B	Cut_C	Cut_D	Cut_E	V	C	T (min.sec.ms)	M (MB)	V	C	T (min.sec.ms)	M (MB)
5 tasks NO RESOURCE CONSTRAINT	232	1263,2	0	40	0	8	12	21	28	00.00.07	1,29	21	32	00.00.45	8,86
	224	1519,2	0	40	0	8	20	21	28	00.00.07	1,44	21	32	00.00.50	9,13
	216	1775,20	0	40	0	8	28	21	28	00.00.06	1,17	21	32	00.00.51	9,40
	208	2021,2	0	40	0	8	36	21	28	00.00.07	1,36	21	32	00.00.39	8,64
	200	2287,20	0	40	0	8	44	21	28	00.00.07	1,48	21	32	00.00.18	9,13
	192	2543,2	0	40	0	8	52	21	28	00.00.04	1,20	21	32	00.00.15	8,41
	184	2835,68	8	40	0	16	52	21	28	00.00.04	1,28	21	32	00.00.18	9,43
	176	3195,6	16	40	8	16	52	21	28	00.00.04	1,38	21	32	00.00.23	7,75
	168	3555,52	24	40	16	16	52	21	28	00.00.03	0,96	21	32	00.00.09	6,47
	160	infeasible	-	-	-	-	-	21	28	00.00.10	1,50	21	32	00.00.29	2,42
										0,06	1,31			0,3	7,96
										average in s & MB				average in s & MB	

For example, as found in the part - 1, the project makespan is 284 hours without any crash for the first data set, 5 activities. On the other hand, the project can take a minimum 168 hours with maximum crash, there is no solution under these hours. It can be interpreted that allowable crash amounts from each activity are applied.

Both techniques are exact and give the same results for the cost objective function. However, their performances are remarkably differentiable in terms of time and peak memory usage. Although the linear programming is limited to model real life problems and does not support discrete optimization, the performance metrics show that LP has a significantly superior advantage in terms of computation runtime and peak memory usage for both problem sizes, which can be viewed on the Fig. 57. The reason is that it uses simplex algorithm unlike there is no explicit algorithm in CP and it relies on search and propagation.



(a) for average solution time



(b) for average memory usage

Figure 57: LP vs. CP numerical performance comparison

In the graph, it can be seen that there is no sharp change in both metrics when the problem size is multiplied for linear programming. Unlike, constraint programming consumes vast amounts of time with the increase of activities from 5 to 25. Especially for the data set - 2, when T_{TARGET} is 350 hours and further, CP cannot find the solution until cutoff time. However, for curiosity, it has been left to run the code and the run time took has taken 7 hours and 50 mins approximately in the case of completing the project in 420 hours. The further values cannot be found within 10 hours for 430-, 440- and 450-hours project makespan. Therefore, the values have not been exactly taken for these durations due to its impracticality and inefficiency. It is quite interesting that one can think that it should be harder to find the solution while trying to make the duration shorter. For example, if the project is able to finish in 400 hours, a schedule can be easily found for the project which ends within 450 hours. However, it is contradictory to the search mechanism of constraint programming. It focuses on constraints and variables rather than objective function. In other words, it tries to reduce a vast number of viable solutions to a smaller subset by imposing constraints to the problem. When there are many possibilities, it slows down since it tries to decide which alternatives are likely to be the most optimal. It works better for tightly constrained problems to reduce the search space as much as possible by removing inconsistent values, which is also known as pruning. It also can be verified with that when the T_{TARGET} is given as 340 hours, CP solves the problem in 16 seconds, as seen on Table 12.

Table 12: A representation for the relationship between tighter target and CP solvability

T_target	CPU
450	15+ min
420	15+ min
390	15+ min
360	15+ min
350	15+ min
340	16 s

Another discussion is that the bar chart Fig. 57a shows that the number of activities is not the only parameter that affect the time performance of the methods. All instances of 50 activity project problem have solved without cutoff unlike the medium project size. Depending on the increasing size of the problem, the number of variables and constraints grow, and the size of search space enlarges.

Optimization Studio User Manual expresses that the CP Optimizer uses two techniques to find a solution which are search heuristics and constraint propagation. The built-in search is carried out based on constructive search and uses other predefined heuristics by the solver to improve the solution. Also, the domains of variables are reduced with constraint propagation, and it reduce the search space however the initial size of the problem is still a dominant factor in the execution time. Considering that, in the part - 2, there are only two types of input data which are T_{TARGET} and the number of activities, an estimation about the problem size can be done. In such a way that these two inputs which are the average of T_{TARGET} taken from all instances considered in part - 2 and the corresponding activity size have been multiplied can be seen on Table 13. In the view of this estimated correlation, when the estimated size of the problem grows approximately 9 times, the average time needed for the instances of the corresponding data set increase 90 times. Similarly, moreover, the approximate proportionality holds for the data set - 3. It can be understood that there is $O(n^2)$ complexity for the problem which means complexity is directly proportional to the square of the input size.

Table 13: Complexity estimation and responding execution time

	Activity Size	Average T_{TARGET}	Estimated Problem Size (ave. T_{TARGET} * #ofActivity)	Average Time
Data – 1	5	196	980	0,3
Data – 2	25	378	9450	27
Data – 3	50	58	2900	10,2

Therefore, CP should be used smartly to improve the performance of it such as if the problem is formulated with time scaling. Using day unit instead of hour, which makes search space narrower considering that it works with discrete values can be a possible approach for reducing number of intervals. So, the performance could be enhanced considering the reduction on the problem size.

In the part - 3, resource constraint has been added to the problem, so a schedule has been looked for a project crashing problem under worker and machine constraints. With the additional resource constraint, it is required to search for solutions that fit into the capacity limit considering that there is no infinite resource in a project environment which has not been taken into consideration in the previous parts of the case. Therefore, satisfying the resource constraints needs to determine the total usage of the resource in an instance, which can be defined with discrete values like 0 and 1. In other words, a drawback has been observed that linear programming does not support a formula to check whether usage of a resource is present or not. Therefore, it is not possible to solve discrete optimization problems with LP, also accepted by the literature. However, mixed integer programming is one of the common discrete optimization techniques and a main competitor of CP considering the area of exact discrete optimization that has been discussed in the operation research section of the literature review.

The Table 14 represents the required cut amounts from each activity and corresponding cost values with respect to the change of instances in terms of target time, number of workers or machines. Moreover, the results of the other project sizes can be seen on Annex - C.3.

At this point, let us also examine the outcomes of the side study of this case analysis, which is demonstrating how these solution techniques, especially CP, are able to assist the decision-making process by managers and it can be best observed from the part 3 problems of the case. Firstly, for the 5 activities with C1 and C2 is fixed to 13 and 6, respectively and project completion time is varied between 232 and 184 by 8, the only visible managerial result is the total cut off cost depending on the target duration. In this manner, a manager by looking at the associated results table should be able to conclude that the cost to shorten the project completion time by 8 units is almost the same in the gap between 232 and 192. Not only optimal solutions are seen but also some infeasible cases can be seen such as impossibility of completing the project with 13 workers and 6 machines in 184 hours frame. It is not likely to complete the project within 184 hours even with taking advantage of maximum allowable cut off rates from each activity. Secondly, again for the 5 activities but this time with the fixed project completion target of 224 fours while the C2 is set to unlimited, 99, and C1 is varied in between 20, which is more than what is needed in max, and 10 which is not feasible, a manager can notice the bottlenecks and trade-offs related. For instance, the cut off cost when there are 18 workers is actually equal to when there are 11 workers so that a logical conclusion would be not to assign more than 11 workers to that particular project since the additional workers are either idle or not necessary and increase the project budget to finalize the project in 224 hours. In addition, by allocating 10 workers to the activities, the project could not be completed within the specified time frame with allowable cut off rates and this implies that if the team has only 10 people available for the project, the company needs to hire more or not to accept the project specifications and ask for a more flexible timeline. Thirdly, once again for 5 activities and the fixed project completion duration of 224 hours but now the C1 is set to unlimited, 99, and C2 is altered in between 8, which is more than maximum needed, and 5, which is not possible to make it.

Table 14: Alternatives for the different instances for 5 activities project

	Parameters			Variables					
5 tasks T_actual = 284 C1_actual=19 C1_available=13 C2_actual=7 C2_available=6	**T_target**	**Worker C1**	**Machine C2**	**Cost (€)**	**Cut_A**	**Cut_B**	**Cut_C**	**Cut_D**	**Cut_E**
	232	13	6	2270,24	24	16	0	16	36
	224	13	6	2526,24	24	16	0	16	44
	216	13	6	2782,24	24	16	0	16	52
	208	13	6	3088,88	24	24	8	16	52
	200	13	6	3395,52	24	31	16	16	52
	192	13	6	3702,16	24	40	24	16	52
	184	13	6	infeasible	-	-	-	-	-
	T_target	**Worker C1**	**Machine C2**	**Cost (€)**	**Cut_A**	**Cut_B**	**Cut_C**	**Cut_D**	**Cut_E**
	224	20	99	1519,2	0	40	0	8	20
	224	19	99	1519,2	0	40	0	8	20
	224	18	99	2238,24	24	40	0	16	20
	224	17	99	2238,24	24	40	0	16	20
	224	16	99	2238,24	24	40	0	16	20
	224	15	99	2238,24	24	40	0	16	20
	224	14	99	2238,24	24	40	0	16	20
	224	13	99	2238,24	24	40	0	16	20
	224	12	99	2238,24	24	40	0	16	20
	224	11	99	2238,24	24	40	0	16	20
	224	10	99	infeasible	-	-	-	-	-
	T_target	**Worker C1**	**Machine C2**	**Cost (€)**	**Cut_A**	**Cut_B**	**Cut_C**	**Cut_D**	**Cut_E**
	224	99	8	1519,2	0	40	0	8	20
	224	99	7	1519,2	0	40	0	8	20
	224	99	6	2526,24	24	16	0	16	44
	224	99	5	infeasible	-	-	-	-	-
	224	99	4	infeasible	-	-	-	-	-

The managerial notion could be that there is no meaning to employ more than 7 staff for the project with these requirements since the cut off cost remains same anyway and also it is not possible to make the project delivery in the right time with 5 workers or less. Furthermore, the cut off cost is 1519,2 € for 7 workers and 2526,24 € for 6 workers so, a rational takeaway is that if dedicating 1 more worker up to 6 is cheaper than 1007,04 € (2526,24-1519,2) then it is a good choice but if not then it would be a false business decision in terms of expenditures. Similar reasoning breakdowns can be also done with respect to the part 3 problems with 25 and 50 activities of the case. Therefore, it has been seen that constraint programming, as well as the other techniques, can be utilized as a decision support application as discussed in decision flowchart Fig. 40.

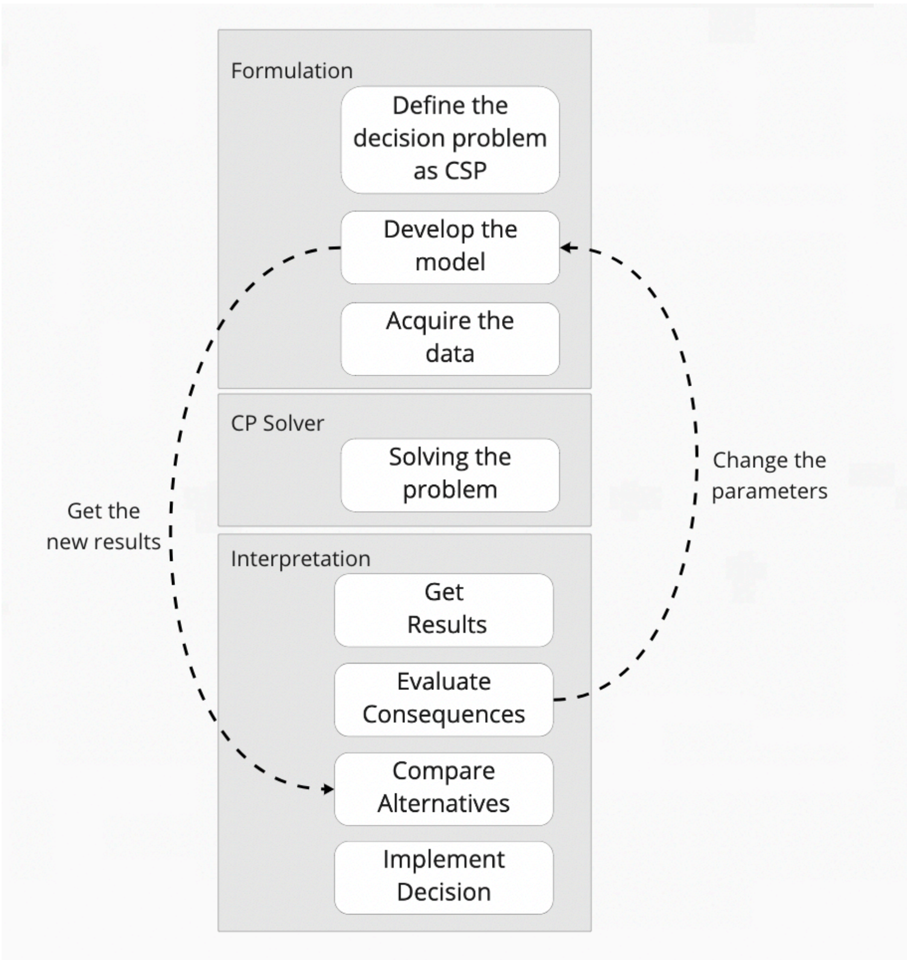


Figure 58: Proposed decision-making flowchart

A problem modeled as CSP can be solved with CP efficiently with a CP solver. It enables the user who can be a project manager, to make a what-if analysis and evaluate the business alternatives such as purchasing a new equipment or upgrading the existing one; or hiring a worker or outsourcing the job while maintaining the cost at minimum and obeying the constraints.

In order to prove why CP is *efficient* or why it is *promising*, the performance values have been examined and compared while the systems were performing their functions, which tabulated under the ‘solving with MIP’ and ‘solving with CP columns’ on the Table 15 which represents the average time and memory need for 23 instances for 5 tasks projects. Also, the time and memory performance of the MIP and CP can be found for the projects with 25 and 50 activities on Annex – C.3. As indicated in the methodology section, the V and C letters indicate the number of variables and the number of constraints, respectively. In addition, the T and M express the time and memory requirement, respectively.

In terms of execution time, there is a slight difference between the techniques and the performance of both ones are quite successful. On the other hand, it is not surprising that memory usage of MIP is much higher than CP considering that it works with large number of variables (V) and constraints (C). For instance, taking a benchmark instance for 5 activities project including 13 workers and 6 machines with the objective cost minimization, the MIP solver has to take 4725 constraints and 5180 variables into account, but the CP solver only considers 11 constraints and 7 variables. The main reason is that it tries to indicate the attribute, that was resource usage in this case, is absent or present by checking each time intervals for each task. For example, an *interval* variable has replaced 3 decision variables of the MIP model, that is *indicate* variables to represent the presence status of a task disjunctively. It means creating one decision variable for each time instance *i*. To put it clearly, in the case, MIP requires three matrix whose dimensions are [*i* * *t*], where *t* represents 7 tasks and *i* is for the time instances from 1 to project duration, 224, to specify the status of the task at a moment. How much indication is required can be seen on the Fig. 50 which is represented in the developing the model sub-section.

In CP, the flexibility and expressiveness can replace large number of variables and constraints usage which is needed in traditional mathematical programming. The advantages come from several features such as the compatibility with global constraints that can express relationship between multiple variables. Moreover, as a strict fact MIP do not recognize equality operators such that to express $x_1 = x_2$, it is needed to use $(x_1 \leq x_2)$ and $(x_1 \geq x_2)$ unlike CP can express equalities. Considering these obstacles, MIP use more variables and constraints which occupy more memory to formulate the problem when it is compared to CP.

Table 15: Performance outputs for the different instances for 5 activities project

	Parameters			solving with MIP				solving with CP			
	T_target	Worker C1	Machine C2	V	C	T (min.s.cs)	M (MB)	V	C	T (min.s.cs)	M (MB)
	232	13	6	4893	5364	00.00.84	39,402	7	11	00.01.09	13,52
	224	13	6	4725	5180	00.00.86	21,19	7	11	00.01.03	14,418
	216	13	6	4557	4996	00.00.89	20,81	7	11	00.00.80	14,191
	208	13	6	4389	4812	00.01.21	19,05	7	11	00.00.68	11,582
	200	13	6	4221	4628	00.00.45	15,60	7	11	00.00.74	13,125
	192	13	6	4053	4444	00.00.31	7,246	7	11	00.00.25	7,227
	184	13	6	147	166	00.00.09	1,574	7	11	00.00.34	2,129
				MIP				CP			
	T_target	Worker C1	Machine C2	V	C	T (min.s.cs)	M (MB)	V	C	T (min.s.cs)	M (MB)
5 tasks T_actual = 284 Unlimited C1 = 19 C2 = 7 Available C1 = 13 C2 = 6	224	20	99	4725	5180	00.00.35	13,578	7	11	00.00.26	8,246
	224	19	99	4725	5180	00.00.37	12,988	7	11	00.00.21	8,535
	224	18	99	4725	5180	00.00.40	23,168	7	11	00.00.46	11,055
	224	17	99	4725	5180	00.00.42	29,902	7	11	00.00.62	10,301
	224	16	99	4725	5180	00.00.41	31,566	7	11	00.00.42	10,934
	224	15	99	4725	5180	00.00.39	29,27	7	11	00.00.43	11,754
	224	14	99	4725	5180	00.00.45	31,29	7	11	00.00.45	9,980
	224	13	99	4725	5180	00.00.39	30,50	7	11	00.00.43	11,48
	224	12	99	4725	5180	00.00.39	21,41	7	11	00.00.53	11,367
	224	11	99	4725	5180	00.00.41	32,94	7	11	00.00.65	10,57
	224	10	99	4725	5180	00.00.09	7,24	7	11	00.00.11	1,961
				MIP				CP			
	T_target	Worker C1	Machine C2	V	C	T (min.s.cs)	M (MB)	V	C	T (min.s.cs)	M (MB)
	224	99	8	4725	5180	00.00.31	12,336	7	11	00.00.38	9,02
	224	99	7	4725	5180	00.00.37	14,031	7	11	00.00.18	8,574
	224	99	6	4725	5180	00.00.35	15,941	7	11	00.00.86	12,789
	224	99	5	4725	5180	00.00.14	8,289	7	11	00.00.17	8,387
	224	99	4	4725	5180	00.00.09	7,508	7	11	00.00.06	2,133
						0,8	19,43			1,3	9,71
						Ave. in s & MB				Ave. in s & MB	

Table 16 shows computational time results for data set - 2 that there is not a regular trend in terms of run time. For example, from instance (440/25/99) to (440/24/99) for the dataset – 2, i.e., defining 24 workers instead of 25 workers to the model creates a remarkable performance issue on CP.

Table 16: A representation for significant sensitivity to input data

Target time T	Worker C1	Machine C2	CPU (min.s.cs)
440	25	99	00.00.22
440	24	99	15 min+
440	23	99	15 min+
440	22	99	15 min+
440	21	99	15 min+
440	20	99	15 min+
440	19	99	15 min+
440	18	99	15 min+
440	17	99	15 min+
440	16	99	15 min+
440	15	99	15 min+
440	14	99	15 min+
440	13	99	15 min+

In other words, the instance (440/25/99) can be solved within a second with CP however the instance (440/24/99) exposed to cutoff due to the defined time limitation. However, for these instances that take at least 15 mins for CP, mostly solved without any cutoff with MIP. The reason is that in MIP, large sections of search space is ignored with using directed search algorithm Branch and Bound and some sophisticated mathematical properties of the program such as symmetry detection, node presolve [42],[43]. On the other hand, CP solvers works with constraint propagation which can be summarized that implementing partial solutions and evaluating the effect on the rest of the solution space, as indicated in the how it works section of the literature review. Unless there is a feasible complete solution, the solver backtracks and try a part of the partial solution to find another potential solution. Therefore, as also seen from the results, CP provide better performance on optimization by imposing more restrictive constraints on the value of the objective. In short, CP and MIP do not apply the same optimization strategies. Therefore, the choice of model and the value of the parameters greatly affects the ability to solve the problem in a reasonable time. The further reasonings and knowledge can be obtained in [7] to observe the importance of parameters and the complexity of the problem related with "phase transition" concept.

Palpant declares that 60 activities resource constrained project scheduling problems are no solvable only with exact methods such as mathematical programming techniques and constraint programming and the use of heuristics are quite needed to tackle the project problems with several activities [49]. In parallel, for some instances, a solution cannot be found within 15 minutes by two methods which has been presented in Table 17. There are some feasible solutions however there is no evidence that it is the optimal solution that makes the objective function minimum among all alternatives. In such a case, an evaluation can be done according to the solution which is presented within 15 minutes.

Table 17: The best solutions by MIP and CP for time-off cases

Target time T	Worker C1	Machine C2	**Best sol. by MIP**	**Best sol. by CP**
65	23	99	225,8	185
65	18	99	620,03	483,98
65	99	30	259,78	218,58
65	99	25	550,76	516,67

In problem solving, a trade-off exists in between execution time and the solution quality. Two kinds of solvers are used to solve models, namely, exact algorithms and approximate algorithms. In the literature, in general, constraint programming is called as an exact algorithm which can always generate the solution of the model. However, in this case the size of the solvable models could be insufficient for business needs. At this stage, heuristic approaches come to stage to fasten the search and therefore they can solve bigger problems, however, the quality of the solution cannot be guaranteed, usually. So, in case of heuristics gets involve, rather than the ability to solve, the quality of the solution would be affected significantly. The tabulated results show that for all instances, CP gives better results because the objective function was for minimization. In other words, CP was able to find a schedule that satisfies the constraints and keeps the cost more minimum than the ones found with MIP. As an inference, a faced problem in a project should be tried to solve with various models.

Due to the methodology applied that is when there is a cutoff, time to optimal solution is considered as T which indicates time to solution for the problem, the Fig. 59 represents the superiority of CP over MIP for all data sets. However, it is strongly related with the selected methodology because among the 105 instances, CP has exposed to cutoff for 28 instances, and it was just 4 for MIP. However, it can be seen from the engine log section that it finds the optimal solution within one second for all these instances, but it was no able to prove it. Therefore, CP has better results in terms of time to solution. Although there is a slight difference for 5 activities project, the average time difference enlarges with number of activities.

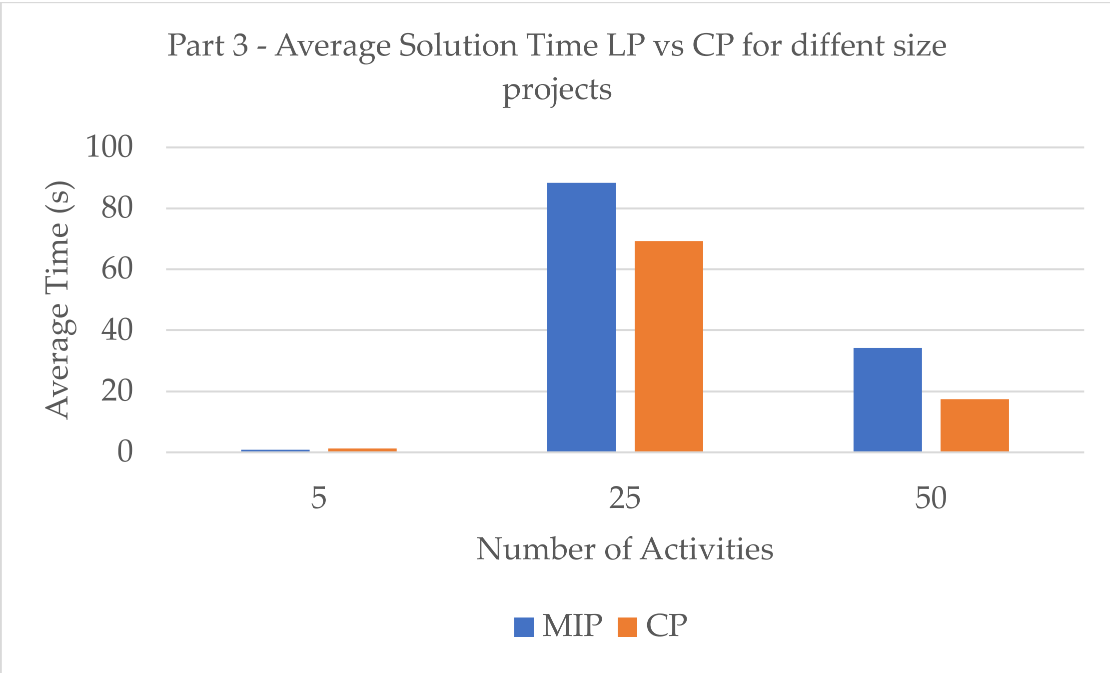


Figure 59: Average solution time vs. Approaches for different project sizes

In the literature, CP is mainly used for not only finding solutions that satisfy the problem but also showing the unsatisfiability of the problem. It means there is no combination of values for the decision variables that satisfies all the constraints. Therefore, the performance of the methods for finding the infeasibility is tabulated on the Table 18. It can be seen that CP give better performance for finding infeasibility and feasibility rather than optimization because it focusses on the constraints and variables rather than the objective function.

It has been experienced that due to the several factors, time to solve the problem is affected significantly and no exact trend or a dominance of a method has been detected. Also, indicated in IBM'S information document, the reasons could be related to the size of problem, the characteristic of the model, the characteristics of the data, the default algorithm parameters settings and used algorithms and the hardware used. Moreover, IBM clearly indicate that making a generalization about how long a given problem will be solved is no meaningful. Although, in the literature, there are several comparison studies for MIP and CP, they can be interpreted just for specific problem formulation and specific algorithms, without making induction as a common sense.

Table 18: Time and memory requirement for finding infeasibility

Parameters			**Objective**	**MIP**		**CP**	
Target T	**Worker C1**	**Machine C2**	**Cost**	**T (s)**	**M (MB)**	**T (s)**	**M (MB)**
380	17	12	infeasible	0,54	26,67	0,13	11,73
440	99	6	infeasible	1,27	32,16	1,51	11,97
440	10	99	infeasible	0,45	44,71	0,12	2,56
65	99	20	infeasible	4,42	95,50	0,75	11,54
65	13	99	infeasible	15,9	114,63	1,79	9,77
45	28	46	infeasible	0,12	11,11	0,29	2,60
65	99	20	infeasible	4,42	95,50	0,75	11,54
184	13	6	infeasible	0.09	1,57	0,34	2,13
224	10	99	infeasible	0.09	7,24	0,11	1,96
224	99	4	infeasible	0.09	7,51	0,06	2,13
			average:	3,9	43,7	0,6	6,8

Also, it is hard to examine every step to make reasonings because the logic and mechanism behind the used commercial software is a kept as an intellectual property by developer companies [105].

In order to evaluate the influence of the techniques with the different data sets, the memory requirement has been tabulated on Fig. 60 with the average values of 3 project sizes to compare the performance of techniques for the project crashing problem under resource constraint. It can be interpreted that under a limited memory capacity, MIP can restrict the problem size which can be handled by the solver and can face with a memory failure.

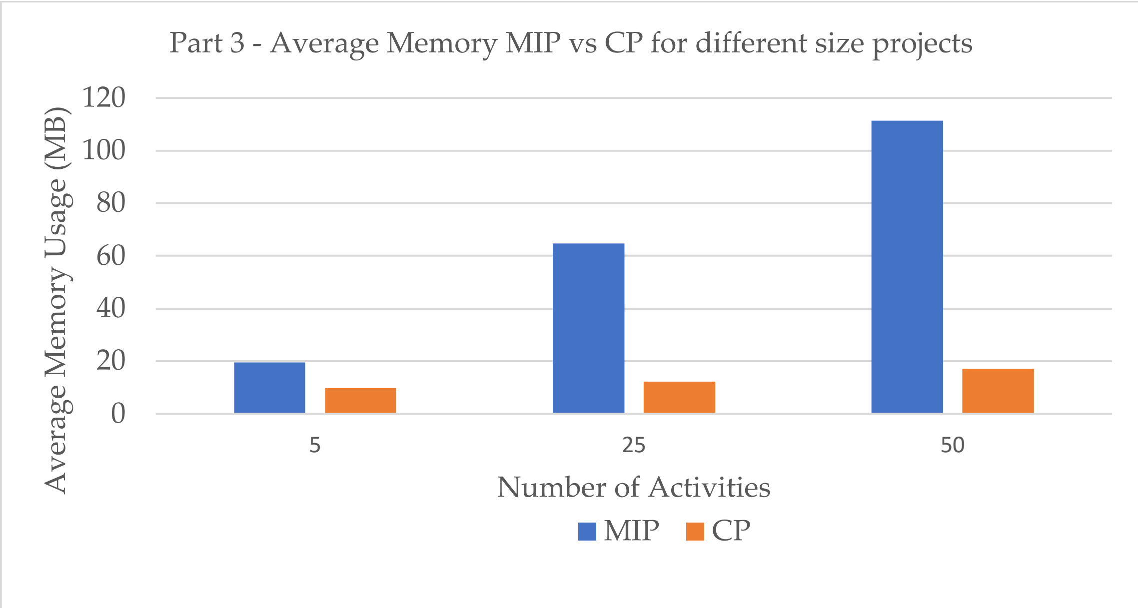
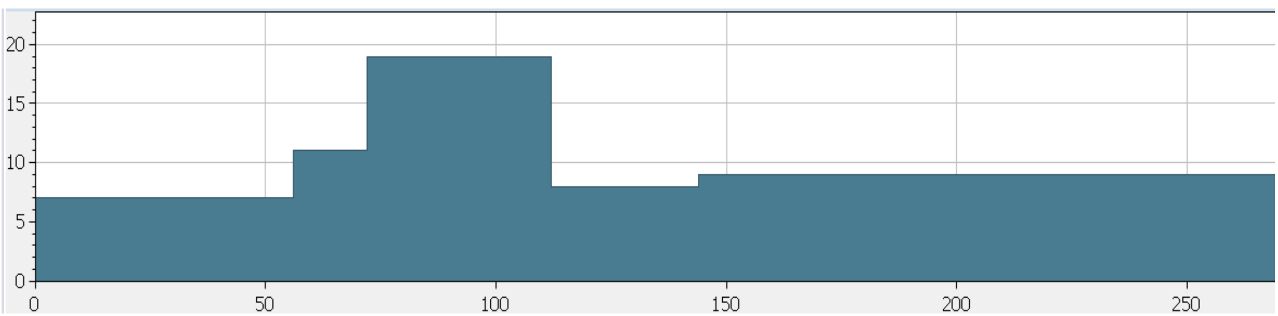
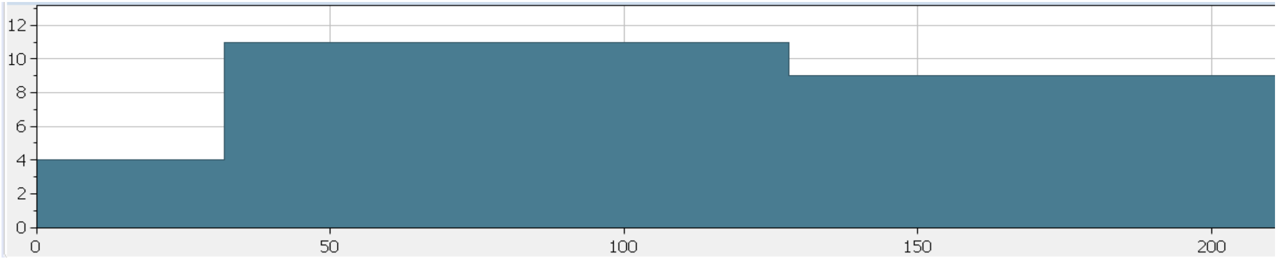


Figure 60: Average memory usage vs. Approaches for different project sizes

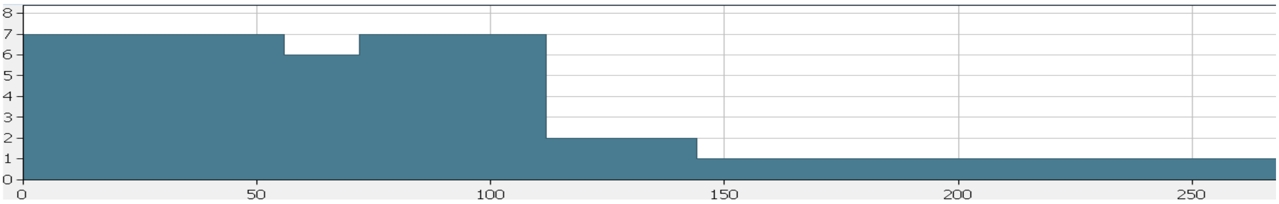
Moreover, there is another important benefit for using constraint programming in the problem, which is not available in the MIP solution. CP Optimizer provides some useful project management charts and graphs such as resource load profiles and Gantt chart, which have been represented on Fig. 61 and Fig. 62, respectively for no crash case and the instance (224/13/6). During the time interval, the time-resource profiles and Gantt chart are at hand that can be useful for a project manager. Although it looks very simple for 5 activities, more crowded and realistic ones can be seen for 25 and 50 activities project on the Annex – D.



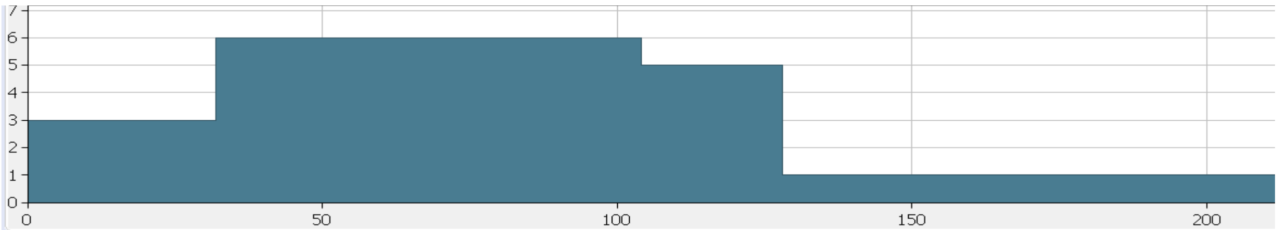
(a) C1 resource load in unlimited case



(b) C1 worker load with constraint defined

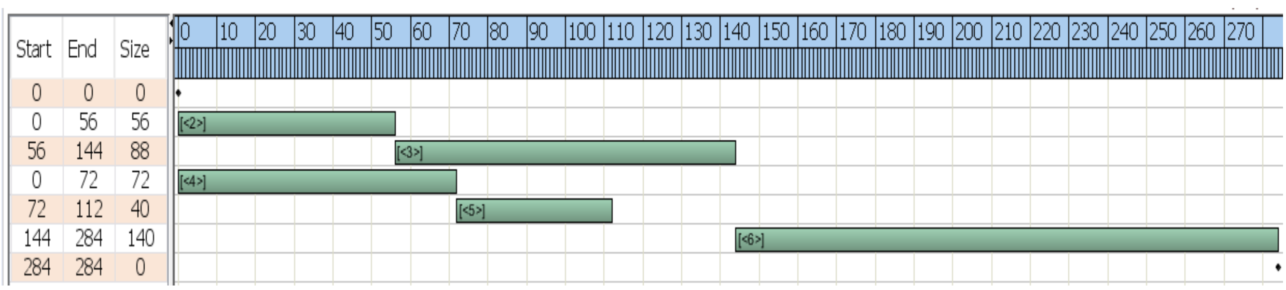


(c) C2 resource load in unlimited case

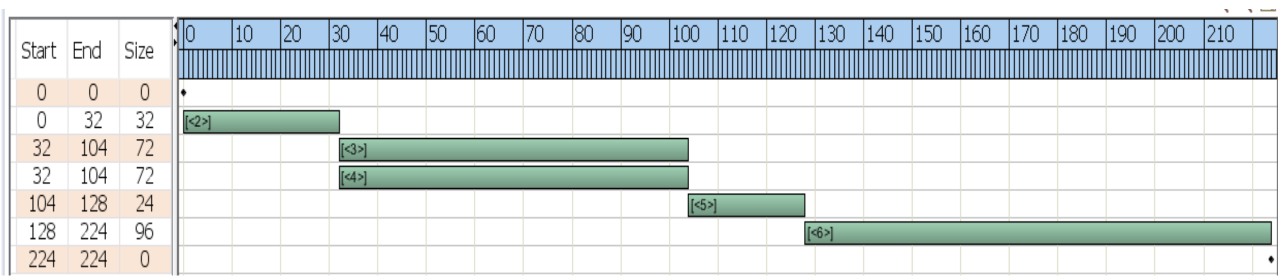


(d) C2 resource load with constraint defined

Figure 61: CP Supplementary Resource Load Profiles



(a) Gantt Chart in unlimited case



(b) Gantt Chart with target time defined

Figure 62: CP Supplementary Gant Charts

Besides, the qualitative reasons point out why CP is promising in PM especially for solving project scheduling problems and supporting the related decisions. First of all, the natural formulation of constraint programming is not far from the real life and to problem description than the mathematical programming such as MIP and LP formulations. It is not only related with expressiveness of the models, but also it has a high-level declarative language for modelling. It can be acceptable as one of the strengths of CP because it is like spoken language like no one say go to the kitchen, open the cupboard, take a glass, tap etc. for asking for a glass of water. Moreover, CP's syntax and keywords are more familiar with natural language such as *cumulative, size, endOf* which are already existing words in our daily language. Moreover, when the models of part – 2 and part – 3 have been compared, it can be seen that CP have a flexible and extensible nature since adding a new problem dependent constraint is possible without having to modify or change the solution strategy as parallel with the declared advantages of CP over MIP [67]. Unlike, a method change has required from LP to MIP in the case of resource constraints. CP can work with non-linear, arithmetic, higher-order or logical constraints as well as supports the global constraints for express some substructures easily such as cumulative, all different, sequence etc. In other words, CP formulation provides built-in constraints and functions for specific purposes such as expressing precedence relations and presence status which are commonly used while dealing with resource allocation, sequencing, scheduling problems.

In fact, it is not surprising that a project manager would prefer to not write a complex coding considering his/her educational background, expertise area and competences. With a basic modeling and programming knowledge, it is possible to automatically solve project problems in CP. In view of the fact that CP packages allows the use of different powerful concepts depending on the requirement. In other words, the user spends the main effort to describe the problem in CSP or COP, not solving it because the CP engine selects the proper search algorithms and heuristics are automatically from MP, CP, metaheuristics etc. In such a way, it is possible to change parameters, evaluate the results and compare the alternatives to select the most fruitful one for the project success by a project manager. Thus, CP is advantageous compared to other two techniques in the means of easiness of modelling. The process of decision making and problem-solving with CP has been illustrated in the flowchart, on Fig. 40.

Before evaluating the performance framework for the models, some drawbacks should be discussed to draw an objective portray for CP. There is no model or technique exist to solve every problem efficiently, each has different characteristics. The advantages and disadvantages should be carefully investigated and analyzed. Firstly, in real word, project activities are subject to change, and it can gradually become clear during project execution. Their duration or even, their presence can be uncertain. However, CP does not consider the uncertainties on the problem. To deal with uncertainty, stochastic optimization techniques are selected. Secondly, although continuous expressions such as time or cost can be used, the continuous expressions must be computed from discrete decision variables, CP can work limited with continuous decision variables. Also, it must be highlighted that the syntax, some of special variables and keywords are specific to IBM's CP package and subject to change in other software. CP has no standard formulation across packages, unlike mathematical programming. Therefore, according to selected package, syntax should be reviewed.

All in all, in terms of time and memory usage LP has dominance over other two techniques however it can be only used for continuous and linearly constrained optimization and cannot for combinatorial optimization and resource constraint scheduling problems while MIP and CP can. In other words, for our case, CP provides a model for the whole scheduling problem whereas LP only describes the tasks and temporary network diagram without the constraint of resource capacity. Therefore, if the nature of problem is suitable with LP, decision making process can be supported with it.

Furthermore, both LP and MIP do not provide any additional complementary materials apart from just displaying numerical solutions. On the other hand, CP is offering some helpful visualization graphics such as Gantt charts, resource diagrams, etc. and managers can highly benefit from these while making complicated decisions where there is not an obvious one and only solution which is almost always the situation in real life.

In the light of these discussions, the designed framework explained in methodology part in detail has fulfilled with the findings from literature and case on Table 19. The framework shows an overall picture of performance of LP, CP and MIP techniques from the viewpoint of the quantitative metrics, time and memory usage, as well as the qualitative metrics, ease of modeling, closeness to natural language and availability of complementary materials.

Table 19: Analysis results evaluations for different approaches

	Part - 2		Part - 3	
Performance Metrics / Solution Techniques	**LP**	**CP**	**CP**	**MIP**
Time	√	?	?	?
Memory Usage	√	√	√	X
Ease of Modeling	X	√	√	X
Closeness to Natural Language	X	√	√	X
Complementary Materials Availability	X	√	√	X
Total Score	2	4,5	4,5	0,5

*√ is 1 point, ? is 0.5 points & X is 0 points

CP has the highest total score with 5,5 while LP scores 2 and MIP gets 1,5. Indeed, CP dominates the comparison results because of its high performance on qualitative metrics mostly, as LP and MIP is equiparable on the quantitative metrics if not better. Although there are various studies that compare the quantitative criteria of the techniques, from project managers perspective, qualitative performance criteria of a technique are as critical and remarkable as the quantitative ones since the latter is mostly dependent on the computational performance such as applied heuristics, algorithms, power of computer and solver which are progressing exponentially over the years whereas the qualitative means in general directly affects the managerial effort and time. So, the framework represents a clear picture that CP is useful for managerial problem solving and decision making.

To sum all the case analysis up, in project scheduling with crashing under resource constraint case presented in the thesis, it is verified that a what-if analysis can be effectively performed to examine the influence of conditions, which was the target time, the number of machines and workers, on the cost of crashing of the project. So, CP can be used as an alternative to other optimization techniques in project management. Even, in order to get better performance results in terms of soliton quality and time while making analytical decisions efficiently, hybrid methods can be used that integrate CP and OR techniques. There are several examples show that hybrid methods which able to complement each other's powerful elements such as the ones indicated in the literature [54] [60] [62]. As another alternative, CP solutions can be enhanced with machine learning and data mining utilization. A combination of ML and constraint satisfaction outperforms than the individual AI techniques [80], [40]. Since generally one of the most challenging concepts of CP is formulating the constraints explicitly thus, ML can learn from the data and experience to provide automatically formulated constraints. So, it can be deducted that there are lots of alternatives available to support problem solving and decision making. A project manager should know about them to be able to select the accurate tools according to the needs.

4 Conclusion

4.1. Research Conclusion

The literature survey and case study analysis have been examined detailly and the combined findings have been presented in the previous part for constraint programming potential support to project management problem solving and decision making within a limited resource and time environment. So, as a short recap and resume, the project management realm has tremendously evolved in the recent past with the technological developments especially in the field of artificial intelligence. AI can be extremely helpful to solve project management problems and help the managers to make complex business decisions efficiently and effectively. However, one of the tough challenges to enable this mechanism of AI utilization on PM is choosing the right technique and building up the appropriate models. Indeed, the right solution technique is obviously highly dependent on the PM application, for example scheduling, resource allocation, etc. in question.

On the other hand, there are also different other project management exact problem-solving techniques such as mathematical programming, dynamic programming or hand calculation approaches devised even before AI. In this manner, the primary finding of this study is that CP, which is considered as a practical application method of AI, can be used to deal with project management problems in order to assist the managerial decision-making process and even has several advantages compared to the others able to do the same job, for instance CPM by hand, LP and MIP. These advantages are mostly based on qualitative performance measures (ease of modeling, closeness to natural language, availability of complementary materials) as for the quantitative ones (computational time, memory usage), mathematical models, LP and MIP, are quite competitive or even superior. These findings have been principally derived from extensive literature search and the case study. The case study consists of designing and solving a really common type of project management problem which is scheduling, and the affiliated variations with varying constraints, inputs, requirements, complexity and etc. by applying different solution techniques CPM, Gantt chart, CP, LP and MIP. After that the performances of these solution techniques are examined based on the framework constructed with the aim of introducing a comprehensive matrix for evaluation and selection. Furthermore, apart from this, another goal of this study especially for the literature survey part, is to provide a clear positioning of constraint programming among optimization models for decision support mechanics.

4.2. Limitations

There are various limitations and complications of this study while coming up with the findings explained which might have caused an incoherency. First of all, the literature comprises of different results in terms of the positioning of the CP. For example, Leila (2018) categorizes the CP as a sub-topic of Operations Research [48]. On contrary, it is mostly accepted as an AI paradigm which is developed by AI community that can solve common problems with the area of OR.

In another classification for the view of CP as an exact or approximate method, there are several divergences. Rolland (2009) prefers categorizing constraint satisfaction in the heuristic search category [115]. Oxenstierna (2019) evaluates the topic of CP as under the metaheuristic methods in between genetic algorithm, simulated annealing, tabu search etc. rather than the category of exact approaches [45]. On the other hand, most of the studies and the thesis consider it as exact method.

Another split is related with the useability of continuous variables in CP. IBM ILOG's CP Optimizer that is used in the solved case in the thesis support only discrete decision variables although continuous expressions such as time or cost can be used. CP can work limited with continuous decision variables. In the literature, there are also unpopular some applications such as Hubble Space Telescope observations that use CP with continuous variables.

Unavailability of the solution strategies applied by the solver has resulted in avoiding digging deeper in this particular topic. In other words, CP Optimizer uses default search settings and propagates constraints with applying predefined procedures. In that sense, it can be accepted as a black box. It was hard to discuss each step to make reasonings because the logic and mechanism such as search algorithms, heuristics behind the used commercial software is a kept as an intellectual property by IBM.

Another remark is that the tool utilized to ruin the software in the case analysis is IBM ILOG CPLEX Studio and if another solver was used, the performance of the techniques might have been deviate. Moreover, the solvers have various parameters which can be set to customize and improve the computations. For our analysis, the default settings have been used however better performance may be taken by experimenting with several settings. Furthermore, only two quantitative performance measures, computational time and memory usage, have been taken into account to compare the exact problem-solving techniques in the area of project management. Although these two can be considered sufficient to come up with conclusions, adding more numerical metrics to the case study may be beneficial to increase the accuracy.

The thesis work was carried out in 10 months by a Management Engineering Master of Science student. In some points, computer science and advance mathematics knowledge had needed and therefore, there could be some lack of reasonings and further explanations about the technicalities in some points in the study.

4.3. Further Work

It is critical to mentioned about the possible future improvements on top of this study for the next researchers.

The real-life business situations are more complex than the case designed in this study thus, it can be fruitful to add up to the complexities of the case and run the solution techniques for that to analyses the performances as an extra study. There are many ways to eliminate the number of assumptions and extrapolate the case to the real-life situation. First of all, in our case, just a single objective function exists which is the minimization of the activities' cut off cost and, in this manner, a multi objective function may be introduced on top of that which could be related with the costs for the resources as worker wage or machinery purchase expenditure minimization and so on. Another improvement to the case would be setting up some additional hard and especially soft constraints over the worker and equipment constraint in the current version of the case such that the required skills to carry out the activity, as well as the employee's skills, proficiencies and availabilities. A hard constraint could be describing proficiency level to the workers, meaning that not every worker has the same capacity or competences to perform a task. Alternatively, a soft constraint can be introduced to express the preferences of the workers such as willingness to overtime. The nonmandatory activities and dynamic nature of scheduling can be also defined such as breakdowns of a machine, a cancellation of an activity, etc. In such a way, a couple of the favorable features of CP, adding additional constraints to an existing model and defining soft constraints has been illustrated. This soft constraint introduction could be particularly improvement for the sake of the results of the analysis since from the literature findings, CP has an advantage in the sense of ability to define and execute soft constraints to the code compared to other techniques.

The various problem-solving techniques can be also used to make an extensive evaluation and to see alternatives although the thesis just has focused to the MIP, LP and CP. For example, as another exact method, dynamic programming could also solve optimization problems. Alternatively, a similar problem even including stochastic can be modeled as Markov decision process or simulation.

Specific to the demonstrated decision framework for RCPSP, a basic user interface can be integrated to enter the input data such as level of resource, target completion date of the project, the parameters of the activities etc. In such a way, the tool can be more user-friendly although it is still easy-to-use due to the separation of model and data file in the IBM ILOG Optimization Studio.

In terms of applications, constraint programming is not only promising for scheduling, resource allocation or planning in PM. The further possible future paths that CP could track to improve its usage on PM. For example, in the literature there are also several CP applications for project portfolio management, configuration management, financial planning and so on. In a project, most of the time the best alternatives are tried to find while choosing a supplier, hiring a worker, or selecting a portfolio. Therefore, in general literature shows that the best solution which could be optimally or feasibly has searched. However, the opposite is not discussed as far as known from in the literature review. An attention can be given to the reverse of the purpose which is finding no solution available. It seems confusing but it could be beneficial to manifest an impossibility, which is called as infeasibility in CP, of a request. Bearing in mind that project managers are also professional negotiators at the same time. They try to reach an agreement for timelines, scope, budgets and so on especially in the beginning of the project. A requested budget or completion date can be denied with analytical results like 'no solution is found that can be taken from CP system. In such a case, the partners or opponent parts should be convinced to a reasonable option to move forward in the project. Possibly, it can be supported with proofs that are produced based on CSP results.

For the practioners, there are special expressions available in CP solvers that are developed for the specific to some problem types such as cumulative function, precedence and sequence constraints, minimum distance constraint, packing constraint. Therefore, these should be examined which can be reviewed in user manual.

At this point, it must be suggested again that there could be several problems in a project environment but now, we have several opportunities to tackle with them. Having knowledge about the tools and software can boost problem solving and decision-making processes.

Bibliogaphy

[1] J. Lustig and J.-F. Puget, "Program Does Not Equal Program: Constraint Programming and Its Relationship to Mathematical Programming," *Interfaces*, vol. 31, no. 6, pp. 29–53, Dec. 2001, doi: 10.1287/inte.31.6.29.9647.

[2] B. A. Nadel, Some applications of the constraint satisfaction problem. Detroit, Mich.: Wayne State University, Dept. Of Computer Science, 1990.

[3] D. Stang, "How AI Will Reinvent Program and Portfolio Management," Gartner, Dec. 03,2018. [Online].Available: https://www.gartner.com/en/documents/3894666

[4] McKinsey, "The state of AI in 2020 | McKinsey," www.mckinsey.com, Nov. 17, 2020. [Online]. Available: https://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/global-survey-the-state-of-ai-in-2020

[5] D. Zhang *et al.,* AI Index Steering Committee, Human-Centered AI Institute, Stanford University, Stanford, CA, Mar. 2021.

[6] B. Ghosh," Technology is business", Accenture, India, 2019. [Online]. Available: https://www.accenture.com/_acnmedia/PDF-114/Accenture-Technology-is-Business-New.pdf#zoom=50

[7] F. Rossi, Peter Van Beek, and Towalsh, Handbook of constraint programming. Amsterdam; Boston: Elsevier, 2006.

[8] P. Snijders, T. Wuttke, A. Zandhuis, and S. Newton, PMBOK Guide: a pocket companion to PMI's: a quick introduction to "A Guide to the Project Management Body of Knowledge" (PMBOK Guide), 5th ed. Zaltbommel: Van Haren Publishing, 2013.

[9] Guidance on Project Management, ISO 21500:2012(E), Switzerland, 2012

[10] S. Armenia, R. M. Dangelico, F. Nonino, and A. Pompei, "Sustainable Project Management: A Conceptualization-Oriented Review and a Framework Proposal for Future Studies," Sustainability, vol. 11, no. 9, p. 2664, May 2019, doi: 10.3390/su11092664.

[11] Y. Li, “Introducing Deep Reinforcement Learning,” Medium, Jan. 13, 2019. https://medium.com/@yuxili/deeprl-6c8c48b6489b

[12] B. Flyvbjerg, “What you Should Know about Megaprojects and Why: An Overview,” Project Management Journal, vol. 45, no. 2, pp. 6–19, Apr. 2014, doi: 10.1002/pmj.21409.

[13] R. Barták, “On-line guide to constraint programming,” kti.mff.cuni.cz, 1998. http://kti.mff.cuni.cz/~bartak/constraints/.

[14] A. Shashkevich, “Ancient myths reveal early fantasies about artificial life,” Stanford News, Feb. 28, 2019. https://news.stanford.edu/2019/02/28/ancient-myths-reveal-early-fantasies-artificial-life/

[15] R. Anyoha, “The History of Artificial Intelligence,” Science in the News, Aug. 28, 2017. https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/

[16] A. M. Turing, “Computing Machinery And Intelligence,” Mind, vol. LIX, no. 236, pp. 433–460, 1950, doi: 10.1093/mind/lix.236.433.

[17] M. Haenlein and A. Kaplan, “A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence,” California Management Review, vol. 61, no. 4, pp. 5–14, Jul. 2019, doi: 10.1177/0008125619864925.

[18] M. M. Mijwel, “History of Artificial Intelligence,” Apr. 2015, [Online]. Available:https://www.researchgate.net/publication/322234922_History_of_Artificial_Intelligence

[19] O. Etzioni, “AI zooms in on highly influential citations,” Nature, vol. 547, no. 7661, pp. 32–32, Jul. 2017, doi: 10.1038/547032a.

[20] S. van Duin and N. Bakhshi, “Part 1: Artificial Intelligence Defined | Deloitte | Technology services,” Deloitte Sweden, Mar. 2017. https://www2.deloitte.com/se/sv/pages/technology/articles/part1-artificial-intelligence-defined.html

[21] S. Russell and P. Norvig, Artificial Intelligence : a Modern Approach, 2nd ed. Upper Saddle River, New Jersey: Pearson Education Limited, 2013.

[22] Sahotra Sarkar, J. Pfeifer, and J. Garson, *The philosophy of science : an encyclopedia / Vol. 1, A-M.* New York, Ny: Routledge/Taylor & Francis, Cop, 2006, p. 30.

[23] R. Fjelland, “Why general artificial intelligence will not be realized,” Humanities and Social Sciences Communications, vol. 7, no. 1, Jun. 2020, doi: 10.1057/s41599-020-0494-4.

[24] C. Robert, “Superintelligence: Paths, Dangers, Strategies,” *CHANCE*, vol. 30, no. 1, pp. 42–43, Jan. 2017, doi: 10.1080/09332480.2017.1302723.

[25] G. Press, “A Very Short History Of Artificial Intelligence (AI),” Forbes, Dec. 30, 2016. https://www.forbes.com/sites/gilpress/2016/12/30/a-very-short-history-of-artificial-intelligence-ai

[26] IBM Cloud Education, “What is Machine Learning?,” www.ibm.com, Jul. 15, 2020. https://www.ibm.com/cloud/learn/machine-learning

[27] Y. E. Bulut and Z. Voulgaris, AI for data science : artificial intelligence frameworks and functionality for deep learning, optimization, and beyond. Basking Ridge, New Jersey: Technics Publications, 2018.

[28] M. Attaran and P. Deb, “Machine learning: the new ‘big thing’ for competitive advantage,” *International Journal of Knowledge Engineering and Data Mining*, vol. 5, no. 4, p. 277, 2018, doi: 10.1504/ijkedm.2018.095523.

[29] “Constraints,” Springer. https://www.springer.com/journal/10601

[30] B. Farhang Moghaddam, “Mapping optimization techniques in project management,” *Journal of Project Management*, pp. 217–228, 2019, doi: 10.5267/j.jpm.2019.3.003.

[31] J. Zhang, G. Ding, Y. Zou, S. Qin, and J. Fu, “Review of job shop scheduling research and its new perspectives under Industry 4.0,” *Journal of Intelligent Manufacturing*, vol. 30, no. 4, pp. 1809–1830, Aug. 2017, doi: 10.1007/s10845-017-1350-2.

[32] R. Barták, “Constraint Programming – What is behind?,” Jan. 2000, [Online]. Available: https://www.researchgate.net/publication/2611569_Constraint_Programming_-_What_is_behind

[33] G. J. Lieberman and F. S. Hillier, Introduction to operations research, 9th ed. Boston: New York Mcgraw-Hill, 2010.

[34] L. Kandiller, *Principles of mathematics in operations research*. New York; London: Springer, 2011.

[35] J. Rajgopal, “Principles And Applications Of Operations Research,” *Maynard’s Industrial Engineering Handbook*, 5th Edition, pp. 11.27-11.44, 2004.

[36] J.C. Regin" Global Constraints and Filtering Algorithms," *Constraints and Integer Programming*. Boston: Springer, 2004.

[37] A. Bockmayr and J. Hooker, Constraint programming, *Handbook of Discrete Optimization*, Elsevier, pp. 559-600, 2005.

[38] J J.-F. Puget and P. Shaw, "Constraint Programming Background and History," presented at the Industry Solutions Optimization, Jan. 16, 2014. [Online]. Available: https://www.lnmb.nl/conferences/2014/programlnmbconference/Shaw-1.pdf

[39] J. N. Hooker and W.-J. . van Hoeve, "Constraint programming and operations research," Constraints, vol. 23, no. 2, pp. 172–195, Dec. 2017, doi: 10.1007/s10601-017-9280-3.

[40] A. Metzger *et al.*, "Comparing and Combining Predictive Business Process Monitoring Techniques," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 2, pp. 276-290, Feb. 2015, doi: 10.1109/TSMC.2014.2347265.

[41] H. Tahri, "Mathematical Optimization Methods: Application in Project Portfolio Management," Procedia - Social and Behavioral Sciences, vol. 210, pp. 339–347, Dec. 2015, doi: 10.1016/j.sbspro.2015.11.374.

[42] M. M. Meerschaert, *Mathematical modeling*. Amsterdam ; Boston: Academic Press/Elsevier, 2013.

[43] S. P. Bradley, A. C. Hax, and T. L. Magnanti, Applied mathematical programming. Reading, Mass. Addison-Wesley, 1992

[44] C. Scott-Young and D. Samson, "Project success and project team management: Evidence from capital projects in the process industries," *Journal of Operations Management*, vol. 26, no. 6, pp. 749–766, Nov. 2007, doi: 10.1016/j.jom.2007.10.006.

[45] J. Oxenstierna, "Warehouse Vehicle Routing using Deep Reinforcement Learning," 2019. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-396853

[46] E. Wari and W. Zhu, "A Constraint Programming model for food processing industry: a case for an ice cream processing facility," *International Journal of Production Research*, vol. 57, no. 21, pp. 6648–6664, Feb. 2019, doi: 10.1080/00207543.2019.1571250.

[47] W. Herroelen, J. Pajares, and A. López-Paredes, "Foreword," Annals of Operations Research, vol. 186, no. 1, pp. 393–394, Mar. 2011, doi: 10.1007/s10479-011-0859-6.

[48] M. Leila, "The big picture of Operations Research," Medium, Aug. 25, 2018. https://towardsdatascience.com/the-big-picture-of-operations-research-8652d5153aad

[49] M. Palpant, C. Artigues, and P. Michelon, "LSSPER: Solving the Resource-Constrained Project Scheduling Problem with Large Neighbourhood Search," *Annals of Operations Research*, vol. 131, no. 1–4, pp. 237–257, Oct. 2004, doi: 10.1023/b:anor.0000039521.26237.62.

[50] T. P. Sturm, "Sudoku and A.I," presented at the Proceedings of the 2009 International Conference on Artificial Intelligence, Las Vegas Nevada, USA, Jan. 2009.

[51] S. Heipcke, "Comparing Constraint Programming and Mathematical Programming Approaches to Discrete Optimisation-The Change Problem," *The Journal of the Operational Research Society*, vol. 50, no. 6, p. 581, Jun. 1999, doi: 10.2307/3010615.

[52] Class Lecture, Topic: "Introduction to Operations Research – An Overview." [Online]. Available: https://egyankosh.ac.in/bitstream/123456789/10605/1/Unit-1.pdf

[53] P. Baptiste, "Combining Operations Research and Constraint Programming to Solve Real-Life Scheduling Problems," *www.ercim.eu*, Jan. 2001. https://www.ercim.eu/publication/Ercim_News/enw44/baptiste.html

[54] S. M. Pour, J. H. Drake, L. S. Ejlertsen, K. M. Rasmussen, and E. K. Burke, "A hybrid Constraint Programming/Mixed Integer Programming framework for the preventive signaling maintenance crew scheduling problem," European Journal of Operational Research, vol. 269, no. 1, pp. 341–352, Aug. 2018, doi: 10.1016/j.ejor.2017.08.033.

[55] "Success in Disruptive Times," Pulse of the Profession, PMI, 2018. [Online]. Available: https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2018.pdf1

[56] J. Blazewicz, J. K. Lenstra, and A. H. G. Rinnooy. Kan, "Scheduling subject to resource constraints: classification and complexity," *Discrete Applied Mathematics*, vol. 5, no. 1, pp. 11–24, Jan. 1983, doi: 10.1016/0166-218x(83)90012-4.

[57] A. Ligęza *et al.*, “Explainable Artificial Intelligence. Model Discovery with Constraint Programming,” *Intelligent Systems in Industrial Applications*, pp. 171–191, 2021, doi: 10.1007/978-3-030-67148-8_13.

[58] R. M. e S. de Oliveira and M. S. F. O. de C. Ribeiro, “Comparing Mixed & Integer Programming vs. Constraint Programming by solving Job-Shop Scheduling Problems,” *Independent Journal of Management & Production*, vol. 6, no. 1, Mar. 2015, doi: 10.14807/ijmp. v6i1.262.

[59] N. Trautmann and P. Baumann, "Resource-constrained scheduling of a real project from the construction industry: A comparison of software packages for project management," 2009 IEEE International Conference on Industrial Engineering and Engineering Management, 2009, pp. 628-632, doi: 10.1109/IEEM.2009.53732

[60] V. Jain and I. E. Grossmann, “Algorithms for Hybrid MILP/CP Models for a Class of Optimization Problems,” *INFORMS Journal on Computing*, vol. 13, no. 4, pp. 258–276, Nov. 2001, doi: 10.1287/ijoc.13.4.258.9733.

[61] T. H. Yunes, A. V. Moura, and C. C. de Souza, “Hybrid Column Generation Approaches for Urban Transit Crew Management Problems,” *Transportation Science*, vol. 39, no. 2, pp. 273–288, May 2005, doi: 10.1287/trsc.1030.0078.

[62] J. C. Beck and P. Refalo, “A hybrid approach to scheduling with earliness and tardiness costs. Annals of Operations Research,” *Annals of Operations Research*, vol. 118, no. 1/4, pp. 49–71, 2003, doi: 10.1023/a:1021849405707.

[63] Y. Wu and H. Kim, “Digital Imaging in Assessment of Construction Project Progress,” *Proceedings of the 21st International Symposium on Automation and Robotics in Construction*, Sep. 2004, doi: 10.22260/isarc2004/0093.

[64] Project Management Institute, “Success Rates Rise 2017 9th Global Project Management Survey,” 2017. [Online]. Available: https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf

[65] M. Probst and M. Lahmann, “Artificial Intelligence and Project Management: Beyond Human Imagination!” presented at PMDay, Nov.23, 2018. [Online]. Available: https://pmi.bg/pmday2018/wp-content/uploads/2018/12/PMDAY2018_Manuel_Probst_Marc_Lahmann.pdf

[66] A. Butt, “Project Management through the lens of Artificial Intelligence,” Chalmers University Of Technology, 2018. [Online]. Available: https://odr.chalmers.se/bitstream/20.500.12380/256311/1/256311.pdf

[67] Ho Geun Lee, R. M. Lee, and G. Yu, “Constraint logic programming for qualitative and quantitative constraint satisfaction problems,” *Decision Support Systems*, vol. 16, no. 1, pp. 67–83, Jan. 1996, doi: 10.1016/0167-9236(94)00057-3.

[68] G. Phillips-Wren, “AI Tools In Decision Making Support Systems: A Review,” *International Journal on Artificial Intelligence Tools*, vol. 21, no. 02, p. 1240005, Apr. 2012, doi: 10.1142/s0218213012400052.

[69] A. M. Ham and E. Cakici, “Flexible job shop scheduling problem with parallel batch processing machines: MIP and CP approaches,” *Computers & Industrial Engineering*, vol. 102, pp. 160–165, Dec. 2016, doi: 10.1016/j.cie.2016.11.001.

[70] Y. Kwak, “Brief History of Project Management,” in *The Story of Managing Projects*, Quorum Book, 2003. doi: 10.1093/oxfordhb/9780199563142.003.0002.

[71] PricewaterhouseCoopers, “A Virtual Partnership? How Artificial Intelligence will disrupt Project Management and change the role of Project Managers - PwC Middle East,” *PwC*, 2019. https://www.pwc.com/m1/en/publications/virtual-partnership-artificial-ntelligence-disrupt-project-management-change-role-project-managers.html

[72] H. J. Wilson and P. R. Daugherty, “How Humans and AI Are Working Together in 1,500 Companies,” *Harvard Business Review*, Apr. 04, 2019. https://hbr.org/2018/07/collaborative-intelligence-humans-and-ai-are-joining-forces

[73] S. Changali, A. Mohammad, and M. van Nieuwland, “The construction productivity imperative | McKinsey,” *www.mckinsey.com*, Jul. 01, 2015. https://www.mckinsey.com/business-functions/operations/our-insights/the-construction-productivity-imperative

[74] T. V. Fridgeirsson, H. T. Ingason, H. I. Jonasson, and H. Jonsdottir, “An Authoritative Study on the Near Future Effect of Artificial Intelligence on Project Management Knowledge Areas,” *Sustainability*, vol. 13, no. 4, p. 2345, Feb. 2021, doi: 10.3390/su13042345.

[75] M. Relich and A. Świć, “Parametric Estimation and Constraint Programming-Based Planning and Simulation of Production Cost of a New Product,” *Applied Sciences*, vol. 10, no. 18, p. 6330, Sep. 2020, doi: 10.3390/app10186330.

[76] A. Champion, "When you should use Constraint Solvers instead of Machine Learning," *Towards Data Science*, Jan. 24, 2022. https://towardsdatascience.com/where-you-should-drop-deep-learning-in-favor-of-constraint-solvers-eaab9f11ef45

[77] J. Liu and M. Lu, "Constraint Programming Approach to Optimizing Project Schedules under Material Logistics and Crew Availability Constraints," *Journal of Construction Engineering and Management*, vol. 144, no. 7, p. 04018049, Jul. 2018, doi: 10.1061/(asce)co.1943-7862.0001507.

[78] A. Kwan and E. Tsang, "Mapping Constraint Satisfaction Problems to Algorithms and Heuristics." Department of Computer Science, University of Essex, 1996. [Online]. Available: https://www.researchgate.net/publication/2441532_Mapping_Constraint_Satisfaction_Problems_to_Algorithms_and_Heuristics

[79] H. Simonis, "Building Industrial Applications with Constraint Programming," *Constraints in Computational Logics*, pp. 271–309, 2001, doi: 10.1007/3-540-45406-3_6.

[80] L. D. Raedt, S. Nijssen, B. O'Sullivan, and P. V. Hentenryck, "Constraint Programming meets Machine Learning and Data Mining (Dagstuhl Seminar 11201)," *Dagstuhl Reports*, vol. 1, no. 5, pp. 61–83, 2011, doi: 10.4230/DagRep.1.5.61.

[81] "LINGO and optimization modeling," Lindo.com, 2019. https://www.lindo.com/index.php/products/lingo-and-optimization-modeling

[82] "MiniZinc," www.minizinc.org. https://www.minizinc.org/index.html

[83] R. Barták, M. A. Salido, and F. Rossi, "New trends in constraint satisfaction, planning, and scheduling: a survey," *The Knowledge Engineering Review*, vol. 25, no. 3, pp. 249–279, 2010.

[84] "Solve your Toughest Planning and Scheduling Problems: How Business Managers can use Mathematical Optimization Technology," IBM Software ILOG Optimization, 2004. [Online]. Available: https://www.ibm.com/downloads/cas/E0NVPXOJ

[85] P. Laborie, J. Rogerie, P. Shaw, and P. Vilím, "IBM ILOG CP optimizer for scheduling," *Constraints*, vol. 23, no. 2, pp. 210–250, Mar. 2018, doi: 10.1007/s10601-018-9281-x.

[86] H. Meyer auf'm Hofe, "Solving Rostering Tasks as Constraint Optimization," *Lecture Notes in Computer Science*, pp. 191–212, 2001, doi: 10.1007/3-540-44629-x_12.

[87] H. Meyer auf'm Hofe, "ConPlan/SIEDAplan: Personnel assignment as a problem of hierarchical constraint satisfaction," in *Proceedings of the Third International Conference on the Practical Application of Constraint Technology*, London, pp. 257–271, 1997.

[88] "The Future of Jobs Employment, Skills and Workforce Strategy for the Fourth Industrial Revolution," World Economic Forum, Jan. 2016. [Online]. Available: https://www3.weforum.org/docs/WEF_Future_of_Jobs.pdf

[89] G. Pesant, "A constraint programming primer," *EURO Journal on Computational Optimization*, vol. 2, no. 3, pp. 89–97, Jul. 2014, doi: 10.1007/s13675-014-0026-3.

[90] R. Barták, "History of Constraint Programming," *Wiley Encyclopedia of Operations Research and Management Science*, Jan. 2011, doi: 10.1002/9780470400531.eorms0382.

[91] A. Marando, "Balancing Project Management Hard Skills and Soft Skills," *Rabb School of Continuing Studies Division of Graduate Professional Studies*, Feb. 2012.

[92] M. J. Mortenson, N. F. Doherty, and S. Robinson, "Operational research from Taylorism to Terabytes: A research agenda for the analytics age," *European Journal of Operational Research*, vol. 241, no. 3, pp. 583–595, Mar. 2015, doi: 10.1016/j.ejor.2014.08.029.

[93] R. Burke and S. Barron, *Project Management Leadership Building Creative Teams*. Hoboken, Nj, Usa John Wiley & Sons, Inc, 2012.

[94] R. Speering *et al.*, "The art of project leadership: Delivering the world's largest projects," McKinsey Capital Projects & Infrastructure Practice, Sep. 2017. [Online]. Available: https://www.mckinsey.com/business-functions/operations/our-insights/the-art-of-project-leadership-delivering-the-worlds-largest-projects

[95] M. pori, "Application of Multi-Criteria Methods in Natural Resource Management - A Focus on Forestry," *Sustainable Forest Management - Current Research*, May 2012, doi: 10.5772/23667.

[96] L. D. Raedt, "Unifying AI Paradigms & Representations," *TAILOR Foundations of Trustworthy AI integrating Learning, Optimisation and Reasoning*. https://tailor-network.eu/research-overview/unifying-paradigms/

[97] Nagraj Balakrishnan, B. Render, and R. M. Stair, *Managerial decision modeling with spreadsheets.* New York: Pearson, 2016.

[98] PwC, "Insights and Trends: Current Programme and Project Management Practices The second global survey on the current state of project management maturity in organisations across the world.," 2004. [Online]. Available: https://www.pwc.com/cl/es/publicaciones/assets/insighttrends.pdf

[99] M. J. Liberatore and B. Pollack-Johnson, "Factors influencing the usage and selection of project management software," *IEEE Transactions on Engineering Management*, vol. 50, no. 2, pp. 164–174, May 2003, doi: 10.1109/tem.2003.810821.

[100] D. White and J. Fortune, "Current practice in project management — an empirical study," *International Journal of Project Management*, vol. 20, no. 1, pp. 1–11, Jan. 2002, doi: 10.1016/s0263-7863(00)00029-6.

[101] P. Patanakul, B. Iewwongcharoen, and D. Milosevic, "An Empirical Study on the use of Project Management Tools and Techniques across Project Life-Cycle and their Impact on Project Success," *Journal of General Management*, vol. 35, no. 3, pp. 41–66, Mar. 2010, doi: 10.1177/030630701003500304.

[102] L. H. Cherri, M. A. Carravilla, C. Ribeiro, and F. M. B. Toledo, "Optimality in nesting problems: New constraint programming models and a new global constraint for non-overlap," *Operations Research Perspectives*, vol. 6, p. 100125, 2019, doi: 10.1016/j.orp.2019.100125.

[103] O. Icmeli, S. Selcuk Erenguc, and C. J. Zappe, "Project Scheduling Problems: A Survey," *International Journal of Operations & Production Management*, vol. 13, no. 11, pp. 80–91, Nov. 1993, doi: 10.1108/01443579310046454.

[104] "IBM ILOG CPLEX Optimization Studio Version 20.1.0." https://www.ibm.com/docs/en/icos/20.1.0

[105] "Problem characteristics affecting CPLEX run time," *www.ibm.com*, Nov. 14, 2012. https://www.ibm.com/support/pages/problem-characteristics-affecting-cplex-run-time

[106] "Guidelines for estimating CPLEX memory requirements based on problem size," *www.ibm.com*, Apr. 09, 2012. https://www.ibm.com/support/pages/guidelines-estimating-cplex-memory-requirements-based-problem-size

A Annex A – Literature Review

A.1. The table of PM methods, methodologies, tools and techniques with their frequency of use [100]

Project management methods, methodologies, tools and techniques — frequency of use

Project management method/methodology/tool/technique	Count of frequency of use	Total used	Mean	Mode	Range
Project management methods/methodologies		206	0.87	1	3
Projects in controlled environments (PRINCE)	23				
Projects in controlled environments 2 (PRINCE2)	14				
Structured systems analysis and design methodology (SSADM)	17				
The European risk management methodology (RISKMAN)	1				
The RIBA plan of work	2				
Other project management methods/methodologies[a]	16				
In house project management methods	128				
In house similar to PRINCE	5				
Project management tools		617	2.61	1	7
Critical path method (CPM)	70				
Work breakdown structure (WBS)	75				
Cash flow analysis (CFA)	43				
Gantt bar charts	152				
Graphical evaluation and review technique (GERT)	4				
Programme evaluation and review technique (PERT)	24				
Strengths weaknesses, opportunities and threats (SWOT)	41				
Other project management tools[b]	21				
Project management software	182				
In house project management tools	5				
Decision making techniques		172	0.73	0	4
Cost benefit analysis (CBA)	88				
Decision analysis (DA)	9				
Sensitivity analysis (SA)	19				
Expressed preferences	23				
Implied preferences	11				
Revealed preferences	11				
Other decision making techniques	9				
In house decision making techniques	2				
Risk assessment tools		147	0.62	0	10
Life-cycle cost analysis (LCCA)	25				
Event tree analysis (ETA)	8				
Fault tree analysis (FTA)	6				
Probability analysis (PA)	34				
Reliability analysis	13				
Uncertainty analysis	3				
Failure mode and effect analysis (FMEA)	10				
Hazard analysis (HAZAN)	9				
Hazard and operability studies (HAZOP)	9				
Operation and maintenance risk analysis (OMRA)	4				
Preliminary hazard analysis (PHA)	5				
Other risk assessment tools	7				
In house risk assessment tools	14				
Computer models/databases/indexes		40	0.17	0	3
CRUNCH	1				
Lessons learnt files (LLF)	23				
Expert systems	4				
In house computer models/databases/indexes	12				
Computer simulations		11	0.05	0	2
Hertz	1				
Monte Carlo	10				
Other techniques		11	0.05	0	2
Other techniques	17				
All methods, tools and techniques		1210	5.13	3	23

[a] Includes other methods used in Information Systems Development Projects.
[b] Includes tools used in Information Systems Development Projects.

A.2. The CSP attributes and their relations with algorithms and heuristics [78]

Problem specification and characteristics		Algorithms and heuristics											
		Complete algorithms				Heuristics				Stochastic search			Other
		BT IB B&B	LA	INB	SS	FFP	MWO	MBO	MC	HC TB SA	GA	CM	
Solution(s) required	any solution required	most of these algorithms can cope with these problems, with some exceptions (see text)			✗	✔	✔	✔	✔	✔			
	all solutions required				✔	✔	✔	✔	✗	✗			
	optimal solution required	most of these algorithms (except B&B) have to find all solutions, then compare them according to the objective function				variable ordering heuristics can help algorithms to find all solutions; value ordering is less useful (see text)				✔ useful when near-optimal solutions are acceptable			
Time available	not critical	✔ all the above algorithms are applicable				✔	✔	✔	✔	there is less incentive to use them			
	limited time	limited by the combinatorial explosion probem				✔ good heuristics may speed up complete algorithms				✔	✔	✔	
	real time / near real time	✗ most of these algorithms cannot solve useful problems of this category				heuristics may help, but unlikely to speed up complete algorithms dramatically				✔	✗	✔	
Problem Tightness	very loosely constrained	✔	✗		✗	application not solely determined by the tightness of the problem				different algorithms may suit different types of problems			
	very tightly constrained	✗	✔		✔								
Characteristics of the CSP's primal graph (PG)	the PG is unconnected									these characteristics may or may not be exploited by the above stochastic methods			D&C
	the PG is a tree						✔						TS (PP)
	degrees of nodes vary significantly			✔			✔						
	PG has small bandwidth			✗				✔					
Other problem characteristics	some variables and values are more constrained than others		✔ likely to be useful			✔			✔				

Keys: ✔ means the algorithm/heuristic should be considered; ✗ means it should not (e.g. inapplicable or ineffective); blank means the problem specification or characteristic is not a deciding factor; PP = Pre-processing

B Annex B – Models for case

B.1. Part – 2

B.1.1. CP Model

```
using CP;
 int N = ...;   // # of tasks
 float T = ...; // project completion

 tuple Task {
 key int id;        // task id
 int dur1;          // activity durations
 int  dur2;         // minimum possible durations
 {int} successor;   // precedence relationship
 float u;           // unit reduction costs
}

{Task} Tasks = ...;

dvar int+ start[Tasks]; //variables
dvar int+ end[Tasks];
dvar int+ cut[Tasks];

minimize sum(t in Tasks) t.u*cut[t]; // objective function of cost of crashing

 subject to {                       //constraints
   forall(t in Tasks)
     start[t] + t.dur2 - cut[t] <= end[t];
// sum of the start time and duration <= completion of the activity

   forall(t_before in Tasks,t_after in t_before.successor)
     end[t_before] <= start[<t_after>] ;
// end of the predecessor activity is earlier than the successor's start

   forall(t in Tasks)
     cut[t] <= t.dur2 - t.dur1;
// crashing of the activities <= maximum allowable crashing

   forall(t in Tasks)
     end[t] <= T;       // project ends after the completion of each activity
 }
```

B.1.2. LP Model

```
int N = ...;   // # of tasks
float T = ...; // project completion

tuple Task {
 key int id;         // task id
 int dur1;           // activity durations
 int  dur2;          // minimum possible durations
 {int} successor;    // precedence relationship
 float u;            // unit reduction costs
}

{Task} Tasks = ...;

 dvar float+ start[Tasks];   //variables
 dvar float+ end[Tasks];
 dvar float+ cut[Tasks];

 minimize sum(t in Tasks) t.u*cut[t]; // objective function of cost of crashing

 subject to {                 //constraints
   forall(t in Tasks)
     start[t] + t.dur2 - cut[t] <= end[t];
// sum of the start time and duration <= completion of the activity

   forall(t_before in Tasks,t_after in t_before.successor)
     end[t_before] <= start[<t_after>] ;
// end of the predecessor activity is earlier than the successor's start

   forall(t in Tasks)
     cut[t] <= t.dur2 - t.dur1;
// crashing of the activities <= maximum allowable crashing

   forall(t in Tasks)
     end[t] <= T;      // project ends after the completion of each activity
 }
```

B.2. Part – 3

B.2.1. MIP Model

```
int N = ...;
int T = ...;
int Capacity = ...;
int Capacity2 = ...;

tuple Task {
  key int id;                      // task id
  int     dur1;                    // min. allowable durations with max crash
  int     dur2;                    // activity duration without crash
  int     worker;            // worker resource requirements of the activities
```

```
  int     machine;           // machine resource requirements of the activities
  {int}   successors;        // successor relationships
  float   u;                 // unit reduction costs
}

{Task} Tasks = ...;

 dvar int+ start[Tasks];
 dvar int+ end[Tasks];
 dvar int+ cut[Tasks];
 dvar int+ indicate[1..3,Tasks,0..T] in 0..1;

 minimize sum(t in Tasks) t.u*cut[t];
//objective function cost of project crashing

 subject to {               // constraints
   forall(t in Tasks)
     start[t] + t.dur2 - cut[t] <= end[t];
// an activity can not ends before its starting time plus duration

   forall(t in Tasks)
     cut[t] <= t.dur2 - t.dur1;
//the allowable cut <= the diff. between (no crash - max crash) duration

   forall(t in Tasks)
     end[t] <= T;        // project ends after the completion of each activity

   forall(t1 in Tasks, t2id in t1.successors)
     end[t1] <= start[<t2id>];      // precedence/successor constraints

   forall(i in 0..T-1)
//to determine the active tasks for all the time intervals
     forall(t in Tasks) {
       i - start[t] + 0.001 <= T*indicate[1,t,i];      // In text explanation!
       end[t] - i - 0.001 <= T*indicate[2,t,i];
       indicate[1,t,i] + indicate[2,t,i] - 1 <= indicate[3,t,i];
     };

   forall(i in 0..T-1)
     sum(t in Tasks) t.worker*indicate[3,t,i] <= Capacity;
// sum of each active task's resource1 usage at a time instance <= capacity

   forall(i in 0..T-1)
     sum(t in Tasks) t.machine*indicate[3,t,i] <= Capacity2;
// sum of each active task's resource2 usage at a time instance <= capacity2
}
```

B.2.2. CP Model

```
using CP;
int N = ...;
int T = ...;
int Capacity = ...;
int Capacity2 = ...;

tuple Task {
  key int id;         // task id, minimum possible durations, activity duration
  int     dur1;       // min. allowable durations with max crash
  int     dur2;       // activity duration without crash
  int     worker;     // worker resource requirements of the activities
  int     machine;    // machine resource requirements of the activities
  {int}   successors; // successor relationships
  float   u;          // unit reduction costs
}

{Task} Tasks = ...;

dvar interval v[t in Tasks] size t.dur1..t.dur2;
//decision variable bounded between max & min duration interval

cumulFunction rsrcUsage =
  sum (t in Tasks: t.worker>0) pulse(v[t], t.worker);
// cumulative machine resource usage

cumulFunction rsrcUsage2 =
  sum (t in Tasks: t.machine>0) pulse(v[t], t.machine);
// cumulative machine resource usage

minimize
  sum(t in Tasks) t.u*(t.dur2-sizeOf(v[t]));
// objective function of the cost of project crashing

subject to {                         // constraints
  rsrcUsage <= Capacity;             // resource capacity constraints for workers
  rsrcUsage2 <= Capacity2;           // resource capacity constraints for machines

  forall (t1 in Tasks, t2id in t1.successors)
    endBeforeStart(v[t1], v[<t2id>]);      // precedence/successor constraints

  max(t in Tasks) endOf(v[t]) <= T;
// project ends if and only if all activities done
}
```

C Annex C – Results for the case

C.1. Input Data

C.1.1. Input data for the project with 25 activities

id	dur1	dur2	worker	machine	successors	u
1	0	0	0	0	{2 4}	0
2	32	56	4	3	{3}	26.66
3	48	88	8	2	{6}	20
4	40	72	3	4	{5}	18.33
5	24	40	11	5	{6}	9.9
6	88	140	9	1	{7}	32
7	8	14	2	2	{8 12}	3
8	18	20	3	6	{9}	10
9	24	50	6	4	{11 16}	3.5
10	15	40	7	5	{12 14}	21
11	32	44	2	3	{12}	6.9
12	6	12	2	1	{15}	7
13	10	24	5	3	{14}	14
14	23	40	2	2	{15}	15
15	31	38	5	3	{18}	25
16	14	22	7	4	{17 20}	10.33
17	62	80	9	6	{18}	19.9
18	12	42	7	3	{19}	19
19	42	50	2	5	{22}	8.75
20	22	38	6	3	{21}	20
21	46	55	11	4	{22 23}	11
22	31	35	5	4	{24}	7
23	22	30	9	0	{25}	8
24	12	20	3	5	{}	28
25	20	30	2	0	{26}	17
26	10	16	0	0	{27}	5.5
27	0	0	0	0	{}	0

C.1.2. Input data for the project with 50 activities

id	dur1	dur2	successor	u
1	0	0	{2 3 4}	26.66
2	4	8	{10 18}	20
3	3	5	{19 24}	20
4	2	3	{5 10 11}	6.75
5	5	10	{6 13}	4.66
6	5	9	{7 8 10}	5
7	8	10	{9 13 17}	4.4
8	10	10	{9 12 16}	0
9	3	5	{14 18 29}	4.33
10	1	2	{22 23}	5.66
11	3	10	{18 28}	30
12	2	2	{14 18 20}	0
13	2	4	{21}	10
14	1	6	{15 21}	7.5
15	5	7	{29 37}	5.99
16	1	1	{17 21 29}	0
17	3	10	{20 24}	4.99
18	7	9	{21 23 25}	17.33
19	4	10	{25 28}	8.5
20	4	5	{29 37}	12
21	1	1	{26 28 30}	0
22	1	1	{32 35}	0
23	5	9	{27 33}	12
24	9	10	{25 27 33}	4.99
25	1	4	{31}	18
26	8	10	{36 41}	9
27	2	6	{35}	11.75
28	1	1	{35 46}	0
29	4	8	{38}	2.5
30	3	5	{31 33}	19.25
31	1	3	{34 47}	4.8
32	2	6	{37 47}	3.33
33	2	2	{48}	0
34	1	7	{47}	11.33
35	2	9	{47}	8.5
36	1	5	{42}	3.99
37	6	10	{38 41}	25.5
38	3	3	{49}	0
39	2	2	{41 42}	0
40	5	9	{42}	4.9
41	1	6	{47}	5.9
42	3	3	{49}	0
43	3	3	{47}	0
44	5	7	{48 50}	2.5
45	1	1	{49}	0
46	1	1	{48}	0
47	4	7	{49}	9.33
48	5	5	{50}	0
49	2	8	{50}	10.99
50	0	0	{}	0

C.2. Results for Part – 2 No Resource Constraint

C.2.1. For 25 activities project

T_actual = 559		LP					CP							
T_target	Cost (€)	V	C	T (s)	T (min.s.cs)	M (MB)	V	C	T (s)	Time to optimal solution	CPU (min.s.cs)	M (MB)	peak memory (MB)	memory usage (MB)
450	1549,74	81	112	0,04	00.00.04	1,523	81	116	3,74	3,74	15 min+	10,1	73,836	10,1
420	2479,04	81	112	0,06	00.00.06	1,336	81	116	137,49	137,79	15 min+	15,5	77,539	15,5
390	3439,04	81	112	0,07	00.00.07	1,52	81	116	3,8	3,8	15 min+	9	75,066	9
360	4613,07	81	112	0,07	00.00.07	1,207	81	116	19,36	19,36	15 min+	10,1	67,32	10,1
350	5062,98	81	112	0,04	00.00.04	1,457	81	116	0,87	0,87	15 min+	7,5	81,48	7,5
340	5553,86	81	112	0,12	00.00.12	1,344	81	116	16,11	1,49	00.16.11	8,1	19,156	8,1
330	infeasible	81	112	0,15	00.00.15	1,73	81	116	7,3	NA	00.07.30	3,02	3,02	NA
			average in s:	0,1	average in MB:	1,45		average in s:	27,0		average in MB:	9,05		

C.2.2. For 50 activities project

T_actual = 85		LP					CP				
T_target	Cost (€)	V	C	T (s)	CPU (min.s.cs)	peak memory (MB)	V	C	T (s)	CPU (min.s.cs)	peak memory (MB)
70	85,87	150	241	0,07	00.00.07	1,805	150	245	3,74	00.05.32	10,1
65	139,16	150	241	0,04	00.00.04	1,605	150	245	17,69	00.17.69	20,867
60	201,23	150	241	0,06	00.00.06	1,844	150	245	14,86	00.14.86	21,184
55	304,68	150	241	0,06	00.00.06	1,523	150	245	20,88	00.20.88	20,523
50	465,26	150	241	0,06	00.00.06	1,648	150	245	2,17	00.02.17	20,238
45	infeasible	150	241	0,14	00.00.14	1,789	150	245	1,64	00.01.64	4,551
			average in s:	0,1	average in MB:	1,70		average in s:	10,2	average in MB:	16,24

C.3. Results for Part – 3 Resource Constraint Project Scheduling Problem

C.3.1. Time and Memory Results for 25 activities project

	Parameters			Objective	MIP				CP			
	Target time T	Worker C1	Machine C2	Cost	V	C	T (s)	M (MB)	V	C	T (s)	M (MB)
25 tasks without crash T_actual = 559 C1_actual=25 C1_available=17 C2_actual=15 C2_available=12	450	17	12	3067,78	36531	37462	49,09*	8,80*	35	27	0,2*	5,9*
	440	17	12	3387,78	35721	36632	65,25*	10,49*	35	27	0,37*	5,3*
	430	17	12	3707,78	34911	35802	38,94*	0,02	35	27	0,47*	5,2*
	420	17	12	4207,78	34101	34972	25,41	195,14	35	27	446,44	39,422
	410	17	12	4347,78	33291	34142	20,84	169,06	35	27	154,8	29,5
	400	17	12	4667,78	32481	33312	4,21	136,914	35	27	13,79	16,945
	390	17	12	5106,18	31671	32482	1,26	99,719	35	27	0,75	11,289
	380	17	12	infeasible	30861	31652	0,54	26,672	35	27	0,13	11,734
					MIP				CP			
	Target time T	Worker C1	Machine C2	Cost	V	C	T (s)	M (MB)	V	C	T (s)	M (MB)
	440	25	99	1849,04	35721	36632	1,74	132,867	35	27	0,22	11,137
	440	24	99	2088,64	35721	36632	10,22*	4,8*6	35	27	0,21*	4,1*
	440	23	99	2088,64	35721	36632	8,75*	3,75*	35	27	0,2*	3,8*
	440	22	99	2088,64	35721	36632	10,19*	5,03*	35	27	0,2*	3,9*
	440	21	99	2088,64	35721	36632	7,73*	4,21*	35	27	0,2*	3,9*
	440	20	99	2088,64	35721	36632	10,97*	3,28*	35	27	0,24*	5,8*
	440	19	99	2359,84	35721	36632	22,25*	7,21*	35	27	0,19*	3,9*
	440	18	99	3078,88	35721	36632	42,81*	6,56*	35	27	0,2*	5,5*
	440	17	99	3387,78	35721	36632	45,14*	4,47*	35	27	0,41*	4,7*
	440	16	99	3528,68	35721	36632	163,08*	29,13*	35	27	0,24*	5,4*
	440	15	99	3571,4	35721	36632	617,19*	30,41*	35	27	1,91*	4,7*
	440	14	99	3621,8	35721	36632	19,36*	0,02*	35	27	0,38*	5,3*
	440	13	99	4464,58	35721	36632	68,5*	19,21*	35	27	0,2*	3,3*
	440	12	99	5382,38	35721	36632	7,32	100,42	35	27	37,73	17,645
	440	11	99	5615,04	35721	36632	77,47	114,57	35	27	436	40,836
	440	10	99	infeasible	35721	36632	0,45	44,711	35	27	0,12	2,562
					MIP				CP			
	Target time T	Worker C1	Machine C2	Cost	V	C	T (s)	M (MB)	V	C	T (s)	M (MB)
	440	99	16	1849,04	35721	36632	1,72	124,238	35	27	0,21	11,75
	440	99	15	1849,04	35721	36632	1,69	126,449	35	27	0,2	11,613
	440	99	14	1849,04	35721	36632	1,55	125,078	35	27	0,32	13,262
	440	99	13	1849,04	35721	36632	00.01.74	128,398	35	27	0,31	12,855
	440	99	12	2088,64	35721	36632	14,19*	5,9*	35	27	0,2*	3,9*
	440	99	11	2546,64	35721	36632	484,24*	23,49*	35	27	0,68*	6,2*
	440	99	10	2546,64	35721	36632	467,8*	20,79*	35	27	0,17*	3,8*
	440	99	9	2666,64	35721	36632	663,23*	24,45*	35	27	0,95*	8,3*
	440	99	8	3594,2	35721	36632	24,92*	232,172*	35	27	388,44*	12,8*
	440	99	7	3600,26	35721	36632	23,41	202,715	35	27	938,09	75,914
	440	99	6	infeasible	35721	36632	1,27	32,164	35	27	1,51	11,965
					average in s, MB:		88,3	62,95	av. in s, MB:		69,3	12,32

C.3.2. Crash amount from each 25 activity

T_actual = 559																													
T_target	Worker C1	Machine C2	Cost (€)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	double check
450	17	12	3067,78	24	40	0	16	0	6	2	26	0	1	0	0	0	7	8	0	30	0	7	9	0	8	0	10	6	3067,78
440	17	12	3387,78	24	40	0	16	10	6	2	26	0	1	0	0	0	7	8	0	30	0	7	9	0	8	0	10	6	3387,78
430	17	12	3707,78	24	40	0	16	20	6	2	26	0	1	0	0	0	7	8	0	30	0	7	9	0	8	0	10	6	3707,78
420	17	12	4207,78	24	40	0	16	30	6	2	26	0	1	0	0	0	7	8	0	30	0	7	9	0	8	0	10	6	4027,78
410	17	12	4347,78	24	40	0	16	40	6	2	26	0	1	0	0	0	7	8	0	30	0	7	9	0	8	0	10	6	4347,78
400	17	12	4667,78	24	40	0	16	50	6	2	26	0	1	0	0	0	7	8	0	30	0	7	9	0	8	0	10	6	4667,78
390	17	12	5106,18	24	40	0	16	52	6	2	26	0	9	0	0	0	7	8	8	30	0	15	9	0	8	0	10	6	5106,18
390	17	12	no solution	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		-	-	-	-	-	no solution
Time T	Worker C1	Machine C2	Cost																										
440	25	99	1849,04	0	40	0	8	0	6	2	26	0	8	0	0	0	0	8	0	25	8	0	3	4	8	0	0	6	1849,04
440	24	99	2088,64	0	40	0	8	8	6	2	26	0	12	6	0	0	0	8	0	17	8	0	9	4	8	0	0	6	2088,64
440	23	99	2088,64	0	40	0	8	8	6	2	26	0	12	6	0	0	0	8	0	17	8	0	9	4	8	0	0	6	2088,64
440	22	99	2088,64	0	40	0	8	8	6	2	26	0	12	6	0	0	0	8	0	17	8	0	9	4	8	0	0	6	2088,64
440	21	99	2088,64	0	40	0	8	8	6	2	26	0	12	6	0	0	0	8	0	17	8	0	9	4	8	0	0	6	2088,64
440	20	99	2088,64	0	40	0	8	8	6	2	26	0	12	6	0	0	0	8	0	17	8	0	9	4	8	0	0	6	2088,64
440	19	99	2359,84	0	40	0	8	18	6	2	26	0	0	0	0	0	0	8	0	21	8	0	9	4	8	0	0	6	2359,84
440	18	99	3078,88	24	40	0	16	18	6	2	26	0	0	0	0	0	0	8	0	21	8	0	9	4	8	0	0	6	3078,88
440	17	99	3387,78	24	40	0	16	10	6	2	26	0	1	0	0	0	7	8	0	30	0	7	9	0	8	0	10	6	3387,78
440	16	99	3528,68	24	40	0	16	36	6	2	26	0	12	6	0	0	0	8	0	4	0	0	9	0	8	0	10	6	3528,68
440	15	99	3571,4	16	40	0	16	44	6	2	26	0	12	6	0	0	0	8	0	4	0	0	9	0	8	0	10	6	3571,4
440	14	99	3621,8	16	40	0	16	38	6	2	26	0	12	6	0	0	0	8	16	0	0	0	9	0	8	0	10	6	3621,8
440	13	99	4464,58	24	40	0	16	39	6	2	26	10	0	0	0	0	2	8	18	30	0	0	9	4	8	0	0	5	4464,58
440	12	99	5382,38	24	40	0	16	52	6	2	26	14	12	6	0	6	1	8	18	30	8	1	9	4	0	8	0	1	5382,38
440	11	99	5615,04	24	40	2	16	52	6	2	26	14	12	6	14	6	1	8	18	30	8	1	9	4	0	8	0	1	5615,04
440	10	99	no solution	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		-	-	-	-	-	no solution
Time T	Worker C1	Machine C2	Cost																										
440	99	16	1849,04	0	40	0	8	0	6	2	26	0	8	0	0	0	0	8	0	25	8	0	3	4	8	0	0	6	1849,04
440	99	15	1849,04	0	40	0	8	0	6	2	26	0	8	0	0	0	0	8	0	25	8	0	3	4	8	0	0	6	1849,04
440	99	14	1849,04	0	40	0	8	0	6	2	26	0	8	0	0	0	0	8	0	25	8	0	3	4	8	0	0	6	1849,04
440	99	13	1849,04	0	40	0	8	0	6	2	26	0	8	0	0	0	0	8	0	25	8	0	3	4	8	0	0	6	1849,04
440	99	12	2088,64	0	40	0	8	8	6	2	26	0	12	6	0	0	0	8	0	17	8	0	9	4	8	0	0	6	2088,64
440	99	11	2546,64	0	40	0	8	26	6	2	26	0	12	0	0	0	0	8	0	13	8	0	9	4	8	0	0	6	2546,64
440	99	10	2546,64	0	40	0	8	26	6	2	26	0	12	0	0	0	0	8	0	13	8	0	9	4	8	0	0	6	2546,64
440	99	9	2666,64	0	40	0	8	32	6	2	26	0	12	6	0	0	0	8	0	7	8	0	9	4	8	0	0	6	2666,64
440	99	8	3594,2	0	40	0	8	46	6	2	26	0	12	0	0	0	0	0	18	0	8	14	9	4	0	7	0	0	3594,2
440	99	7	3600,26	1	40	0	9	44	6	2	26	0	12	0	0	0	0	0	18	0	8	14	9	4	0	8	0	1	3600,26
440	99	6	no solution	-	-	-	-	-	-	-	-	-	-	-		-	-		-	-	-	-	-	-	-	-	-	-	no solution

C.3.3. Time and Memory Results for 50 activities project

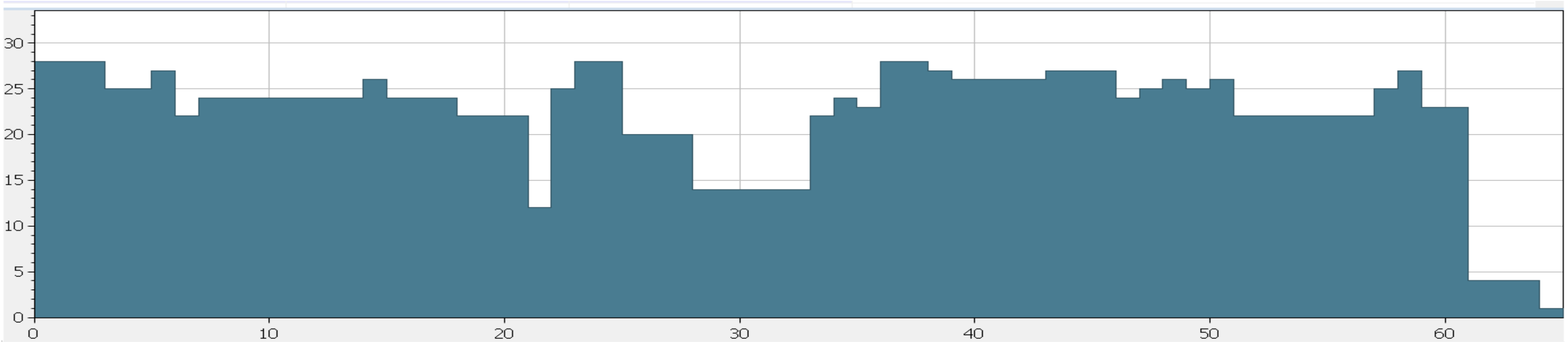
	Parameters			Objective	MIP				CP			
	Target time T	Worker C1	Machine C2	Cost (€)	V	C	T (s)	M (MB)	V	C	T (s)	M (MB)
50 tasks without crash T_actual = 85 C1_actual=41 C1_available=28 C2_actual=54 C2_available=46	70	28	46	85,87	10650	10881	10,43	151,367	95	50	0,7	17,887
	65	28	46	139,16	9900	10121	5,04	127,36	35	27	0,95	16,773
	60	28	46	204,86	9150	9361	7,39	96,76	35	27	24,15	22,73
	55	28	46	342,43	8400	8601	275,3	135,59	35	27	178,34	37,09
	50	28	46	543,54	7650	7841	18,07	88,97	35	27	27,1	23,285
	45	28	46	infeasible	6900	7081	0,12	11,11	35	27	0,29	2,602
					MIP				**CP**			
	Target time T	**Worker C1**	**Machine C2**	**Cost (€)**	V	C	T (s)	M (MB)	V	C	T (s)	M (MB)
	65	**48**	**99**	139,16	9900	10121	1,25	110,699	50	95	0,39	15,355
	65	**43**	99	139,16	9900	10121	1,37	112,543	50	95	0,4	16,672
	65	**38**	99	139,16	9900	10121	1,68	110,012	50	95	0,45	16,242
	65	**33**	99	139,16	9900	10121	3,1	115,742	50	95	0,57	15,293
	65	**28**	99	139,16	9900	10121	5,45	116,164	50	95	0,73	16,227
	65	**23**	99	no solution	9900	10121	NA	NA	50	95	NA	NA
	65	**18**	99	no solution	9900	10121	NA	NA	50	95	NA	NA
	65	**13**	99	infeasible	9900	10121	15,9	114,625	50	95	1,79	9,77
					MIP				**CP**			
	Target time T	**Worker C1**	**Machine C2**	**Cost (€)**	V	C	T (s)	M (MB)	V	C	T (s)	M (MB)
	65	99	**55**	139,16	9900	10121	1,25	114,078	50	95	0,28	14,234
	65	99	**50**	139,16	9900	10121	1,82	116,125	50	95	0,36	15,32
	65	99	**45**	139,16	9900	10121	1,79	115,922	50	95	0,39	15,652
	65	99	**40**	139,16	9900	10121	6,46	136,496	50	95	0,780	16,328
	65	99	**35**	156,62	9900	10121	254,6	136,094	50	95	77,23	25,434
	65	99	**30**	no solution	9900	10121	NA	NA	50	95	NA	NA
	65	99	**25**	no solution	9900	10121	NA	NA	50	95	NA	NA
	65	99	**20**	infeasible	9900	10121	4,42	95,496	50	95	0,75	11,539
						average in s:	34,2	111,40		**average in s:**	17,5	17,14

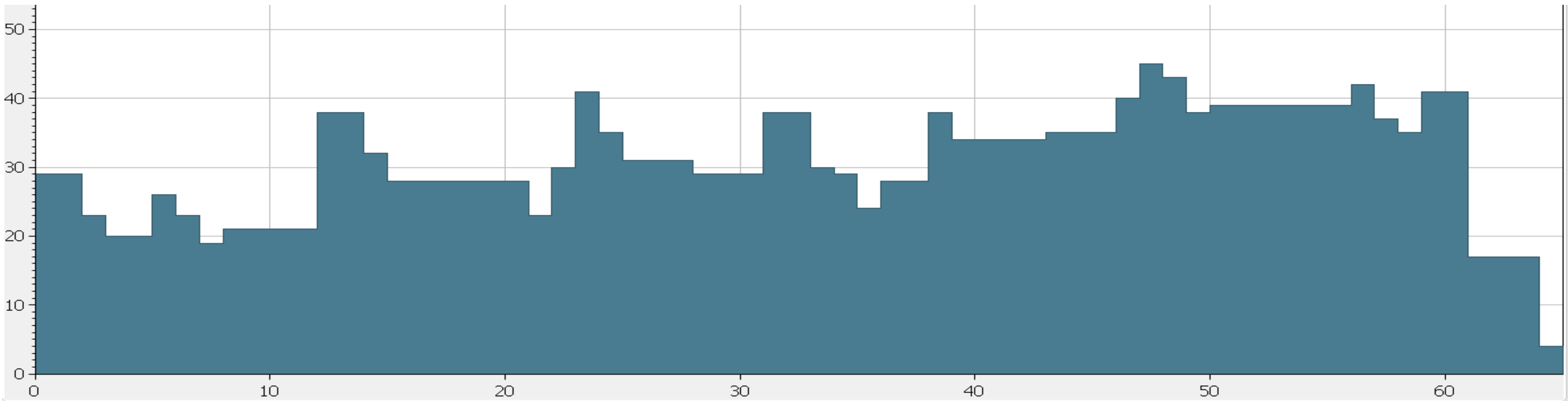
C.3.4. Crash amount from each 50 activity

Target T	C1	C2	Cost	
70	28	46	85,87	0 0 0 1 5 4 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0
65	28	46	139,16	0 0 0 1 5 4 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 3 0 4 0
60	28	46	204,86	0 0 0 1 5 4 0 0 2 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 3 0 6 0
55	28	46	342,43	0 0 0 1 5 4 0 0 2 0 0 0 0 0 1 0 7 0 0 0 0 0 0 1 0 2 4 0 0 0 1 0 0 0 0 5 2 0 0 0 0 4 0 0 0 0 3 0 6 0
50	28	46	543,54	0 0 0 1 5 4 0 0 2 0 0 0 0 4 2 0 7 2 0 0 0 0 1 1 0 2 4 0 2 0 2 0 0 6 7 4 0 0 0 2 5 0 0 0 0 0 3 0 6 0
45	28	46	infeasible	
Target T	**C1**	**C2**	**Cost**	
65	48	99	139,16	0 0 0 1 5 4 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 3 0 4 0
65	43	99	139,16	0 0 0 1 5 4 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 3 0 4 0
65	38	99	139,16	0 0 0 1 5 4 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 3 0 4 0
65	33	99	139,16	0 0 0 1 5 4 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 3 0 4 0
65	28	99	139,16	0 0 0 1 5 4 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 3 0 4 0
65	23	99	No solution	
65	18	99	No solution	
65	13	99	infeasible	
Target T	**C1**	**C2**	**Cost**	
65	99	55	139,16	0 0 0 1 5 4 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 3 0 4 0
65	99	50	139,16	0 0 0 1 5 4 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 3 0 4 0
65	99	45	139,16	0 0 0 1 5 4 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 3 0 4 0
65	99	40	139,16	0 0 0 1 5 4 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 3 0 4 0
65	99	35	156,62	0 0 0 1 5 4 0 0 2 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 5 0
65	99	30	No solution	
65	99	25	No solution	
65	99	20	infeasible	

D Annex D – Supplementary documents for case

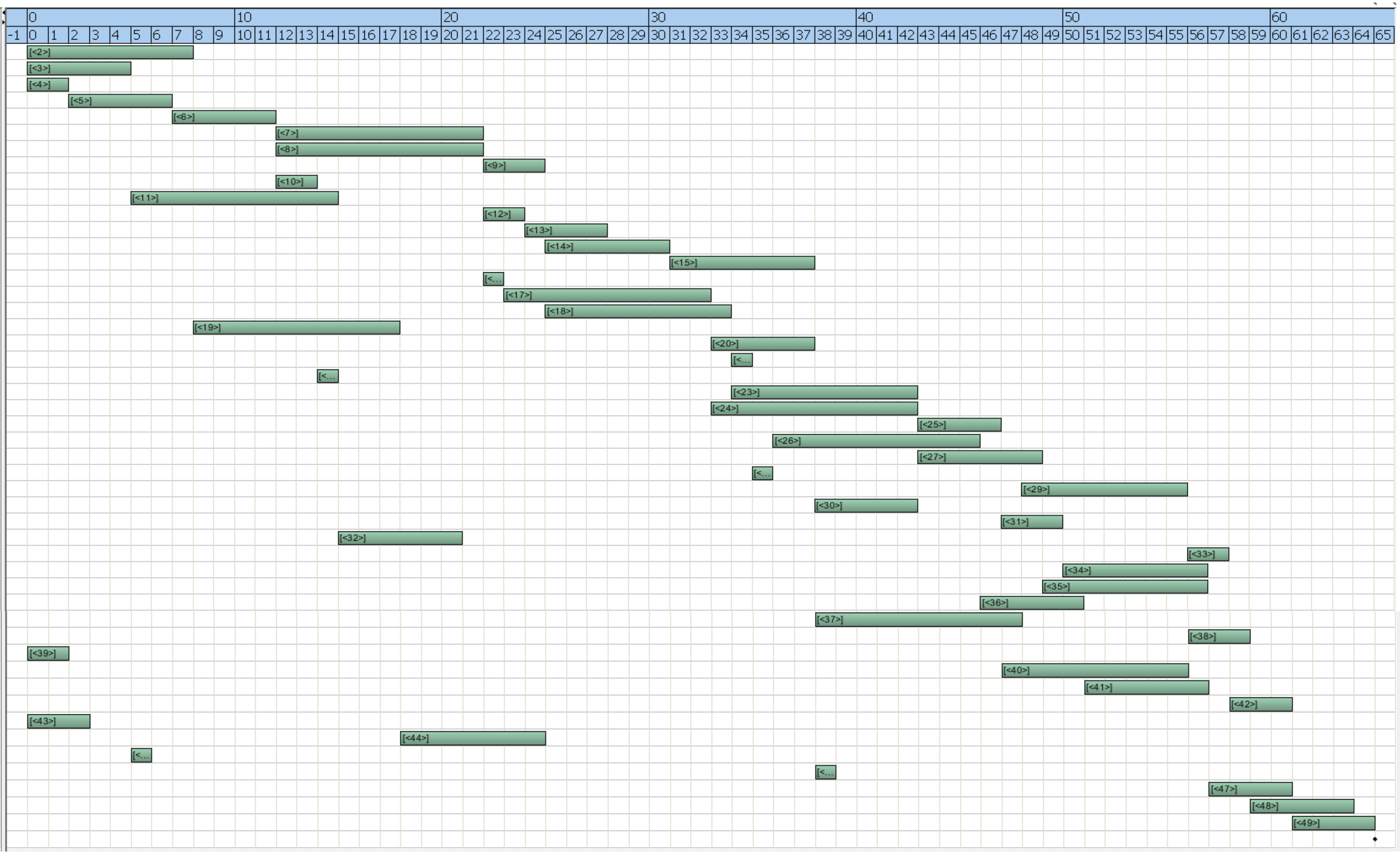
D.1. Resource Load Chart C1 and C2 for instance (65/28/46) for 50 activities



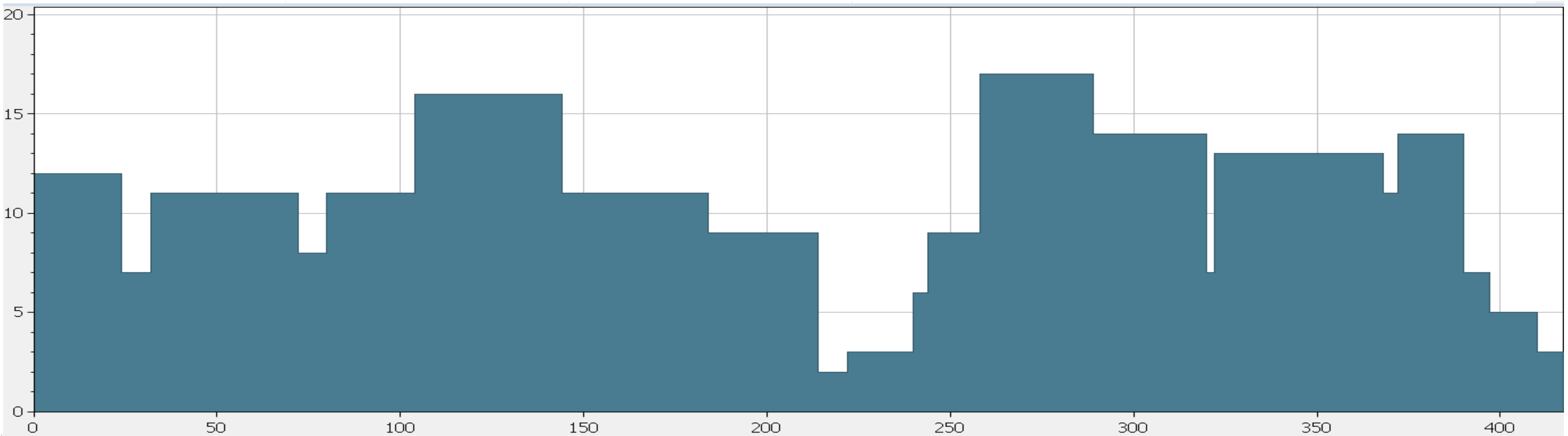


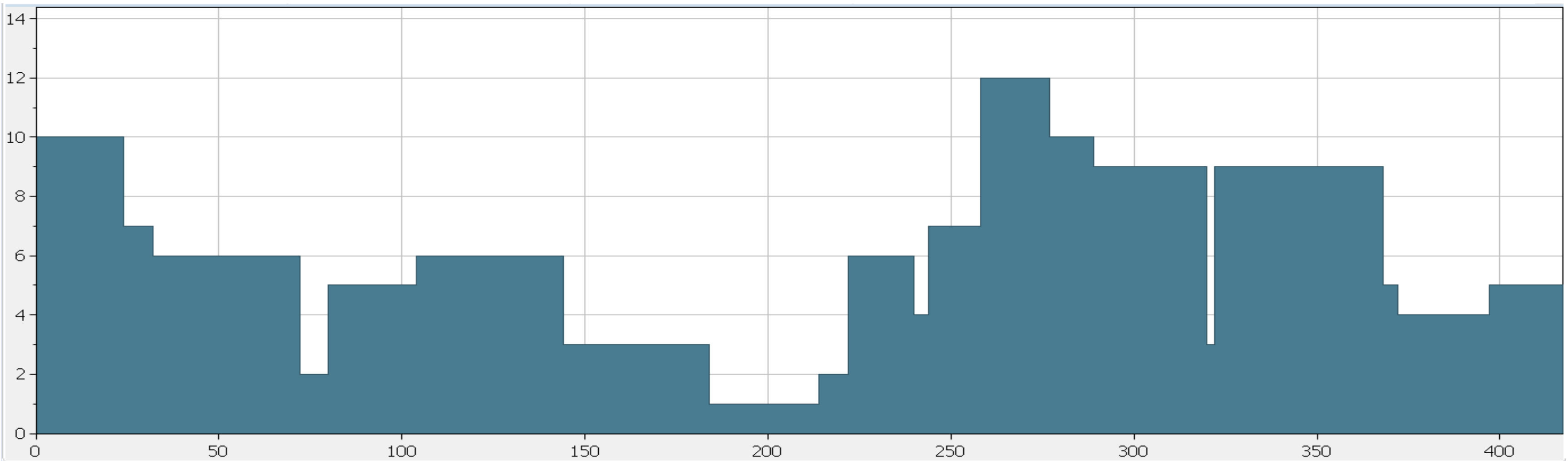
D.2. Gantt Chart for instance (65/28/46) for 50 activities

Start	End	Size
0	8	8
0	5	5
0	2	2
2	7	5
7	12	5
12	22	10
12	22	10
22	25	3
12	14	2
5	15	10
22	24	2
24	28	4
25	31	6
31	38	7
22	23	1
23	33	10
25	34	9
8	18	10
33	38	5
34	35	1
14	15	1
34	43	9
33	43	10
43	47	4
36	46	10
43	49	6
35	36	1
48	56	8
38	43	5
47	50	3
15	21	6
56	58	2
50	57	7
49	57	8
46	51	5
38	48	10
56	59	3
0	2	2
47	56	9
51	57	6
58	61	3
0	3	3
18	25	7
5	6	1
38	39	1
57	61	4
59	64	5
61	65	4
65	65	0



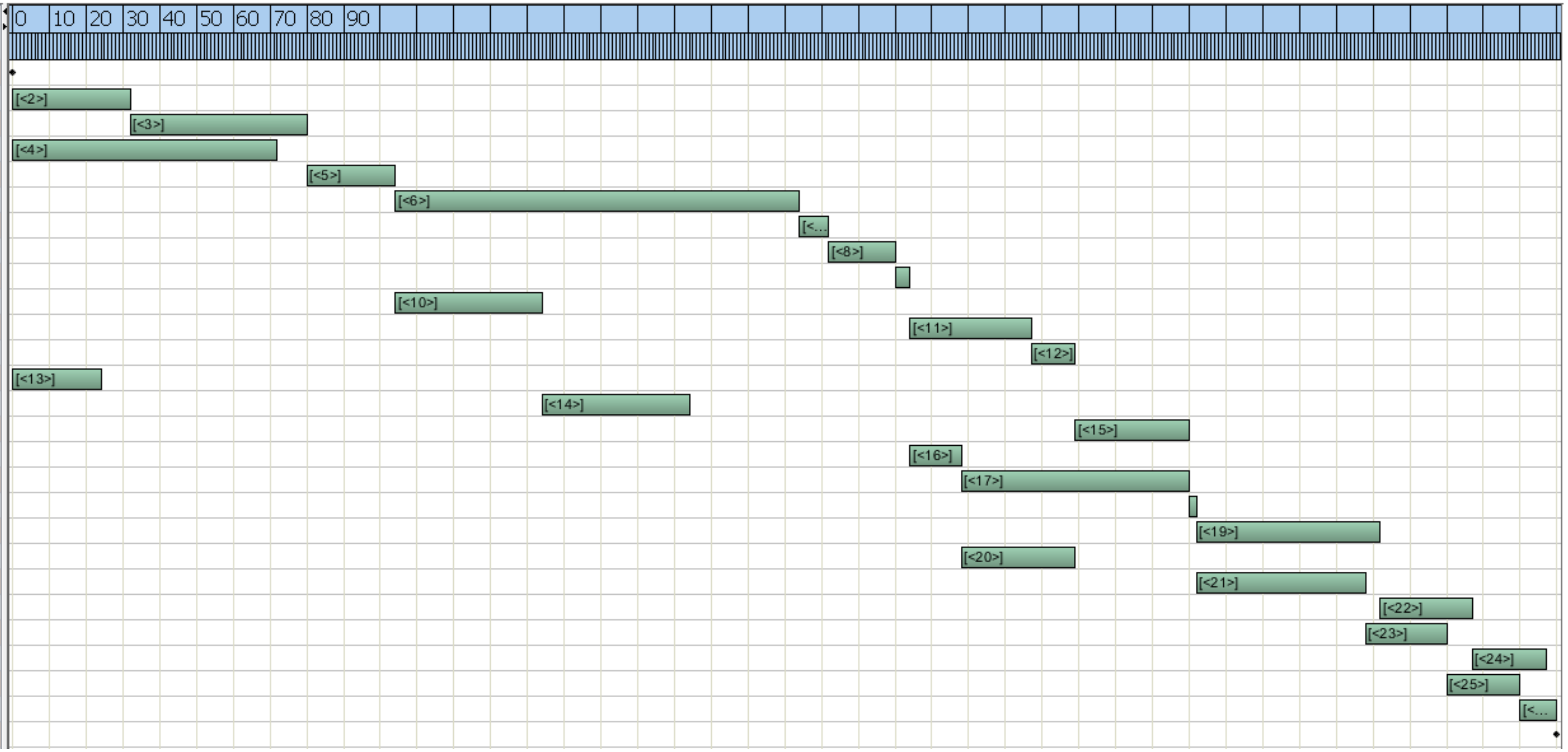
D.3. Resource Load Chart C1 and C2 for instance (420/17/12) for 25 activities





D.4. Gantt Chart for instance (420/17/12) for 25 activities

Start	End	Size
0	0	0
0	32	32
32	80	48
0	72	72
80	104	24
104	214	110
214	222	8
222	240	18
240	244	4
104	144	40
244	277	33
277	289	12
0	24	24
144	184	40
289	320	31
244	258	14
258	320	62
320	322	2
322	372	50
258	289	31
322	368	46
372	397	25
368	390	22
397	417	20
390	410	20
410	420	10
420	420	0



List of Figures

List of Tables

Acknowledgments

First and foremost, I am extremely grateful to my advisor Prof. Mauro Mancini and co-advisor Prof.ssa Costanza Mariani. Starting from the first days of our acquaintance from Industrial Project Management course, I have learned a lot thanks to their invaluable efforts and knowledge. Their courage to study AI and PM on this thesis gave me a great space to explore. Moreover, without their tremendous understanding and patience, it would be impossible to complete my study.

My deepest gratitude to Prof. Dr. Armagan Tarim who inspired me to write the thesis on this topic and provided invaluable support at the most though times of the thesis and guidance as an expert on this field.

Besides, this page of the thesis represents the end of a long journey. Throughout this story, lots of adventures has been experienced; some of them were pleasure and some were painful. The good ones motivated, and the bad ones taught. However, one stayed eternal, the favor and kindness of my loved ones. That is firstly to say, special thanks to my mother and father who have been so tolerant and wonderful through my thesis and to my dear sister Naz who gives me chance to prove I am the best kid of this family. Obviously, my love for you shall last forever. Likewise, to my friends who ask me if I am still writing my thesis, I am also thankful to you for your support and fun. Grazie! Any attempt at any level is achievable heartily with you.