POLITECNICO DI TORINO

TITLE OF THESIS:

Optimization of Internal Rate of Return and Net Present Value for investments

SUPERVISOR

Fabio Guido Mario Salassa

CANDIDATE

Andrea Perna

Academic Year 2023 - 2024

ABSTRACT

Optimization of Internal Rate of Return and Net Present Value for investments

This thesis aims to expose the effectiveness of scheduling techniques to get to the right investment channel through the analysis and optimization of the Internal Rate of Return (IRR) and the Net Present Value (NPV).
Optimizing, creating and analyzing project scheduling and various financial metrics will try to give an indication for the assessment of investment opportunities.
The performance of various algorithms created and used for this thesis, through statistical techniques will be evaluated, demonstrating the improvements and characteristics of the various proposals and giving a broader and financially innovative perspective of what project scheduling is today.

TABLE OF CONTENTS

Page

LIST OF TABLES

Tables Page

LIST OF FIGURES

Figure Page

1 CHAPTER

INTRODUCTION

Project scheduling is a fundamental aspect of project management that directly affects costs, resources, profits and people.
With the advance of ever more unusual techniques and design challenges, three different industries require an increasingly customized adaptation for each investment and scheduling choice.

Traditional methods of scheduling do not offer a vision that considers all the aspects involved in decision making, from scheduling to the choice of ad hoc financial metrics, we have therefore tried to meet a more advanced and tailor-made demand for design approaches.

Passing first from the theoretical and technical definition of each variable in play is then sought later in reading to give space to advanced computational techniques compared to the classics that reflect more criteria and have a more precise solution regarding the project scheduling and that better answer to the questions faced in investment decision making.

We will create algorithms and drivers of new design choices and respective datasets with detailed variables will be explained and analyzed in various forms, from the technical point of view (respecting the literature and the real way), statistical and graphic.

We will therefore try to give an answer to the way of seeing projects today in the widest possible spectrum and with a level of detail enriched by custom design standards.
Through method and analysis, attention will therefore be paid to questions relating to scheduling, the economic return to the various trade-offs between the criteria used and will try to figure out which algorithm is more reliable and competitive by observing the significant differences with the benchmarks used.

2 CHAPTER

QUALITATIVE AND QUANTITATIVE METRICS FOR INVESTMENT DECISION

2.1 Introduction

In the investment decision-making landscape, the application of both qualitative and quantitative metrics is essential for achieving aims and maximizing returns. Often these metrics are used under different assumptions and for achieving different aims, however these metrics used together may simplify the investment decision-making. This chapter research into these two types of metrics, explaining their unique roles and application impact in guiding investment decisions.

2.2 Quantitative Metrics

Quantitative metrics are fundamental in providing an aim, data-driven foundation for investment analysis. These metrics, often numerical in nature, allow for a measurable and comparative assessment of investment opportunities. Key quantitative metrics include financial ratios, cash flow analysis, and statistical measures. They offer a clear, empirical basis for decision-making, stripping away subjectivity to reveal the intrinsic value and potential risks of an investment.

The use of quantitative metrics is useful because through them you can make decisions with a certain confidence given by the assumptions and the underlying values.
Qualitative metrics can be used for different purposes, in addition to everyday life when making investment decisions such as buying a new car or creating a mortgage and can be used in a corporate investment context where investment decision-making scenarios are on the agenda.
As explained above the analysis of quantitative metrics can be developed through mathematics statistics and all the various numerical sciences that together try to model a problem, in our decision case, with the aim of giving excuses and providing possible future scenarios. [1]

Quantitative metrics serve as the navigational compass in the sea of investment opportunities, guiding investors through data-driven insights. Here below there are some of the fundamental metrics (significant in the overall scenario and in our case study), their calculations, and some graphical representations to elucidate their application and application:

2.2.1 Price-to-Earnings Ratio (P/E)

The P/E ratio is a fundamental resource for investors aiming to estimate the market's valuation of a company's earnings, based on their historical values or in relation to the industry average.

$$EPS\ (P/E) = \frac{Net\ Income}{Number\ of\ Outstanding\ Shares}$$

It is calculated by dividing the Net Income of a company by the Number of Outstanding Shares around the world.
EPS gives a sign of how much money a company makes for each share of its stock, providing a gauge of its profitability.
An elevated P/E ratio could indicate that the company's stock is overvalued, possibly due to investors' opinion that expecting high growth rates in the future, and so there might be willing to pay more for the stocks of this company under analysis.
On the other hand, a lower P/E might suggest an undervalued stock or if the company is not expected to grow significantly maybe because is facing challenges, also for the investors, a lower P/E ratio define a lack of confidence by the market in the company's expected prospect.
By comparing the P/E ratio of the same company over time, investors can find opportune moments to buy or sell.
This indicator, how the definition suggests in the field of investment decision, is commonly used for buying or selling stock from a company.

2.2.2 Debt-to-Equity Ratio (D/E)

The Debt to Equity (D/E) ratio is another essential financial metric, particularly in terms of understanding a company's financial risk profile and use of capital. Here's a detailed look at the D/E ratio and its components:

$$D/E = \frac{Total\ Liabilities}{Shareholders'\ Equity}$$

It is common to see the D/E Ratio of a company in comparison to the others in the same industry and possibly within the same revenues/cost size (usually with a simple bar graph).
A rising D/E ratio may show that a company is gradually relying on debt to finance its growth, which could increase its risk during economic downturns. Investors use this metric to evaluate the financial health and stability of a company, furthermore excessively high D/E ratios that could signify financial distress or a higher risk of bankruptcy in adverse conditions. This metric is fundamental for a general overview of a certain company. However, like any financial metric, it should be considered alongside other indicators for a complete financial analysis.

2.2.3 Return on Equity (ROE)

For a complete overview after the explanation of P/E and D/E, the ROE helps the investors to figure out the profitability and the efficiency in generating a company's return on the investment from its shareholders.

$$ROE = \frac{Net\ Income}{Shareholders'\ Equity}$$

The ROE measures a company's ability to generate return, from a monetary point of view, from its equity investments. It is commonly associated with the return on the investment family financial metrics.

The "Net Income" part:

- Net Income is the profit a company earns after deducting all its expenses, including operating and capital expenses, investments of several natures, interest, taxes, and other charges.
- Net Income is reported on the company's income statement document and is a key driver of the company's growth.

The "Shareholders' Equity" part:

- Shareholders' Equity stands for the net value of a company, or in other words, what remains in terms of capital borrowed after all liabilities have been paid off.
- It can be calculated as Total Assets subtract the Total Liabilities and this type of data is reported on the company's balance sheet.
- Shareholders' Equity includes common stock, preferred stock, kept earnings, and treasury stock owned.

By analyzing ROE over time, the investors and stakeholders can find companies that deliver high returns on equity, showing strong performance in terms of product or service and a potentially profitable investment.
High ROE values may indicate a company is effectively using its assets to generate profits, a positive sign for investors, however it is complicated to analyzed by itself. [2]

2.2.4 Net Present Value (NPV)

We begin the discussion of investment decision considering the tools available that help in our context to choose the best project or item in the context of investment.
On the other hand, there is a value that came in our help, it is called Net Present Value (NPV). Thanks it, we could compare a several stand-alone projects together and choose the one that fits better for us, just applying a simple rule, called the NPV Investment Rule: "*When making an investment decision, take the alternative with the highest NPV. Choosing this alternative is equivalent to receiving its NPV in cash today*". [2]

To summarize, the NPV rule says that we have to compare the project's NPV to zero (the NPV of doing nothing) and accept the project if its NPV is positive, otherwise, reject it if its negative.

Here below the Net Present Value formula:

$$NPV = \sum_{t=0}^{n} \frac{CF_t}{(1+r)^t} - I_0$$

Where:

- *I_0 = Initial investment at time t=0 (usually negative Cash flow, called cash outflow), typically include the initial investment and any additional costs associated with the project.*
- *n = Number of times in periods, for the calculation consider the entire lifespan of the project.*
- *CF_t = Net Cash Flow during the period t (usually positive Cash flow, called cash inflow, the cash flows are estimated for each period of the project.*
- *r = Discount rate, it is a critical factor in NPV calculation and in the next chapter we will see the importance of this component, it reflects the time value of money, the principle that "One dollar today is worth more than a dollar in the future".*
  It can be based on different assumptions with different scenarios but in a poor language it is associated to the cost of capital.

Interpreting the NPV is easily, achieved the NPV from the calculation there are three possible results for each stand-alone project.
The NPV could be Negative (less than zero), and it suggests that the cash outflow spent on the initial investment is greater than the cash inflow generating by the project itself, in other words, the project's costs outweigh the benefits and so the investment may result in a net loss.
The NPV could be positive (greater than zero), and so indicates that the project earnings (the cash inflow discounted to present value) exceed the initial costs anticipated at time zero in addition to the other costs (also discounted). It means that if the assumptions to the future cash flow are correct, the investment is likely to be profitable and could be a good choose in an investment decision.

In alternative, the NPV could be zero (equal to zero). In that case choosing the project or doing nothing has the same impact because the NPV of the project implies a break-even situation.

There are so many applications of the NPV as said before, here below a simple example of a NPV analysis with some chart on output, created by a simple script with a few line of code explained here below.
The assumption and constraint used for the creation of the code are simple and just for the application, the forecast time is set to 5 years, an initial investment of 100 $ with a positive inflow starting at the first year after the investment.
Furthermore, for the calculation of the Graphic NPV a rate of return at 10% is used for the increased inflow.

```
# Libraries
import numpy as np
import matplotlib.pyplot as plt
from numpy_financial import npv

# Assumption
initial_investment = -100  # Negative for outflow
cash_flows = np.array([20, 30, 40, 50, 60])  # Positive for inflow
years = np.arange(0, 6)
discount_rate = 0.1  # 10%

# Calculate NPV
npv_value = npv(discount_rate, np.append(initial_investment, cash_flows))

# Create Cash Flow Graph
plt.figure(figsize=(10, 6))

# Initial Investment
plt.bar(0, initial_investment, color='red', label='Initial Investment')

# Cash Inflows
plt.bar(np.arange(1, 6), cash_flows, color='green', label='Annual Cash Inflows')

plt.title('Project Cash Flows')
plt.xlabel('Year')
plt.ylabel('Cash Flow')
plt.xticks(years)
plt.axhline(0, color='black', lw=1)
plt.legend()
plt.grid(True, axis='y')
plt.show()

# Cumulative Cash Flow Graph
cumulative_cash_flows = np.cumsum(np.append(initial_investment, cash_flows))
```

```
plt.figure(figsize=(10, 6))
plt.plot(years, cumulative_cash_flows, marker='o', linestyle='-', color='blue')
plt.title('Cumulative Cash Flows Over Time')
plt.xlabel('Year')
plt.ylabel('Cumulative Cash Flow')
plt.xticks(years)
plt.axhline(0, color='black', lw=1)
plt.grid(True)
plt.show()
```

For clarifying the above script created print in output a value of NPV equal to 44.43$, a positive value, so, in this case the acceptance of the project is implicit.
Here below the chart created by running the previously code in python in a environment that allows after the license of the several libraries installed.

The first chart called “Project Cash Flow” captures the financial trajectory of the theoretical investment project (explained before) across a five-year span. At the outset, depicted by the red bar at the year zero mark, we observe a substantial negative cash flow, signifying the initial investment.
Moving to the right, along the time axis marked by years, a series of green bars ascend above the breakeven line, each representing the annual cash inflows the project is expected to yield. These inflows are indicative of the revenue or savings generated by the project and display a pattern of growth (the rate of growth is explained before in the next chapter), suggesting that the project's financial benefits increase as it matures. The vertical “Cash Flow” axis quantifies the cash movements: outflows below the line and inflows above it.
This graphical representation serves as an insightful tool for investors and stakeholders to assess the initial financial commitment against the potential profit gains along the years, it offers a clear visual narrative of the investment's anticipated financial evolution from initial costs to eventual returns.

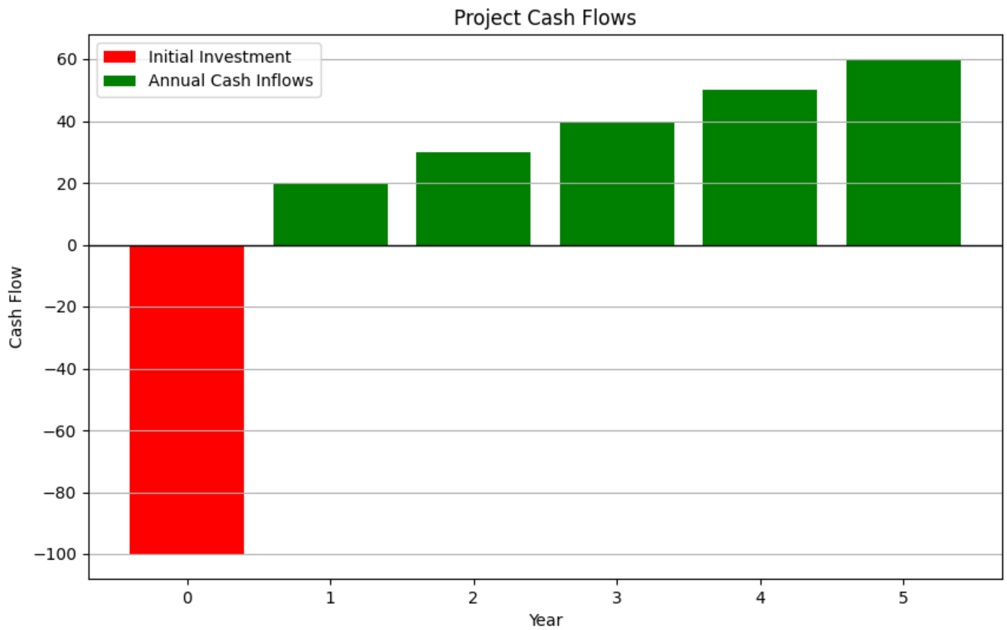

Figure 1: Graph of project Cash Flows

The second chart instead provides a visual representation of the cumulative cash flows over the life of the project.
It begins with the initial investment at year zero, showing a steep drop below the zero line at level -100$.
As time progresses, each year adds the annual cash inflow to the previous total.
By the end of the fifth year, the graph reveals that the cumulative cash flow has risen well above the initial outlay, reflecting the project's return on investment over time. This line graph also demonstrates the project's overall financial health of the investment, mapping out the cost and the profit.

This graph is particularly useful for investors and project managers as it visually communicates the time it takes for an investment to pay off and the rate at which profitability is achieved over the project life.

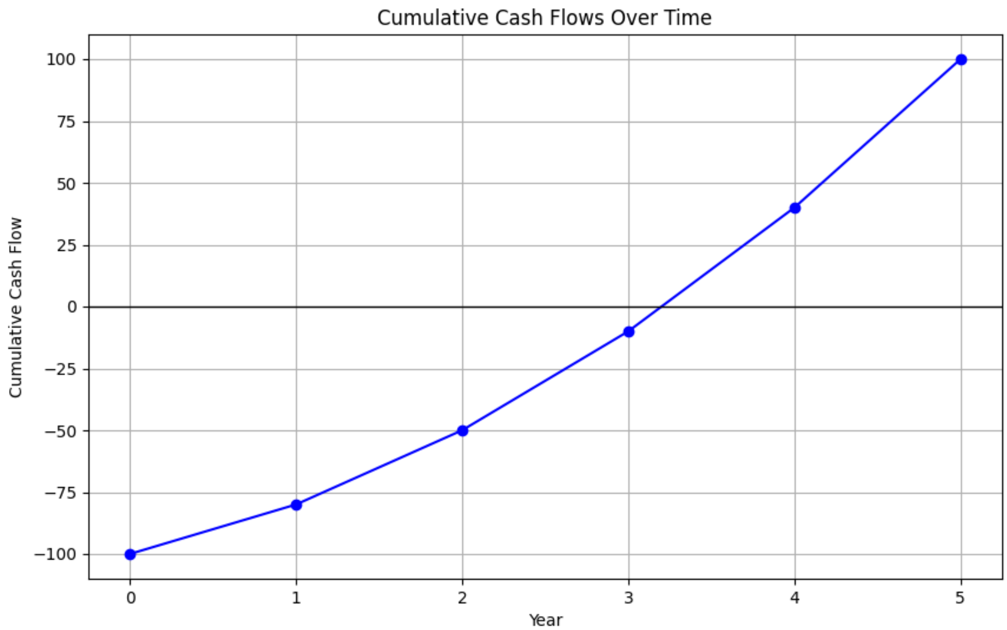

Figure 2: Graph of cumulative Cash Flows over time

In addition to these charts, there are various other ways to visualize NPV and related financial metrics:

Scenario Analysis Graphs: These graphs depict the NPV under different scenarios, such as best-case, worst-case, and most-likely case scenarios, providing a visual representation of potential risks identifying other metrics like IRR that we will see in the next paragraph.
Probabilistic NPV Graphs: These use probability distributions for uncertain variables like cash inflows and discount rates, displaying a range of possible NPVs.
These are only some examples of charts, each graphical representation offers unique advise and is selected based on the specific purpose of the investment decision needed.

2.2.5 Internal Rate of Return (IRR)

In the scene of financial decision-making tools, the Internal Rate of Return (IRR) stands as a critical and fundamental measure for evaluating the profitability of investments. The IRR is particularly insightful for investors and not only, as it represents the discount rate at which the Net Present Value (NPV) of all cash flows from a particular project or investment equals zero. The IRR, as we will see below, provides a break-even rate of return. [3]

There is a rule, called "The IRR Investment Rule" that simplifies the selection process of projects and says:
"Choose the project with the highest IRR, provided it exceeds the required rate of return".
This rule implies that the chosen investment is expected to generate a rate of return greater than the opportunity cost of capital.

The IRR is calculated using the equation below, which is conceptually like the NPV formula but solves for the discount rate that create the NPV equal to zero:

$$0 = \sum_{t=0}^{n} \frac{CF_t}{(1 + IRR)^t} - I_0$$

Where:

- I_0 is the initial investment at time t=0 (usually a cash outflow).
- n is the number of time periods.
- *CFt* is the net cash flow during the period t.
- *So, IRR* is the discount rate that makes the NPV of the project equal to zero.

The interpretation of IRR is intuitive, if the IRR of a project exceeds the required rate of return, it signifies that the project is expected to generate a return greater than the cost of the capital borrowed and used. If the IRR is less than the cost of capital, the project will destroy value (So the NPV will be negative).
A project with an IRR that equals the cost of capital will break even and generate a NPV equal to zero as explained in the formula above.

There are so many applications of the IRR, here below, a simple example of a NPV analysis with some chart on output, created by a simple script with a few line of code explained.
The assumption and constraint used for the creation of the code are simple and just for the application, an initial investment of 1000 $ with a positive inflow starting at the first year after the investment, reported below on the "cash_flow" array.

```
# Libraries
import numpy as np
import matplotlib.pyplot as plt
from numpy_financial import npv, irr
```

```
# Initial investment and expected cash flows
initial_investment = -1000
cash_flows = np.array([200, 250, 300, 350, 400])

# Calculate the internal rate of return (IRR)
irr_value = irr(np.append(initial_investment, cash_flows))

# Generate a range of discount rates
discount_rates = np.linspace(0, 0.3, 200)

# Calculate the net present value (NPV)
npvs = np.array([npv(rate, np.append(initial_investment, cash_flows)) for rate in discount_rates])

# Create the NPV graph
plt.figure(figsize=(6, 4))
plt.plot(discount_rates * 100, npvs, label='NPV Curve')
plt.axhline(0, color='black', lw=1, linestyle='--')
plt.axvline(irr_value * 100, color='red', linestyle='--', label=f'IRR ({irr_value*100:.2f}%)')

# Graph generation
plt.xlim(0, 30)
plt.ylim(-1000, max(npvs)+100)
plt.title('NPV Curve and IRR')
plt.xlabel('Discount Rate (%)')
plt.ylabel('NPV ($ millions)')
plt.legend()
plt.grid(True)
plt.show()
```

As said before, the IRR of a project provides useful information regarding the sensitivity of the project's NPV to errors in the development of its cost of capital.

For visualize the concept of IRR, a Python script similar to the one for NPV analysis has been used. The script shows the IRR relative to different discount rates, and here below the graph generated.

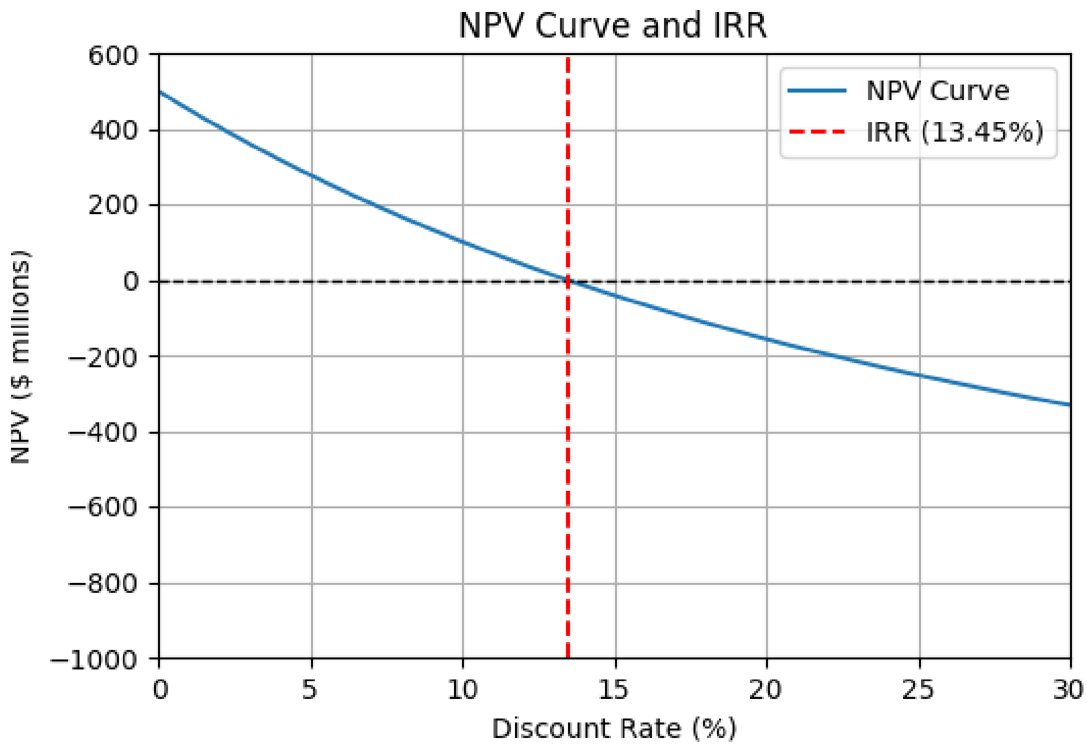

Figure 3: NPV and IRR Curve

The code has been executed, and the IRR for the project has been calculated as 13.45%. This matches the discount rate at which the NPV curve intersects the horizontal axis (Discount Rate%) at an NPV of zero on the graph. The graph, showing the relationship between the discount rate and the NPV, highlighting the IRR as the point where the NPV equals zero.

2.2.6 Payback Period

The Payback Period is an essential metric in project evaluation, particularly for understanding the liquidity aspect of an investment and its possible return in periods. It measures the duration required for an investment to repay its initial cost (investment at time zero) from its cash flows (the future inflow). Unlike NPV and IRR, the Payback Period does not account for the time value of money, making it a simpler metric that focuses solely on the time aspect of profitability, so in these metrics the discount factor is not used. [3]

Payback Period Rule

The decision rule for the Payback Period is straightforward: "Prefer investments with shorter payback periods." This rule is based on the premise that investments that pay back their initial costs sooner are less risky and provide quicker liquidity (the concept is similar as the duration used in the Bond evaluation).

Formula for Payback Period:

There are two possible ways to evaluate the Payback Period.
It is possible to calculate by summing the cash flows until the initial investment is recovered, and so, counting the period of the final CF_t used, or in alternative, we can calculate as initial Investment over the average cash inflow.

$$Cumulative\ Cash\ Flow_t = \sum_{i=0}^{t} CF_i$$

Or

$$Payback\ Period\ (PB) = \frac{Initial\ Investment\ (I_0)}{Average\ Cash\ Flow}$$

Where:
- CF_i is the net cash flow during period "i".
- t is the time where the cumulative cash flow equals or exceeds the initial investment.

Continuing using a visual demonstration, to demonstrate the Payback Period, here below a simple script on Python that calculates and plots a smooth curve of cumulative cash flows. In this case, the payback period is indicated at the point where the curve intersects the x-axis.

```
# Libraries
import numpy as np
import matplotlib.pyplot as plt

# Assumptions
```

```
initial_investment = -1000
cash_flows = np.array([200, 250, 300, 350, 400])

# Calculate the cumulative cash flows
cumulative_cash_flows = np.cumsum(np.append(initial_investment, cash_flows))

# Find the payback period
payback_period = np.where(cumulative_cash_flows >= 0)[0][0]

# Creation of graph
plt.figure(figsize=(10, 6))
years = np.arange(0, len(cumulative_cash_flows))
year_interp = np.linspace(years.min(), years.max(), 500)
cumulative_cash_flow_interp = np.interp(year_interp, years, cumulative_cash_flows)

plt.plot(year_interp, cumulative_cash_flow_interp, label='Cumulative Cash Flow')

# Highlight the payback period on the graph
plt.axhline(0, color='black', lw=1, linestyle='--')
plt.axvline(payback_period, color='red', linestyle='--', label=f'Payback Period = Year {payback_period}')

plt.title('Payback Period Analysis')
plt.xlabel('Year')
plt.ylabel('Cumulative Cash Flow ($)')
plt.legend()
plt.grid(True)
plt.show()
```

Several assumptions and decision are made on the constraints used for the chart and all of them are similar to the previous python application and showed in the code above.

The output graph is the following one.

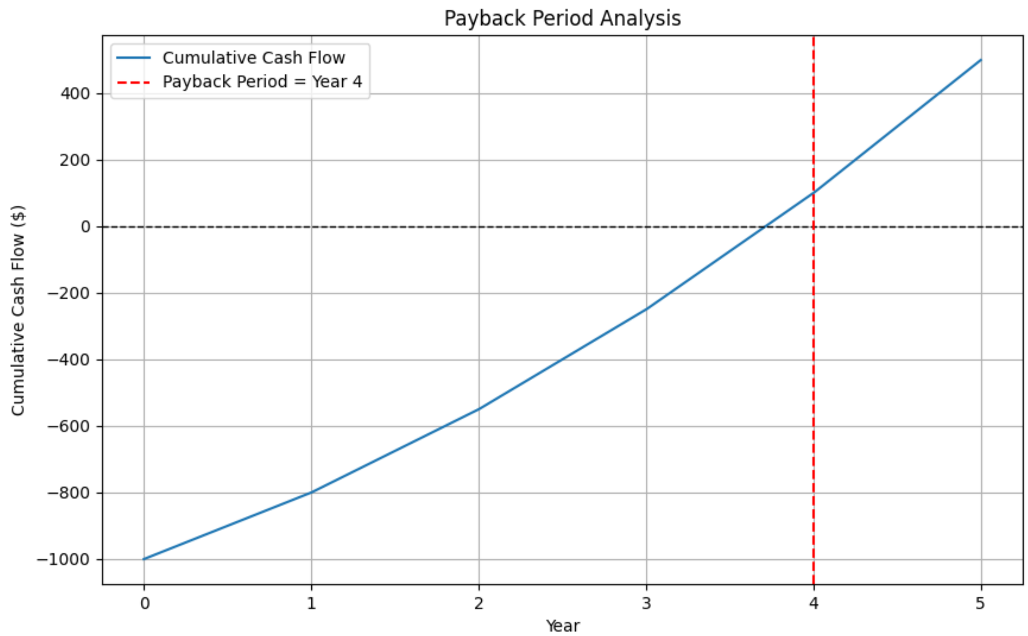

Figure 4: Payback Period

The graph presents the cumulative cash flows as a smooth curve, showcasing the recovery of the initial investment over time. The Payback Period is marked where the curve intersects the x-axis, as explained before. This visualization provides a clear and intuitive understanding of when the project becomes profitable, in other world, where is the point of mass of the entire project, it is a useful tool for investors who prioritize liquidity and risk minimization.

Application of the Payback Period:

The Payback Period is particularly favored in industries where rapid recovery of investment is crucial for the future. It is also used as a preliminary screening tool before conducting more sophisticated analyses like NPV or IRR.
Most of the time, the Payback Period is used with the NPV and IRR for having a complete and overall view for an initial screening of a project.

2.3 Qualitative Analysis

While quantitative analysis is indispensable, it often falls short in capturing the full spectrum of factors influencing an investment decision. This is where qualitative metrics come into play. These metrics incorporate the non-numerical aspects such as market trends, brand value, management quality, and company culture. Qualitative analysis provides depth and context, offering insights into the less tangible, yet equally pivotal elements that can significantly sway investment outcomes.

Qualitative methods are used on fewer occasions in an investment valuation but are no less important.
As for investments through qualitative methods, we find ourselves evaluating not yet developed realities that do not yet have certain cash flows or do not have them at all, in which the only variables in play are not numerical and therefore thanks to the qualitative metrics placed side by side to the quantitative ones for as much as possible and taking external data one looks for an answer to the investment decision-making. This scenario is widespread within Venture Capital or for investors that invest in early-stage companies or ideas in which the above characteristics are most used.

2.4 Qualitative tools

Investment decision-making is made principal by financial metrics and numerical data as explained above with the explanation of Qualitative Analysis.
However, it also demands a thorough understanding of qualitative factors that influence the potential success or failure of an investment.
This understanding is made through a variety of analytical tools designed to assess the sub-problems that financial and numerical prevalent ratios alone cannot capture. These tools enable investors to research into the operational, strategic, and competitive aspects of businesses or project decisions, providing a holistic view of investment opportunities.
The subsequent section introduces the principal tools and metrics utilized in qualitative analysis, each offering unique insights into the multifaceted nature of investment decisions.
After the discussion of these tools, a view of some core metrics in aim of qualitative analysis is discussed.

2.4.1 SWOT Analysis

The SWOT Analysis is the most popular strategic tool used to identify and evaluate the Strengths, Weaknesses, Opportunities, and Threats involved in a project or in a business idea. It is an integral part and a core instrument of qualitative analysis that allows investors to understand the internal and external factors that could impact the success of an investment.

The primary purpose of a SWOT Analysis is to facilitate a long-term forecast that helps in strategic planning. It helps to model a niche in the competitive landscape or new market opportunities. By categorizing factors into the SWOT quadrants, as we see below, investors and managers can make informed decisions that align with the company's principles and market view.

There are four components that characterize the SWOT Analysis, the first two here below are internal parameters the further two instead reflect the entire ecosystem and represent the external factors:

- Strengths:
  These are the attributes of the business or project that give it an advantage over others company or in comparison between projects.
  Strengths could include a strong brand, proprietary technology, superior quality of products or working status, strategic alliances, or a skilled workforce in terms of resources.

- Weaknesses:
  These are the areas where the business or project may be at a disadvantage compared to competitors.
  Weaknesses, the other face of the internal components, might involve a lack of patents, weak brand recognition, limited financial resources, or gaps in the product or available task.

- Opportunities:
  External chances to improve performance in the environment.
  Opportunities could be favorable market trends, regulatory changes, technological advancements, or untapped customer segments in the field of a product or a service.

- Threats:
  External elements in the environment that could cause trouble for the business or project.
  Threats, the other face of the external components, may include a changing regulatory framework, increasing competition, changes in consumer behavior, or macro and micro economic changes.

Here below an explained example of what it means create a SWOT Analysis.

SWOT Analysis

Strengths	***Weaknesses***
Skilled workforce *Competitive advantages* *Advanced technology* *Company reputation*	*Resource limitations* *Dependence on specific* *Service quality issues* *Geographical limitations*
Market trends *Technological innovations* *Strategic partnerships* *Favorable regulations*	*Competition* *Regulatory changes* *Economic risks* *Disruptive technologies*
Opportunities	***Threats***

To recreate the SWOT Analysis explained above, a cross-functional team involved in a project or with the aim to make an investment decision with able to insights into the company's operations, market, and competitive landscape should be assembled have to complete some technical steps include:

Data Collection:
Gathering data from internal records, market research, financial analysis, and stakeholder feedback.

Brainstorming Sessions:
Drive the team to identify as many factors as possible before classifying them into four categories.

Evaluation and Prioritization:
Evaluating the significance of each factor and prioritizing them based on their potential impact on the investment decision.

Strategy Development:
Using the SWOT Analysis's advice to formulate strategies that encourage strengths, mitigate weaknesses, exploit opportunities, and avoid threats.

Investors utilize SWOT Analysis, in the application of investment decision, to understand where a company stands before committing capital or starting a project. It helps in inform whether a company's position aligns with investment objectives of stakeholders and if it possesses the capabilities to overcome its weaknesses and threats creating a possible future profitability.

In conclusion, the SWOT Analysis is the most versatile qualitative tool that can be applied to a broad range of situations in investment decision-making. By providing a structured approach to evaluate a company's strategic position, it offers a clear perspective that supports complex decision-making processes. In further chapters we will explore the practical applications of other methodologies, also in comparison to this one, illustrating its value to the real-world scenarios.

2.4.2 PESTLE Analysis

PESTLE Analysis is a fundamental framework used to have a complete view of the macro factors that may have a large and deep impact on an organization's performance. This analytical tool considers five areas: the Political, Economic, Social, Technological, Legal, and Environmental factors, hence the name PESTLE, and thanks to that it can shape investment opportunities and risks. In this chapter, we will explore how PESTLE Analysis is used as an indispensable tool for investors to understand the border and the context in which a company operates.

Regarding the Political factors we can mention the Political stability, government policies, taxation changes, and tariffs, all of these are considered because they can directly affect a company's operations and profitability. For instance, a governance change can lead to policy shifts that impact investment incentives or regulatory frameworks or in a simplistic view a change in priorities.

Regarding the Economic factors we can consider the economic growth trends, interest and exchange rates, inflation, employment levels, consumers level, disposable income of consumers and further. Economic indicators can significantly influence demand for a company's products and services or activity routine and alter its financial stability and the assumption made.

In what concern the social factors, we are looking at Social trends, demographic changes, consumer behaviors, and cultural behavior, those metrics can affect the market size for a product and the potential for market rise or decline. Social factors can also impact labor markets and corporate social responsibility practices for example.

Regarding the Technological factors, instead, we are looking at the Rapid technological advancements that can create new opportunities and also declare existing services obsolete. Technology impacts not just product/service innovation but also the efficiency of business operations and supply chain logistics, a fundamental aspect to monitor in the project context.

In The Legal factors purpose the main metrics to consider are employment, antitrust, health and safety, and environmental regulation can affect costs, and demand that affects the Legislation areas. Compliance with laws and regulations is not optional for several reasons, and non-compliance can have a consequence to liabilities.

Finally, the Environmental factors are regarding the Environmental considerations are increasingly important due to climate change, resource scarcity, and shifting consumer preferences towards sustainability are only the main into this consideration. Environmental factors can influence corporate strategies and operations in the present and for the future, particularly for industries like energy, manufacturing, and agriculture in which the consumption of electricity or the use of earth resources are the main business margin.

Conducting PESTLE Analysis

To develop a PESTLE Analysis, investors, and analysts, collect data on each of the six elements explained before, often starting with few information for each type and then drilling down into more detail for factors that appear particularly salient and interesting for the prospect of the project or the company.
This process can involve a range of activities, from reviewing economic projection and policy file to conducting market research and technological overview.

Application of PESTLE Analysis

By identifying how each PESTLE factor affects the investment, analysts and stakeholders can draw conclusions about the potential risks and opportunities within assumption the metrics. This analysis is crucial for making informed investment decisions.

PESTLE Analysis provides investors with a comprehensive understanding of the external landscape that influences a company's potential for success in short and long-term vision. By considering each factor in the PESTLE framework, investors can make more robust and resilient investment decisions, with a clear view of the border environment.

2.4.3 Porter's Five Forces Analysis

Porter's Five Forces is a qualitative model used to analyze an industry's environment and to formulate strategies within the data provided by the market. It is a vital tool in qualitative analysis for investment decisions, providing insights regarding the competitive forces that can embrace profitability.
This chapter will clarify the investor prospective and how they can use Porter's Five Forces to discover the competitive attractiveness of a project or a sector.

The Porter's Five Forces is composed of five segment forces that shape a strategy, here below the objective.

- Threat of New Entrants:

This force examines how easy or difficult it is for competitors to join the marketplace. The threat of new entries depends on the existence of barriers to entry, economies of scale, product differentiation and variety, and access to industry supply channels. A high threat of new entrants can diminish a firm's profitability in the future because they have to spend more to retain customers.

- Bargaining Power of Suppliers:

Suppliers have power in the market when there are few substitutes for their goods, they serve a significant number of buyers, or there is a cost associated with switching suppliers. Powerful suppliers can raise material costs, affecting the cost structure and profitability of firms in the industry, a low bargaining power for company suppliers is required for optimal profitability.

- Bargaining Power of Buyers:

Buyers can negotiate power when they purchase large volumes or when the products are standardized so they can easily switch to a competitor's product. This force, that are the lowest in a monopoly or duopoly scenario, examines the influence that customers have on the pricing and quality of goods within the market and the respective company.

- Threat of Substitute Products or Services:

The presence of alternatives, in terms of products variety, can limit the potential of an industry. Substitutes products that offer an attractive price-performance trade-off can reduce industry profitability by limiting the price firms can charge for their products or services.

- Rivalry Among Existing Competitors:

This incorporates the intensity of competition among existing players in the market, which can be influenced by the number of competitors, rate of industry growth, and the diversity of rivals. High rivalry leads competition to price wars, advertising battles, and product innovations, which can erode profit margins.

Utilizing Porter's Five Forces for Investment Analysis

Using Porter’s Five Force can understand the competitive dynamics of an industry and identify the strategic positioning of a company in comparison to the existing or possible rivals. By assessing how these forces affect a sector or a environment, investors can make informed decisions about the risks and the possible returns related to the investments in different industries.

Conclusion

Porter's Five Forces Analysis is a must have framework for stakeholders or who need to better understand the competitive landscape. It aids in recognizing the underlying pressures in an industry and discovers the way for strategic thinking in investment selection. This comprehensive analysis can significantly influence investment decisions, providing a clearer path to understanding where the profitability hides around sectors and companies.

2.4.4 Brand Equity Models

Brand equity signifies a brand's value as perceived by consumers and is a crucial qualitative factor for investment decision making. It represents the commercial value that derives from consumer perception of the brand name and product itself. This chapter aims to explore the concept of brand equity, its relevance in evaluation context, and the models used to analyze it.

A brand with high equity is seen as reliable, high-quality, and leading to customer loyalty, pricing power, and a competitive advantage in the market compared to other companies. For investors, understanding a company's brand equity is essential because impacts the company's profitability and market share.

Brand equity is made of different components, involving various dimensions:

Brand Awareness that extent to which consumers are familiar with the qualities or image of a particular brand.

The Brand Associations that is composed of attributes, memories, and relationships that consumers connect to the brand.

The Perceived Quality represents the consumer's perception of the overall quality of a product or service relative to alternatives.

Brand Loyalty defines the tendency of consumers to continue buying the same brand repeatedly without choosing other competitors.

Furthermore, there are several ways to measure Brand Equity with relative models.

For example there are the Keller's Brand Equity Model (Customer-Based Brand Equity Model - CBBE), that helps a brand needs to take to build a strong equity, the Aaker's Brand Equity Model, that identifies five categories of brand equity: brand loyalty, brand awareness, perceived quality, brand associations, and other relevant such as patents; the BrandZ Model, developed by Millward Brown, that measures brand equity based on brand data collected from interviews with consumers, combining financial data to assess how and how much a company's value is related by its brand.

Investors can use brand equity models to measure the impact of a company's brand on its forecast earnings. A high "Brand Equity" can lead the company to profits and growth, touching the consumers are willing to pay relative to products from a reputable brand.

Conclusion

Brand equity models provide a step-by-step approach to evaluating the intangible assets of a company's brand. In a marketplace where the perception of the customers can determine success, brand equity is a crucial consideration for making informed investment decisions based on true data, financial and not.

While qualitative metrics are invaluable, they are most powerful when used in conjunction with quantitative analysis, brand equity is the proof. Together, they allow investors to make the best decisions by considering both the measurable performance and the qualitative factors that drive that performance in the moment and in the future, in terms of profitability and returns.

2.5 Qualitative Metrics

Qualitative metrics include various elements that are not readily in a quantitative way but have an impact on an investment's value. They can involve management expertise, brand strength, positioning of the company, regulatory environment, and market.
These intangible and tangible factors can significantly affect a company's ability to generate future earnings and achieve strategic goals.

Here below the core qualitative metrics used in the most popular decision in investment, all of them are applicable on a project or a company, in our case we can also use it with the aim of discovering the best investment decision among several alternatives.

2.5.1 Management Quality

Management quality is a qualitative metric that significantly influences an organization's success and, consequently, its investment attractiveness. This metric is related to the abilities, experiences, leadership styles, and decision-making processes of a company's management team. It reflects how well the management faced challenges and his ability to achieve strategic goals.

There are different components that compose the Management quality metric, here below a few.

Leadership and Vision:

The ability of the company to create a clear direction for the future and encourage employees to follow the same aim. This includes communicating a strategic vision that aligns with the company's goals and adapting that vision as the market environment evolves.

Experience and Track Record:

The performance of the management team, including their experience in the industry and past successes or failures. A strong historical record of growth, innovation, and effective crisis management is often indicative of high-quality management.

Decision-Making:

Is the process by which management makes strategic and operating decisions. This includes not only the results of those decisions but also the speed and the adaptability of decision-making processes.

Corporate Governance:

The structures and practices in place to ensure accountability, and transparency in the company's relationship with its stakeholders and actors among the lifecycle projects. Effective governance is a crucial indicator of management's stability and commitment to shareholder interests.

Management quality is an asset of any company.
It influences every aspect of the organization, from strategic planning in a long-term way to day-to-day management.

For investors, a thorough assessment of management quality can provide a useful insights into the company's future prospects, making it an reliable component of a clear investment analysis.

2.5.2 Industry Trends

Industry trends give information of directions or developments within a specific market or sector that significantly influence business and investment decisions. Identifying and understanding these trends are crucial for investors that have the aim to capitalize on emerging opportunities or mitigate potential risks.
This qualitative metric is one of the most used in the real world for its variety of applications and its precision.

The Industry Trends could be customized by several attributes, here below some of them for a complete metric view.

Technological Advances:

Innovations and technological innovations can redefine industry standards, markets, and show existing products or services obsolete. Looking at technological trends is essential for anticipating switch in consumer behavior and competitive advantages.

Regulatory Changes:

Legislation and regulatory policies can dramatically change the competitive landscape of an industry, affecting costs, and market entry barriers. Being aware of the potential impact of regulatory changes helps in assessing long-term investment viability.

Consumer Behavior:

Changes in consumer preferences, purchasing willingness of consumers, and general behaviors can influence market demand and impact the strategic direction of companies within the industry.

Economic Factors:

This factor is related to Macro-economic trends, such as inflation rates, interest rates, and economic growth but also on the microeconomics and social sphere. These variables can dictate the overall health and trajectory of an industry.

Analyzing Industry Trends

Analyze of industry trends involves several approaches within different outcomes such as Market Research Reports from research firm, Trade publication an journal but also an expert opinion can used for assessing a analysis regarding the Industry Trends.

For investors, the ability to accurately identify and interpret these trends is indispensable for developing investment strategies that anticipate future market movements. By integrating industry trend analysis into their overall evaluation process, investors can increase their ability to make informed decisions, and achieve superior investment outcomes in a constantly evolving market environment, like the real world give us.

2.5.3 Corporate Culture

Corporate culture takes in the shared values, beliefs, and practices that shape the behavior and decision-making processes within an organization. It is metric that significantly influences a company's efficiency, innovation capacity, and employee satisfaction
but also, its sustainability profile.

There are different dimensions that came into play while the Corporate Culture is analyzed:

Innovation:

Cultures that encourage innovation are often more adaptable and can seize market opportunities more effectively. This aspect of culture is crucial for companies that work in rapidly evolving industries.

Integrity and ethics:

A culture situated in ethical practices of a company, or a project aim builds trust with customers, regulators, and shareholders.

Employee engagement and general satisfaction:

Companies with positive corporate cultures tend to have higher employee satisfaction, which can lead to increased productivity, decreased turnover rates, and can generate a great place to work for the top talent.

There are several ways for evaluating corporate culture involves qualitative analysis through various approaches:

It is possible through employee reviews and surveys that reach out insights from current and former employees that can provide real-world perspectives of the company's culture.
Although, another instrument could be the leadership statements that through official communications can reflect the underlying corporate culture.
At least, also a corporate social responsibility (CSR) initiative can be used to assess the company culture, which can indicate a company commitment to social and environmental issues.

Conclusion

Corporate culture is a vital aspect of a company's identity. It influences every facet of a business, from employee engagement to customer satisfaction. For investors, a deep understanding of a company's culture can provide critical insights into its potential for long-term success and its sustainability ambitious.

In general, qualitative analysis is an indispensable part of investment decision-making as stakeholders point of view. It requires an approach that considers various aspects of a business and its environment that numbers alone cannot capture entirely.

2.6 Balancing Both Perspectives

The most astute investment decisions arise from a harmonious trade-off between qualitative and quantitative analyses. Thanks to this chapter we explored how integrating both perspectives can lead to more robust, and strategic investment choices.
The balancing between these two complementary metrics is indispensable for assessing an informed and reliable investment decision among the investors and stakeholders.

2.6.1 Strategic Integration of Analyses

The integration of qualitative and quantitative analyses does not add layers of complexity to investment decision-making (maybe it is common to think that adding metrics to the analysis should be more difficult to elaborate), however it boosts the robustness of the process.
By combining financial data with soft insights, investors can form a well-rounded view of investment opportunities, understanding not just the 'what is' status of a company or a project but the 'why' and ‘which’ decision fit best in terms of returns and risks. For instance, quantitative data (for example the Net Present Value) might suggest that a project is undervalued based on its financial ratios, however qualitative analysis might reveal underlying issues like poor management or deteriorating brand value that justify the low valuation and help the investors to the investment decision.

To illustrate the power of this integrated approach, consider the case when a venture capital firm that wants invests in tech startups at an early stage. By using quantitative metrics, they discover the financial health and its growth potential, based on cash flow projections and revenue forecasts.
However, understanding only that numbers don't tell the whole story, also conduct in-depth qualitative assessments, evaluating the strength of the management team, the innovativeness of the product, and the competitiveness of the market landscape is fundamental in this early stage of business. This dual approach has enabled them to identify startups that not only have strong financial fundamentals but also the right strategic positioning for long-term success.

Another compelling example is that of an investment fund specializing in sustainable investments. The quantitative analysis used to screen investments for financial health, including profitability, debt levels, and growth rates are indispensable but not complete. Concurrently, qualitative assessments conducted to evaluate the sustainability practices of the companies, their compliance with environmental regulations, and their contributions to social and environmental causes are a must.

2.6.2 Challenges in Integration

Integrating qualitative and quantitative research can provide a comprehensive, and usually complete view of investment opportunities, however it is not without challenges. One of the main challenges is reconciling data that can sometimes point in different directions and which the merger is counterintuitive. For example, a company may have a strong but inefficient investment strategy, making the investment decision thesis less strong. Facing up this challenge requires the ability to balance various factors and know a deeper understanding of the tradeoff between quantitative performance and qualitative outcomes. [4]

2.6.3 Overcoming Integration Challenges

Investors can mitigate these challenges by developing a systematic framework that allow to customization between projects for the analysis that assigns relative weights to qualitative and quantitative factors based on their investment strategy and risk tolerance. This framework can help in making consistent decisions, even when faced with conflicting information as usually happens.
Additionally, staying informed about industry trends and market dynamics is crucial for understanding the context behind the numbers, both methods and analysis are essential for a consistent investment decision.

2.6.4 Conclusion

Balancing qualitative and quantitative perspectives is an art. It demands a deep understanding of financial metrics and market trends but also an insightful approach to evaluating company culture, management quality, and other intangible factors. By mastering this balance, investors can navigate the landscape of investment decision-making with greater confidence and achieve a higher probability of success in their investment accomplishments.
The key is in leveraging the strengths of both types of metrics while being aware of their limitations, continuously refining the framework to adapt to the change in market conditions.

3 CHAPTER

PROJECT SCHEDULING AND PRIORITY RULES

Competent project management is the cornerstone of successful project completion, lying at the heart of this discipline are the principles of project scheduling and priority rules. These concepts not only ensure that projects are completed within their specified timelines and budgets but also ensure the optimization of resource allocation and the project workflows. This chapter aims to go deep through project scheduling and the establishment of priority rules, essential components of any project manager.

Project scheduling is a critical task that involves the mapping of the project tasks, their durations, dependencies, and the resources assigned to them, in the following chapters, an assumption of these will be proposed as application to understand in the deep how relevant is the resource allocation. It serves as a blueprint for the entire project lifecycle, providing a clear timeline and sequence for all project activities. By establishing a comprehensive schedule, project managers can anticipate potential bottlenecks, balance resource loads, and ensure that critical milestones are achieved on time, having on the other hand the optimizing of the budget outcome of the entire project.

Equally important are the priority rules that guide the sequencing of project tasks. These rules determine the order in which tasks are undertaken (sequentially picked up by the project manager), based on factors such as task urgency, resource availability, and strategic importance (in some case also the financial metrics plays a factor of priority). Priority rules play a pivotal role in decision-making processes, especially in complex projects where multiple tasks compete for limited resources.

In this chapter, we will explore various types of projects scheduling methods, including traditional techniques like the Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT), as well as agile and lean scheduling practices that supply to dynamic project. We will also examine different priority rules and their application in project scheduling, highlighting how they can be effectively used to compete with the challenges of project management.

Through theoretical discussions, the chapter's aim provide readers with a deep understanding of how effective project scheduling and the application of priority rules can lead to more efficient project outcomes. Whether you're managing a small project or a large-scale initiative, the insights gained from this chapter will give you the tools and knowledge needed to excel and fit better in the project management environment. [5]

3.1 Types of Project Scheduling in Details

Project scheduling is not a one size for all the process. The diversity of projects ranging from construction and engineering to software development and event planning or in every aspect of life, necessitates a variety of scheduling methods tailored to specific project requirements, team dynamics, and stakeholder expectations. This subchapter aims to explore the project scheduling types in detail, shedding light on their methodologies, and best-fit scenarios for each, with the aim to develop critical thinking.

At the heart of project management, there are several scheduling methodologies that serve as the plan for project execution, to guiding teams through the complexities of task sequences, milestone, and resource allocations. The choice of scheduling method can significantly impact a project's efficiency, and overall success.

From traditional to contemporary approaches, looking at the global project or a single section, each scheduling type offers unique advantages and challenges, making the selection process significant to aligning project goals with execution strategies.

We will go on board a detailed examination of the most prominent scheduling techniques, including the Critical Path Method (CPM) and the Program Evaluation and Review Technique (PERT) the most used scheduling techniques. These methodologies are recognized for their precision in time management and risk assessment, providing clear pathways through the project tasks and deadlines.

In addition to these classic approaches, we will explore a more flexible and adaptive scheduling strategies such as Agile Scheduling, which solve the project issues with the evolving nature of projects through iterative cycles and continuous stakeholder feedback. This section will also cover the Lean Scheduling approach, emphasizing efficiency, waste reduction, and value maximization, as typical attributes of the Lean means.

Each scheduling type will be separated to reveal its core principles, procedural steps and tools offering a comprehensive understanding of how it can be applied to various project scenarios. Through comparative analysis, we can highlight the strengths and limitations of each method, providing project managers and teams with the insights needed to make informed decisions on the most appropriate scheduling approach for their unique project demands.

By the end of this subsection, readers will be equipped with a knowledge of the diverse landscape of project scheduling methods. This understanding will empower project manager to navigate the hardest work of project planning, ensuring that projects are delivered on time, within budget, and in alignment with strategic objectives that outcome different scenario for each type of project request. [6]

3.1.1 Critical Path Method (CPM)

The Critical Path Method (CPM) is a key component of project management, which provides a constructed approach to scheduling and time management related to a project or an investment decision.
The CPM delivers a framework for planning and executing projects, particularly beneficial in recognizing crucial time constraints and identifying the longest path (a fundamental point) to project completion. [7]

Calculation of Early Start and Finish Times, Late Start and Finish Times metrics used into the CPM model:

Using the CPM model involves the calculations of some metrics like: Early Start (ES), Early Finish (EF), Late Start (LS), and Late Finish (LF) times for each project activity, and through them it develops an approach to solve a great part of issue that could involve a project or a decision. These calculations, though apparently formulaic, give a comprehensive understanding of project timelines and dependencies.

Early Start (ES): This means the earliest possible beginning time for an activity, factoring in predecessor dependencies. It involves determining the maximum Early Start times of preceding activities and adding the activity's duration.

$$ES = max\ (\ ES_{predecessor\_activities}\ ) + Duration$$

Early Finish (EF): Reflecting the earliest completion time for an activity, EF is calculated by adding the Early Start time to the activity's duration.

$$EF = ES + Duration$$

Similarly, Late Start (LS) and Late Finish (LF) times are computed to determine the latest allowable start and finish times for activities without exposing project deadlines.

Late Start (LS): Denotes the latest time an activity can start without delaying the project completion. It is calculated by subtracting the activity duration from the Late Finish time of the activity, obviously starting with a backward approach.

$$LS = LF - Duration$$

Late Finish (LF): Represents the latest time an activity can finish without delaying the project completion. It is the minimum of Late Start times of subsequent activities.

$$LF = min(\ LS_{successor\_activities}\ )$$

Conventionally, project teams observe conventions in CPM implementation. This includes the adoption of standardized terminology as explained above and notations to ensure clear communication and consistency between project stakeholders and all the other relevant actors involved in the project. Additionally, project managers control CPM in parallel with other project management techniques, such as Work Breakdown Structures (WBS) and Resource Leveling, for provide a complete view of all project's attributes to optimize resource allocation and mitigate scheduling conflicts. [8]

Identification of the Critical Path and Analysis of Project Time Impacts:

The identification of the critical path helps all the activities developed in project planning and execution. Project managers meticulously analyze activity dependencies and durations to discern this critical path, often represented graphically through network diagrams or graphs. This visual representation facilitates communication among project teams and stakeholders, essential for achieving the project's aim, development a shared understanding of project priorities and constraints.

Moreover, the critical path analysis enables project managers to anticipate potential delays and allocate resources carefully. By identifying critical tasks and their associated risks, project teams can proactively address challenges and streamline project workflows, highlighting a critical issue or potential gain. This proactive approach minimizes disruptions and increases overall project resilience, ensuring timely project delivery even in the face of unforeseen obstacles.

In summary, the Critical Path Method excels in its role as a scheduling technique; it represents a crucial function in project management philosophy. Its systematic approach to time management and risk assessment permits project teams, and stakeholders, to navigate complexities with confidence and clarity. By embracing CPM's principles and best practices, project managers drive innovation and excellence across industries to success and to optimize the project outcomes.
In the following paragraphs we will go depth into the analysis and the comparison of the CPM and other model used in the project scheduling for the completion and the resolution of projects, helping the lectors to estimate by himself the importance of the correct use of a tailormade approach for every standalone project and not.

3.1.2 Program Evaluation and Review Technique (PERT)

The Program Evaluation and Review Technique (PERT) emerges as another instrument in the project scheduling options of project management methodologies, it offers a structured approach to planning and scheduling projects.
Like its counterpart, the Critical Path Method (CPM), PERT finds application across diverse industries and for several decision problem with different problem nature, it also offers to project managers a toolset to navigate uncertainties and workflow project workflows under critical circumstances.

In order to assess the PERT, there are a few calculations to be developed.
The calculation of Expected Time, Optimistic Time, Pessimistic Time

PERT revolves around the calculation of Expected Time (TE), Optimistic Time (TO), and Pessimistic Time (TP) for each project activity, allowing manager and stakeholders to assess and manage project risks efficiently, here below the definition of the calculation required to develop a PERT.

Expected Time (TE): This represents the best estimate of the time required to complete an activity, factoring in uncertainties and variations. It is calculated as a weighted average of Optimistic Time (TO), Most Likely Time (TM), and Pessimistic Time (TP), (all the variables will be explained below) using the formula:

$$TE = \frac{TO + 4TM + TP}{6}$$

Optimistic Time (TO): Denotes the shortest possible time in which an activity can be completed under ideal conditions, in case of no issues and no change in assumption.

Pessimistic Time (TP): Represents the longest possible time required to complete an activity, considering adverse conditions or setbacks, caused by an infinitive possible problem faced during the project completion.

Conventionally, PERT employs specialized notations and terminology to facilitate communication and collaboration among project stakeholders. Project teams meticulously take notes of project activities and dependencies, often represented graphically through PERT charts or network diagrams. These visual aids provide project managers with a holistic view of project timelines and critical paths, enabling informed decision-making and resource allocation, helpful not only to the project manager but also to all the involved stakeholders.

Probability Analysis and Risk Management:

One of PERT's distinctive features is its emphasis on probability analysis and risk management.
By incorporating probabilistic estimates of activity durations, of previous assumption by knowledge for example, project teams gain insights into project uncertainties and potential risks. Combining a Monte Carlo simulation techniques (a methodology explained in the next subparagraph) allowed project managers to develop contingency plans and mitigate risks proactively, creating a complete instrument.

Resource Optimization and Schedule Compression:

PERT also enables project managers to optimize resource allocation and schedule compression, balancing competing priorities and constraints, giving a project manager the opportunity to always look at the budgeting. Through techniques such as resource leveling and crashing, project teams can efficiently allocate resources and accelerate project timelines without compromising quality or scope. This strategic approach to resource management develops project efficiency and resilience, guaranteeing, at certain level of probability, timely project delivery in dynamic and challenging environments.

Differences between PERT and CPM:

The PERT and CPM techniques share common feature in their approach to project management, however there are some different key aspects in their methods:

Stochastic vs. Deterministic Approach: PERT embraces a stochastic approach, incorporating probabilistic estimates of activity durations, whereas CPM follows a deterministic approach, relying on fixed estimates of activity durations. It changes all the constraints at the beginning and the final output, using a different weight for every task creating a different output.

Focus on Time vs. Time-Cost Tradeoff: PERT prioritizes time management and risk analysis, whereas CPM emphasizes the tradeoff between project time and cost, allowing project managers to make informed decisions regarding resource allocation and project scheduling.
These two focuses are used in different project situations and developed looking for a different aim.

Handling Uncertainty: PERT has better behavior for projects with high levels of uncertainty and variability.
In contrast, CPM is more suitable for projects with well-defined and stable requirements, where precise scheduling and resource allocation are paramount.

Calculation of Critical Path in Uncertain Scenarios:

In scenarios of uncertainty, calculating the critical path becomes nuanced work. Project managers must consider not only the deterministic durations of activities but also the probabilistic nature of uncertainties. Techniques such as Monte Carlo simulation can be employed to simulate multiple scenarios and determine the critical path under varying conditions. This comprehensive approach allows project teams to assess project risks more effectively.

In conclusion, the Program Evaluation and Review Technique (PERT) is a concrete tool or way to approach (it depends at which prospective it is seen) of modern project management methodologies. Its systematic approach to uncertainty management and risk analysis empowers project teams to navigate complex projects with confidence and resilience. By embracing PERT's principles and best practices, project managers follow the way for successful project outcomes, driving innovation and excellence across several sectors. [5]

3.1.3 Agile Project Scheduling

Within the dynamic landscape of project management, Agile Project Scheduling was born as a transformative approach, tailored to address the evolving needs and uncertainties in the modern and complex world of projects. Thanks to the Agile methodology, Agile Project

Scheduling offers a flexible method for adaptive planning, iterative development, and continuous improvement, encouraging collaboration and responsiveness among project team's contrary to the waterfall methodology.

Calculation of Iteration Time, Velocity, and Sprint Planning:

Agile Project Scheduling revolves around the calculation of some metrics that helps managers to evolve the scope of the project in order to achieve the expected result. These metrics are traditionally the Iteration Time, Velocity, and Sprint Planning, that enabling project teams to iteratively plan and execute project activities, here below some definition of the metrics commonly used.

Iteration Time: This represents the duration of a development cycle or iteration, typically ranging from one to four weeks. It allows project teams to focus on delivering a tangible increment of work within a fixed timeframe, creating a cadence of continuous delivery and interaction among the team, collecting feedback regarding previous or future milestones.
Velocity: Denotes the amount of work completed by a team within a single iteration. It is calculated based on historical performance, reflecting the team's capacity to deliver value over time and across Iteration Time.
Sprint Planning: Involves collaborative sessions where project teams prioritize and commit to delivering specific user stories or features explaining the challenge and discovering the desirable outcome. It provides a structured framework for aligning stakeholder expectations, and refining project requirements.

Conventionally, Agile Project Scheduling relies on Agile frameworks such as Scrum or Kanban, which provide, in addition to all the figure requested, structured guidelines and ceremonies for iterative planning and execution. Most of these ceremonies are daily stand-up meetings, sprint reviews that excel the projects thanks to communication, and continuous improvement created by project teams.

Adaptive Planning and Continuous Feedback:

One of Agile Project Scheduling's features is its emphasis on adaptive planning and continuous feedback. Unlike traditional project management approaches, commonly known as waterfall approach, which rely on rigid plans and extensive documentation, Agile supports change and uncertainty as natural aspects of project development.

Through iterative cycles of planning, execution, and review, Agile Project Scheduling enables project teams to respond rapidly to changing requirements, market conditions, and stakeholder requirements. This iterative approach allows projects to evolve organically and deliver maximum value to stakeholders.

Dynamic Resource Allocation and Team Empowerment:

Agile Project Scheduling also underlines dynamic resource allocation, leveraging cross-functional teams and self-organizing principles to optimize project outcomes. Instead of assigning tasks based on predefined roles or hierarchies among the team, Agile encourages collaboration, autonomy, and collective ownership.

By empowering teams to make informed decisions and adapt to changing priorities, Agile Project Scheduling enhances creativity, and morale within project teams. This distributed leadership model raises a culture of responsibility and innovation, driving continuous improvement allowing all the members of the team to growth together.

Principal differences between Agile Project Scheduling and Traditional Approaches:

While Agile Project Scheduling shares common goals with traditional project management approaches, it differs in some key aspects:

Flexibility vs. Predictability: Agile Project Scheduling prioritizes flexibility and responsiveness, allowing project teams to adapt to changing requirements and market dynamics. On the other hand, traditional approaches emphasize predictability and observance to predefined plans and schedules defining them before and not continuously.
Customer Collaboration vs. Contract Negotiation: Agile Project Scheduling emphasizes close collaboration with customers and stakeholders, incorporating their feedback and insights throughout the project lifecycle to assure a commonly defined outcome. Traditional approaches, on the other hand, rely on formal contracts and negotiations to define project scope and deliverables, without the possibilities of renewable constraint and aim.
Iterative Development vs. Waterfall Methodology: Agile Project Scheduling embraces iterative development cycles, where requirements evolve incrementally based on customer feedback and changing priorities. Traditional approaches, such as the Waterfall methodology, follow a sequential process with distinct phases for planning, development, testing, and deployment, all of them creating separately with a lower grade of communication.

In conclusion, Agile Project Scheduling represents a differ way to see the world around in project management philosophy, empowering project teams to embrace change, uncertainty, and complexity with confidence and support in every phase. By adopting Agile principles and practices, project managers unlock new possibilities for innovation, collaboration, and value delivery, driving success in today's unpredictable business environment which is hard to manage.

3.1.4 Resource-Constrained Scheduling

Resource-Constrained Scheduling (RCS) develops as a critical aspect of project management, particularly in environments where resources are finite and demand exceeds supply, this is the main characteristic that define the use of RCS instead of other methodologies. This approach prioritizes the efficient allocation and utilization of resources to optimize project outcomes while minimizing bottlenecks and resource struggles in the project constraints and among the actors created by the project team.

Calculation of Resource Availability, Task Dependencies, and Resource Levelling, used in the Resource-Constrained Scheduling (RCS):

Resource-Constrained Scheduling involves a calculation of some metrics like resource availability, and task dependencies to ensure a helpful resource allocation and utilization.

Resource Availability: Project managers evaluate the availability of resources, including personnel, equipment, and materials, considering factors such as skill sets, availability windows in a time measure, and constraints.
Task Dependencies: Project tasks are analyzed for dependencies, including sequential, parallel, and overlapping relationships, to identify critical paths and resource constraints, in the following chapter we will go deeper into that analysis with example and also some application.

Conventionally, Resource-Constrained Scheduling relies on specialized software tools and algorithms to analyze resource availability and constraints, identify critical paths, and develop resource allocation policies. Project teams collaborate closely with stakeholders to align resource allocation decisions with project goals, priorities, and purpose.

Dynamic Resource Allocation and Capacity Planning:

One of the key features of Resource-Constrained Scheduling is its emphasis on dynamic resource allocation and capacity planning. Project managers continuously monitor resource utilization and adjust resource allocation strategies in response to changing project is more requirements, priorities, and constraints. This way to challenging the project is easier with using a Agile methodology.

Through proactive capacity planning and model created with several tools, project teams anticipate resource demands and constraints, identifying potential bottlenecks and resource shortages before they impact project timelines and deliverables, a key aspect for this approach. This enables project managers to make informed decisions regarding resource allocation, outsourcing, and contracting to mitigate, and transfer, when possible, risks and ensure project success.

Tradeoff Analysis and Decision-Making:

Resource-Constrained Scheduling also includes a tradeoff analysis and decision-making to balance competing priorities and constraints. Project managers evaluate alternative resource allocation scenarios, considering factors such as deadlines of the tasks, budget constraints, resource availability, and stakeholder priorities, looking at the same time at two different but relevant metrics the economic impact and the temporary.

By performing sensitivity analysis and what-if scenarios, project teams manage the impact of resource constraints on project schedules, costs, and deliverables, enabling informed decision-making and risk management proposal. This iterative process allows project managers to optimize resource allocation strategies to maximize project efficiency and achieve project objectives with at higher level of satisfaction. Sensitivity analysis and what-if scenarios are a helpful way to evaluate an opportunity.

Differences between Resource-Constrained Scheduling and Traditional Approaches:

Resource-Constrained Scheduling differs from traditional project scheduling approaches in several key aspects:

Resource Optimization vs. Task Sequencing: Resource-Constrained Scheduling prioritizes resource optimization and allocation, whereas traditional approaches focus primarily on task sequencing and scheduling without considering resource and any other types of constraints.
Dynamic Resource Allocation vs. Fixed Resource Assignments: Resource-Constrained Scheduling enables dynamic resource allocation and adjustment solved day by day going forward with the project plan (similar to the Agile methodology), whereas traditional approaches often rely on fixed resource assignments and schedules.
Capacity Planning vs. Time Management: Resource-Constrained Scheduling integrates capacity planning and resource management into project scheduling, whereas the traditional approaches focus primarily on time management and task scheduling.
In conclusion, Resource-Constrained Scheduling represents a strategic approach to project management, given the opportunity at the project teams to optimize resource allocation, mitigate risks, and achieve project objectives in resource-constrained environments. By using Resource-Constrained Scheduling techniques and best practices, project managers can increase project efficiency, minimize resource conflicts, and deliver successful outcomes in dynamic and challenging projects, having the possibility to look at the future and possible challenge. [5]

3.2 Information Regarding the Activity Attributes

In the intricate world of project management, understanding the attributes of project activities is paramount to effective planning, execution, and control. These attributes are used as the building blocks upon which project schedules are constructed, providing vital information that directs decision-making, resource allocation, and risk management throughout the project lifecycle.

Introduction to Activity Attributes:

Activity attributes cover a diverse range of characteristics and parameters associated with each project task of a different nature. These attributes used to define, categorize, and contextualize activities within the project framework, enabling project managers to gain insights into their scope, dependencies, constraints, and criticality for achieving the aim of the project in an easier way.

The activity attributes serve a wealth of information that empowers project teams to navigate the complexities of project management with precision and prevision. From task durations and dependencies to resource requirements and milestones, these attributes form the structure upon project schedules are constructed, providing a roadmap for project success.

Understanding Activity Attributes:

Activity attributes can be classified into various categories, each providing unique insights into different facets of project activities.
In this chapter we will see a various number of attributes used nowadays in the real and theoretical world.
Although, in the further chapters an assumption and a proposal of attribute will made, ordinary and not, in the purpose to discover the optimal scenario of resource allocation and decision-making preference:

Duration and Timing Attributes: Attributes such as estimated duration, start and end dates, and scheduling constraints provide crucial information about the temporal aspects of activities, enabling project managers to sequence tasks and allocate resources effectively by their duration and completion time.

Dependencies and Relationships: Attributes related to task dependencies, precedence relationships, and critical paths looking on the interconnections between activities.
An important attribute of this kind used in the next chapters will be the parallelization dependence between tasks.
This type of attributes guiding project teams in identifying and managing project dependencies.

Resource Requirements: Attributes pertaining to resource allocations, skill sets, and availability inform resource planning and allocation decisions, ensuring that project teams have the necessary resources to complete tasks efficiently, that type of attribute could include the material resource, the staff resourced, site resources and all the other type of resource that as to be allocated to a relative task.

Risk and Uncertainty: Attributes related to task uncertainties, risks, and mitigation strategies enable project managers to proactively identify and address potential threats to project aim, improving project adaptability.

Progress and Milestones: Attributes indicating task status, completion percentages, and milestone achievements provide real-time insights into project progress without having the complete view of the project under control but just looking at track performance and make timely adjustment if requested.

By going into the details of activity attributes, project managers gain a comprehensive understanding of project tasks and their underlying dynamics. Thanks to this knowledge, project teams can develop robust schedules, allocate resources in a intelligent way, and navigate project complexities with agility. In the subsequent sections, we will explore various aspects of activity attributes in detail, lighting in on their significance and practical implications in project management.

3.2.1 Task Priority Assignment

In the changeable landscape of project management, the allocation of priorities to tasks stands as a crucial aspect in the effective execution of project activities. Task priority assignment involves the systematic evaluation of tasks based on their relative importance, urgency, dependencies, and strategic alignment with project objectives. This sub-chapter research into the intricacies of task priority assignment. [9]

Introduction to Task Priority Assignment:

Task priority assignments define the strategic allocation of resources, time, and attention to tasks based on their criticality and impact on project development. By assigning priorities to tasks, project managers can ensure that resources are directed concerning activities that contribute most significantly to project objectives and deliverables. This proactive approach enables project teams to focus their efforts on high-value tasks, mitigate risks, and optimize project outcomes.

The allocation of priorities to tasks holds immense significance in project management for several reasons, from “resource optimization” enables project manager to allocate resources following a path and using a logic behind that, to “time management” enables teams to focus on the activities and the milestone using a pattern or for example for the “risk management” focusing the team on impact and minimizing the possible delay.

Task priority assignment can be approached using tailored methodologies, each developed to the specific needs of the project:

Urgency or Importance: This methodology classifies tasks based on their urgency In time and importance in comparison to other task related to resources and other factor.

Critical Path Analysis: Critical path analysis identifies the sequence of tasks that determine the overall duration of the project as the CPM does, allowing project managers to prioritize tasks allocated in the critical path to ensure project deadlines are following with the minimum distortion.

Value-Based Prioritization: This methodology in specific, categorizes tasks based on their contribution to project objectives in terms of work achieved. It is different from the previous one methodology that looks only at the resources point of view.

Risk-Based Prioritization: Task priority assignment permit to prioritized tasks that mitigate the highest project risks or dependencies (where no parallelization constraint are defined most of the time) and also have the bigger economic impact (as we will see in the next chapter).

Using a task priority assignment as general rule to define all the activities has some implications in the project management.
First of all, improved the resource allocation, ensuring that resources are allocated to activities that deliver the most value to the project, so create more valuable outcomes.
It also enables project teams to focus their efforts on high-priority tasks (for example prioritizing on the Critical Path), make sure that project milestones are met efficiently.
It implements and raises stakeholder satisfaction as a consequence.
Adopting systematic approaches and methodologies for task priority assignment, managers can optimize project or Investment decision and drive to success the team in today's challenging environment.

3.2.2 Monitoring Activity Attributes

In the world of project management, monitoring and updating activity attributes play a fundamental role in enabling project success and adaptability. This chapter explores the importance of actively tracking and revising activity attributes during the project lifecycle.

Monitoring and updating activity attributes involve the ability to track and choose key characteristics and parameters associated to the project activities. This iterative process enables project managers to maintain alignment with project objectives, respond to changing circumstances, and optimize project performance, being informed and had the control to the full journey.

The monitoring of activity attributes holds significant importance in project management some of them are listed below:

Real-Time Visibility: By monitoring activity, project managers gain real-time visibility into project progress, enabling corrections as needed and where occurs.

Risk Management: Regular updates to activity attributes permit project managers to identify emerging risks and dependencies and manage all the possible critical activities.

Stakeholder Communication: Updated activity attributes facilitate transparent communication with stakeholders, providing insights into project status, and milestones.

Several methodologies and approaches can be used for effective monitoring and updating of activity attributes following different main approaches:

For example, the Regular Reviews and Status Meetings allow regular reviews and status meetings between project teams to assess progress, clarify issues, and update activity attributes as needed.
In alternative an Automated Tracking Systems came to help the project management for automated tracking systems, in particular when real-time monitoring of activity attributes is needed.
Approaching the Change Control Processes instead ensures that updates to activity attributes are managed systematically, minimizing disruptions and maintaining project integrity, it become more easily within a project based on horizontal capabilities, where the change in control are done without issues on knowledge.
Another basic rule used in the most of project is the use of Performance Metrics or KPIs that provides a quantitative basis for monitoring project progress and updating activity attributes accordingly with customization in advance and a clear updating when required.

Practical Implications of Monitoring and Updating Activity Attributes:

Effectively monitoring and updating activity attributes have several practical implications that simplify the life of project management in general.

It improved Decision-Making with timely updates to activity, mitigate the risk on every single activity or in general of the project thanks facilitating the identification of risks and dependencies.
The monitoring of the different also creates adaptive planning creating a schedule in response to the changing requirements and constraints.
In conclusion, monitoring and updating activity attributes are essential components of effective project management, enabling project teams to maintain visibility, manage risks, and adapt to changing circumstances. By asses a regular review of every task with a systematic approach, project managers can guarantee that activity attributes remain accurate, relevant, and aligned with project milestones throughout the project lifecycle.

Advanced Scheduling Techniques

3.2.3 SPT (Shortest Process Time)

In a project scheduling environment, the Shortest Processing Time (SPT) algorithm stands out as one of the tools used for optimizing task sequencing and utilization of all the resources use for achieved project goals. This paragraph goes deep into the principles, the utilization and through the benefits of the SPT algorithm in project management, learning its role in improving efficiency and minimizing project lead times and failure.

The Shortest Processing Time (SPT) algorithm is a scheduling technique that set priority tasks based on their processing times (also known as the duration of the activities), with shorter tasks receiving higher priority over longer ones, without considering an earlier prioritizing.
This algorithm aims to minimize the average completion time of tasks project and optimize resource utilization by focusing on quick wins and reducing task lead times. [8]

The SPT algorithm is composed of different components, here below are a few of them for a clear view and a better understanding of the following chapters.

In the SPT algorithm, the task processing time plays a fundamental role, obviously because the algorithm prioritizes tasks based on their processing time.
The task queue is made based on the processing time of every singular task and tasks are organized into a queue or list, the position is determined by its processing time according to the SPT algorithm.
Furthermore, the resource allocation allocates resources based on the scheduled tasks, ensuring that shorter tasks are completed promptly to maximize resource utilization.

The SPT algorithm finds numerous applications in project management, including:
Job Shop Scheduling, prevalent used in manufacturing environments, the SPT algorithm is used to sequence production jobs on machines, minimizing setup times and maximizing throughput of the chain.
Task Sequencing that applied SPT algorithm to prioritize project tasks, ensuring that shorter tasks are completed first to reduce project lead times.

The SPT algorithm, at who utilized them, offers several benefits.
It reduced the Lead Times, by prioritizing shorter tasks, allowing projects to be completed more quickly and efficiently in every task.
It also improved the Resource Utilization by focusing on tasks with shorter processing times, this way to schedule, maximizing resource utilization and efficiency.
It also affects the global flexibility of the project from task sequencing to adapting through the amendments of the project requirements and priorities.

In conclusion, the Shortest Processing Time (SPT) algorithm is a valuable tool in project scheduling, offering several benefits.
By understanding and applying the principles of the SPT algorithm, project managers can optimize resource allocation, transforming all the project efficiency, and conducting to more successful project outcomes.

3.2.4 LPT (Longest Processing Time)

In the domain of project scheduling, the Longest Processing Time (LPT) algorithm develop a strategic approach for task prioritization and resource optimization, it has a similar aim of SPT algorithm, using the same variables but pursuing an opposite outcome. This subparagraph goes deep into the principles, advantages, and the comparison of the LPT algorithm in project management, showing why resource utilization and project efficiency are the core interests. [8]

Introduction to the Longest Processing Time (LPT) Algorithm:

The Longest Processing Time (LPT) algorithm is a scheduling technique that shows the other face of the medal with respect to the SPT algorithm, prioritizes tasks based on their processing times, with longer tasks receiving higher priority over shorter ones. This algorithm aims to optimize resource utilization and project efficiency by tackling larger tasks early in the schedule, thus reducing their impact on project lead times, not looking at quick wins but looking at variability lead time.

The key components of the LPT Algorithm are pretty the same considered in the SPT algorithm.
As explained before the core components are the task processing time, considered as reversal approach of SPT algorithm, the task queue that remain the same, and the resource allocation, allocating dynamically the resource based on the scheduled task and looking at the longer assuring an adequate resource to expedite their completion.

The LPT algorithm finds various applications in project management in the same way that SPT algorithm does.

From Project Planning, prioritizing tasks, focusing on larger tasks first to minimize their impact on the timelines, touching Resource Allocation ensuring the correct resources allocation to expedite the completion of the project or the stand-alone task, and going till the Task Sequencing in a manner to maximizes resource efficiency and minimizes project lead times.

The LPT algorithm offers several benefits for project management, also like the SPT algorithm for the optimization of the Resource Utilization and reducing the Project lead Time, but with an exception for what concern the Project Efficiency, because using the LPT algorithm helps minimize project lead times by addressing larger tasks early in the schedule for their implicit variability.

In conclusion, the Longest Processing Time (LPT) algorithm serves as a fundamental tool in project scheduling,
The LPT and the SPT algorithm are the principal methods used to maximize efficiency, minimize resource utilization and so the waste of time and money offering other several benefits.

By leveraging the principles of the two-algorithm discussed in these 2 subchapter, project managers can enhance resource allocation, optimize task sequencing, and achieve more successful project outcomes.

3.2.5 Monte Carlo Simulation

Monte Carlo Simulation is a one of the powerful probabilistic technique available in sciences, it finds extensive application in project scheduling for risk analysis and determining completion probabilities of each activity and for the entire project when the probability of each task is known. This chapter aims to explore this methodology, his implementation, in project management. Looking at its role in risk assessment and decision-making processes.

Monte Carlo Simulation is a computational technique that creates random sampling to analyze the possible impact of uncertainty and variability in project schedules with a certain confidence. It develops a method to assess project computation, thanks to the statistics, by generating many simulations based on probability distributions of project parameters, for example using the task durations and the resource availability, Monte Carlo Simulation provides valuable insights into project risks and uncertainties, useful for the project team and secondly for all stakeholders.

The application of the Monte Carlo Simulation finds widespread application in several activities and for various purposes, regarding the project scheduling instead the application is several and some of them are defined here below.

The Risk Analysis: Monte Carlo Simulation enables project managers to assess all the challenges that the project deadlines and the milestone that faced up before or during the completion of all activities

The Resource Allocation: incorporating resource constraints and uncertainties into simulations, allocating them into all activities attributes, Monte Carlo Simulation helps to optimize resource allocation and so with the mitigation of the risk associated them.

The Schedule Optimization: Monte Carlo Simulation facilitates the identification of critical paths, bottlenecks, and delay in scheduling by simulating various project scenarios defining the subsequence as a sensitivity analysis.

Methodology of Monte Carlo Simulation:
Assess a Monte Carlo Simulation involves into the methodology various steps.
The methodology of Monte Carlo Simulation involves the following steps:

First, the team must define the Probability Distributions, such as task durations, resource requirements, and cost estimates, to address at every objective of the project and characterized by probability distributions based on historical data or a judgment of an expert.
After the definition of the Probability Distribution, The Monte Carlo Simulation allow to generate a Simulations, so, a large number of simulations are generated by randomly sampling values from the defined distributions, this is a fundamental step of the simulation representing different scenarios of project execution.
Then, the ultimate step of the Monte Carlo Simulation is analyzing the results to assessing project risks, and determine the probability of meeting project milestones, such as completion deadlines and budget targets.

Monte Carlo Simulation offers several benefits for project management, from Enhanced Risk Assessment, by quantifying uncertainties and variability in project parameters,
to make informed decisions by evaluating the likelihood of different project scenarios and identifying strategies to mitigate risks and uncertainties among all the projects taken into consideration for the general scope.
It also improves the Planning Schedule, helping to develop more robust and resilient project plans that can adapt to changing conditions and uncertainties, as the world today offers us.

In conclusion, Monte Carlo Simulation is a decisive tool in project management for risk analysis and decision-making. Using leveraging probabilistic techniques to simulate various project scenarios, Monte Carlo Simulation manages risk assessment, improves planning, and ultimately contributes to more successful project outcomes.

4 CHAPTER

ANALYSIS OF ALGORITHM

4.1 Introduction

In this chapter, we will research into the analysis of the activity scheduling algorithm. The primary goal is to provide a comprehensive explanation of the attributes involved, the rules that manage the scheduling (looking at the assumption made) the algorithm itself, and the underlying principles guiding its functionality.

The algorithm's purpose is to determinate the convenience of using custom algorithm created for the occurrence instead to the usual one used nowadays, balancing multiple criteria such as duration, cost, and value and examining financial aspects like NPV, IRR and other composite metrics.

The objectives of this chapter are to explain the main part of the Analysis through the algorithm and the drivers of picking activities and describe the methodology process used to generate data, define an evaluation criterion used to evaluate the performance of algorithms themselves and show the output obtained from algorithms scenarios.

4.1.1 Overview of the Developed Algorithm

4.1.1.1 Explanation of Attributes

In this section, we will detail the key attributes used form as the main attributes of algorithm choices of the activity scheduling. These attributes define the characteristics of each activity and will determine how activities are scheduled inside the project. How explained in previous chapters, especially in the "2nd Chapter" the understanding of these attributes is crucial for comprehending the overall functioning of the algorithm and the logic that move the scheduling decisions.

Here below all the subparagraphs explaining each of them all the core attributes for the financial to the scheduling point of view.

4.1.1.2 Duration

Each activity has a duration that defines the time required to complete the activity itself. The duration of the activity is not defined at the beginning of the attributes choice but once the scheduling has been defined and the Start Date and the End Date are established.

Inside the project, the duration is one of critical factors in scheduling because it impacts the project timeline.
Thanks to that purpose, shorter duration allows a quicker project completion, instead of longer duration that require more time and more delay subsequent activities looking at the probability itself.

During the developing of the algorithm several ranges of durations have been for created a various and heterogeneous dataset looking at numerous combinations, the duration range (within a minimum days of activity duration to a maximum) selected for the ultimate data frame are the following explained in the *"Table 1: Duration range"* below

	Minimum Days activity [days]	Maximum days of activity [days]
1st Duration range	1	5
2nd Duration range	1	10
3rd Duration range	3	20

Table 1: Duration range

This set of duration ranges has created the best fit for our heterogeneous amount of data.

Furthermore, the durations have been used for optimizing the algorithm sequence and the timing of every algorithm in discussion, ensuring that the project is completed as efficiently as possible.

4.1.1.3 Value & Cost

The other two attributes used for the development of the scheduling algorithms are value and cost. We are going to see both together because are related to the same financial activity and collected starting from the same thought.
The value of an activity could be resume as the economic value generated by completing an activity.
On the other side, the cost of an activity could be seen in different ways, however his definition could be the expense incurred to perform the activity.
The cost of activities can be made of several links to project expenses like Human cost, Material cost, time cost and time cost.
In our discussion this cost is comprehensive of all of the cost explained above and in the "2nd Chapter".

When the value represents the financial benefit or the return that an activity brings to stakeholders, the cost affects in the same way the overall budget of the project but within and an opposite sign.
The attributes value and cost in our analysis are not used only for evaluate the project scheduling overall (we will see in the next section) but also for prioritize activities, influencing decision in custom scheduling algorithms that weigh value and cost against other factors, and so make the difference in cost optimization.

In our creation of these attributes, we are taking into consideration not only the "gross margin" that a task can bring (Value – Cost), usually the percentage part of return considering only value and cost, but also a random value starting from a minimum gross margin to a cap value (a maximum), instead the cost must be affected to a bottom minimum value.

4.1.1.4 Predecessors

Each activity could have a predecessor or successor that defines the link to other activities and the right sort of activities completion. In the case of predecessor means that other activities must be completed before the given activity can start.

The predecessors of the activity are defined at the beginning of the attribute's choice.

Inside the project, the predecessor is one of critical factors that define the relationship between activities and the dependency structure, anyway, it is relevant in scheduling because it impacts the project timeline, the activities sorting and as consequence all the metrics related.
Each activity can have the constraint to be done before or after a single or multiple tasks.

In our cases, likewise the common project for each activity has been establish a range number of predecessors have been defined.

During the developing of the algorithm different ranges of number of predecessors have been for created a various and heterogeneous dataset looking at numerous combinations, the predecessor range (within a minimum activity to a maximum) selected for the ultimate data frame are the following explained in the *"Table 2: Predecessors range"* below:

	Minimum Predecessor [task]	Maximum Predecessors [task]
1st Predecessors range	0	1
2nd Predecessors range	0	2
3rd Predecessors range	0	4
4th Predecessors range	0	20

Table 2: Predecessors range

4.1.1.5 Start Date and End Date

The Start Date and End Date play an important role for the definition of scheduling and the creation of the Duration in general that impacts different evaluation indicators.

These attributes are created starting from an algorithm scheduling and defined only after the scheduling order is established.

In our case the Start Date and the End Date are taken only looking at the date itself and not using an hour format.
Furthermore, these two attributes do not follow a meticulous schema but take after a general rule explained before that take into consideration the duration of the activities itself.

4.1.2 Discount factor

The Discount factor refers to the rate used for evaluating forma a financial prospective a task or the project in general.

The algorithm, therefore, thanks to this attribute, can develop the quantitative financial metrics explained in the next chapter.

This, however, is a fundamental aspect for the evaluation of activities.
For taken the most realistic discount factor into consideration due to a low time duration the level of deposit cost and the most used discount factor in project which the overall duration is less than 1 or 2 years have been considered.

Various values of the discount factors have been considered as shown below in the "*Table 3: Discount factors"* table:

	Discount factor
1st Discount factor	3 %

2nd Discount factor	4,5 %
3rd Discount factor	6 %
4th Discount factor	7,5 %

Table 3: Discount factors

4.2 Activity Generation Process

The activity generation process is a critical component of the scheduling algorithm, responsible for creating a diverse set used for subsequent scheduling and analysis. The aim of this process is to generate realistic activities, varied, and reflective of actual project scenarios. In this section, we will see the key aspects of the activity generation process, including seed usage and attribute distribution.

4.2.1 Seed Usage

During the generation of activities among projects a seed has been used.
The seed ensures that the randomization process can be reproduced during the changing of the constraints, that helps us to compare the analysis of single project in different runs.

For our purpose, 42 seeds (searching for the answer of everything) have been implemented before the generation of all attributes related to every runs. [10]

This way of generating attributes allows for controlled experimentation and repeatability in the analysis themselves.

The activity generation process is foundational to the scheduling algorithm, furthermore the process ensures that the generated activities are suitable for testing the effectiveness and efficiency of various scheduling approaches.

4.2.2 Assigning Predecessors

During the creation of the scheduling and only after generating the various activities, attention was paid to a crucial constraint that goes to change not only the activity itself but also has impacts in the total scheduling of the project.

4.2.2.1 Predecessor Assignment

As explained after the creation of the tasks a finite set of predecessors has been assigned to them, randomly starting only from the number of tasks that the project itself contains.
The use of predecessors makes it possible to make scheduling as similar as possible to reality, and on this basis more instances have been created to quantify the impact between them.

4.2.2.2 Cycle Avoidance

During the allocation of the predecessors, a very important factor that would have led to the project not being successful, namely the presence of cycles between groups of activities, was considered and eliminated.

This feature of the algorithm not only allowed us to get even closer to reality, being one of the most realistic constraints.
The construction of the set of predecessors for each activity has therefore taken into consideration as well as classical basic factors such as: minimum and maximum number of predecessors that have set limits in the allowed predecessors but also to avoid the presence of cycles.

Options of sets of predecessors were then developed as shown in the *"Table 2: Predecessors range"* table above.

4.2.3 Drivers of Scheduling

4.3 Resource Constraints

Up to now we have talked about constraints that allow the most likely representation of reality.
A further constraint has been created to better represent the project within the time axis in the form of Gantt, or the constraint on the availability of resources.

During the execution of the activities, the insertion of predecessors has the task of defining when and how an activity must be finished first, usually refers to a structural need, of similar skills or in the assembly or cascade components.

There are also other constraints that can arise due to lack of resources such as personnel, machinery or space.
This design constraint, which once again fully reflects reality, has been defined and replaced using parallelization constraints between activities.

4.3.1.1 Parallel Tasks

The parallelization limits refer to the non-performance of several activities carried out at the same time (simultaneously).

The algorithm, therefore, thanks to special functions, limits the parallelization of several activities at the same time checking at the time of scheduling the number of activities that are carried out for each date.
This way of working has a high impact on the scheduling itself by changing not only the total time of completion of the project but also all the metrics that derive from it.

This, however, is one of the fundamental aspects of the planning of activities that cross parallelization makes it as realistic as possible

Various values of the maximum number of parallelizable activities have been taken into account as shown below in the *"Table 4: Maximum parallel activities"* table:

	Maximum Activity in parallel [task]
1st Assumption	1
2nd Assumption	2
3rd Assumption	3
4th Assumption	6

Table 4: Maximum parallel activities

After several analyses, most of the time was spent on a constraint of parallelization of tasks with three maximum activities in parallel.

Below in the Figure *"Figure 5: Gantt Chart for Gantt Chart with maximum three parallel task*" we wanted to give a preview of the creations of the tasks according to a random order throughout a Gantt chart and considering only the design constraints explained so far.

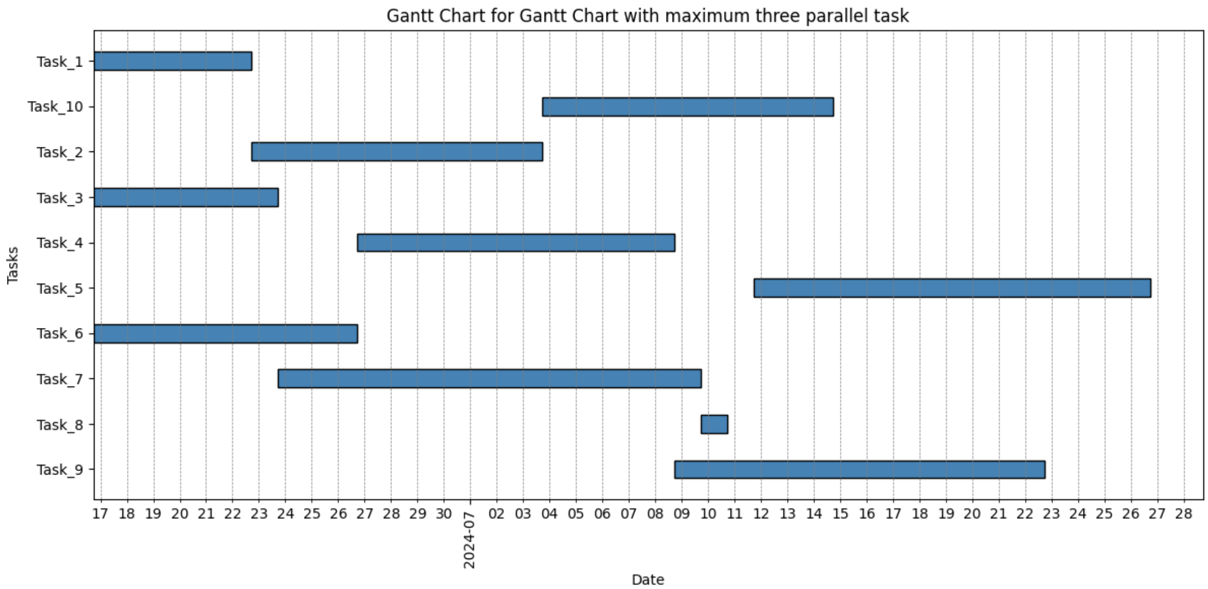

Figure 5: Gantt Chart for Gantt Chart with maximum three parallel task

4.3.2 Scheduling Strategies

The scheduling strategies implemented in the algorithm are designed to optimize the project schedule by considering different criteria and constraints. This section provides a quick explanation of the Shortest Processing Time (SPT) and Longest Processing Time (LPT) algorithms, used as benchmark and in comparison, as well as the custom algorithms developed in this research.

4.3.2.1 Shortest Processing Time (SPT)

The SPT algorithm prioritizes tasks with the shortest duration, allowing them to be completed first to reduce the average time to complete tasks as explained in the "2nd Chapter".

This scheduling process is developed and executed always keeping in mind all project constraints, resources and not defined previously.

Below, in the figure *"Figure 6: Gantt Chart for SPT Algorithm"* an example of the SPT algorithm as an example created using the algorithm, using also the predecessor's constraint with a maximum of one predecessor.

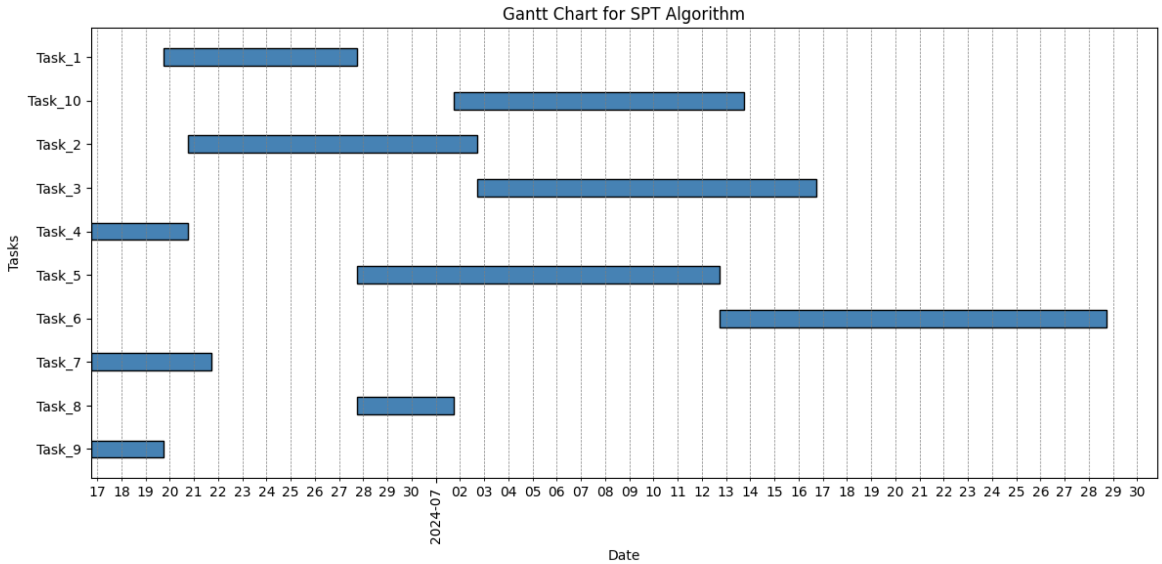

Figure 6: Gantt Chart for SPT Algorithm

4.3.2.2 Longest Processing Time (LPT)

Unlike the SPT algorithm, the LPT algorithm instead considers the opposite scenario, first completing the tasks with longer duration and then considering the others.

This scheduling process is developed and executed always keeping in mind all project constraints, resources and not defined previously.

Below, in the figure *"Figure 6: Gantt Chart for LPT Algorithm"* an example of the algorithm LPT as an example created using the algorithm, using also the predecessor's constraint with a maximum of one predecessor.

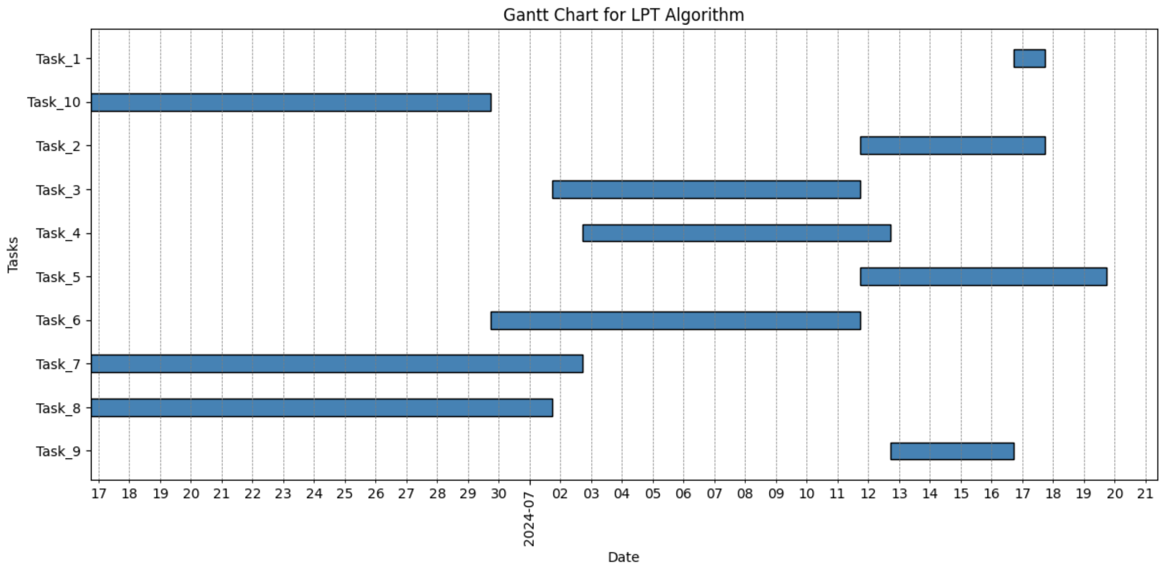

Figure 7: Gantt Chart for LPT Algorithm

4.3.2.1 Custom Algorithms

Custom algorithms, explained in the next paragraph, are proposals for scheduling developed within the algorithm itself, which take inspiration from our benchmarks such as SPT and LPT, adding to the classic drivers of selection of economic variable activities creating in turn new drivers that will be analyzed in the next chapter.

In our analysis, six drivers were created that contributed to six scheduling algorithms.
Thanks to the drivers of both temporal and economic-financial nature it has been possible not only to evaluate the scheduling of the project from the point of view of completion of activities and total duration of the project but also through qualitative financial metrics such as: Net Present Value (NPV), cost and Internal Rate of Return (IRR) that have been explained in the chapter. "1st Chapter"

Each custom algorithm is designed to address specific project needs, providing adaptability to different project scenarios due to easy conversion.

The Entire analysis will be explained in the next chapter with the consideration of the variables in assumption. [11]

4.3.2.2 Custom Algorithms definition

This chapter defines the drivers with which Custom algorithms were created.
The implementations and their formulation will be clarified, thus giving a more in-depth vision of the objective to be pursued.

Six different drivers have been taken together by the attributes that characterize the activities but each with a different formulation.

4.3.2.2.1 Algorithm Custom 1

The *"Custom 1"* algorithm choice driver has been designed to balance not only the value component but also the relative weight of durability.

A weight was then assigned through a vector 'ω' of the weights characterized as described below that considered both the duration of the single activity and its return (value - cost).

The *"Custom 1"* driver therefore has the characteristics to be used in a context where the economic component and the time component both have value.

Then, the *"Custom 1"* driver used for the *"Custom 1"* algorithm has been implemented as follows:

$$Custom\ 1 = \omega_1 \cdot Duration + \omega_2 \cdot (Value - Cost)$$

Where ω is defined as:

$$\omega = \{ \omega_1 , \omega_2 \}\ with\ \omega_1 = 0{,}3\ and\ \omega_2 = 0{,}7$$

The driver was taken into consideration starting from the maximum value to the minimum value, although the main constraint as parallelization and predecessor rules has been always taken into account.

The result of the *"Custom 1"* algorithm and driver, used for the analysis of project scheduling, will be discussed in the following chapter.

4.3.2.2.2 Custom Algorithm 2

The *"Custom 2"* algorithm choice driver has been designed as *"Custom 1"* with the aim to balance not only the value component but also the relative weight of durability, but in the opposite Duration side, so it favors activities that have shorter durations and higher financial returns.

A weight was then assigned through a vector 'ω' of the weights characterized as described below that considered both the inverse of duration of the single activity and its return (value - cost).

The *"Custom 2"* driver therefore has the characteristics to be used in a context where the economic component and the time component both have value.

Then, the *"Custom 2"* driver used for the *"Custom 2"* algorithm has been implemented as follows:

$$Custom\ 2 = \ \omega_1 \cdot \frac{1}{Duration} + \ \omega_2 \cdot (\ Value - Cost\ )$$

Where ω is defined as *"Custom 1"* as follow:

$$\omega = \{\ \omega_1\ , \omega_2\ \}\ \ with\ \ \omega_1 = 0{,}3\ and\ \omega_2 = 0{,}7$$

The driver was taken into consideration starting from the maximum value to the minimum value, although the main constraint as parallelization and predecessor rules has been always taken into account.

The result of the *"Custom 2"* algorithm and driver, used for the analysis of project scheduling, will be discussed in the following chapter.

4.3.2.2.3 Custom Algorithm 3

The *"Custom 3"* algorithm choice driver has been designed to balance the value-duration ratio.

The *"Custom 3"* driver therefore has the characteristics of being used in a context where every single day has importance in the total return of the project and it is suitable for time-sensitive projects.
This approach aims to maximize efficiency by prioritizing activities that provide the greatest economic return in the shortest amount of time.

Then, the *"Custom 3"* driver used for the *"Custom 3"* algorithm has been implemented as follows:

$$Custom\ 3 = \frac{Value}{Duration}$$

The driver was taken into consideration starting from the maximum value to the minimum value, although the main constraint as parallelization and predecessor rules has been always taken into account.

The result of the *"Custom 3"* algorithm and driver, used for the analysis of project scheduling, will be discussed in the following chapter.

4.3.2.2.4 Custom Algorithm 4

The *"Custom 4"* algorithm choice driver has been designed to balance the value-cost ratio.

The *"Custom 4"* driver therefore has the characteristics of being used in a context where every single job has importance in the total return of the project and it is suitable for economic-sensitive projects.
This approach aims to maximize efficiency by prioritizing activities that provide the greatest economic return.

Then, the *"Custom 4"* driver used for the *"Custom 4"* algorithm has been implemented as follows:

$$Custom\ 4 = \frac{Value}{Cost}$$

The driver was taken into consideration starting from the maximum value to the minimum value, although the main constraint as parallelization and predecessor rules has been always considered.

The result of the *"Custom 4"* algorithm and driver, used for the analysis of project scheduling, will be discussed in the following chapter.

4.3.2.2.5 Custom Algorithm 5

The *"Custom 5"* algorithm choice driver has been designed to balance the cost of every activity, in overall the amount of money into expenses.

The *"Custom 5"* driver therefore has the characteristics of being used in a context where every single job has importance in the negative cashflow of the project and it is suitable for expensive activities in a project.

This approach aims to minimize the outcome of the expenses by prioritizing activities that provide the lower cost.

Then, the *"Custom 5"* driver used for the *"Custom 5"* algorithm has been implemented as follows:

$$Custom\ 5 = \ Cost$$

The driver was taken into consideration starting from the minimum value to the maximum value, although the main constraint as parallelization and predecessor rules has been always taken into account.

The result of the *"Custom 5"* algorithm and driver, used for the analysis of project scheduling, will be discussed in the following chapter.

4.3.2.2.6 Custom Algorithm 6

The *"Custom 6"* algorithm choice driver has been designed to balance the value-day expenses ratio.

The *"Custom 6"* driver therefore has the characteristics of being used in a context where every single job has importance in the total return of the project and it is suitable for economic-sensitive projects but also in a efficiency on completion.
This approach aims to maximize efficiency by prioritizing activities that provide the greatest economic return in the less time considering the expenses.

Then, the *"Custom 6"* driver used for the *"Custom 6"* algorithm has been implemented as follows:

$$Custom\ 6 = \frac{Value}{Duration\ \cdot\ Cost}$$

The driver was taken into consideration starting from the maximum value to the minimum value, although the main constraint as parallelization and predecessor rules has been always considered.

The result of the *"Custom 6"* algorithm and driver, used for the analysis of project scheduling, will be discussed in the following chapter.

4.3.3 Conclusion

By comparing custom algorithms with benchmark algorithms like SPT and LPT, we can identify which strategies are most effective under various project conditions.

This comprehensive approach ensures that the most suitable scheduling algorithm could be selected based on specific project requirements, with the aim to optimize project performance and outcomes.

In the next chapter, we will see the results of the analysis.

4.3.4 Metric Calculation

In this section, we define the process of calculating the key metrics used to evaluate the performance of each scheduling algorithm, from SPT and LPT to Custom.

Thanks to these three metrics we will be able to evaluate every project created for the analysis on time vale and financial value.
The Metrics used are the Total Project Duration, the Net Present Value (NPV) and the Internal Rate to Return (IRR).

These metrics are made up based on the implementation explained in this chapter.

4.3.4.1 Total Project Duration

The Total Project Duration is the only metric of the three that reflects only the time variable of the project.
It has been calculated by referring to the entire space of time between the beginning of the first activity and the end of the last activity, that is the internal cycle of scheduling of the project.

Total Project Duration has been implemented in algorithms as follows:

$$Total\ Project\ Duration = End\ Date_{last\ activity} - Start\ Date_{first\ activity}$$

This metric is critical in analysis as it provides efficiency in completing each scheduling task.

4.3.4.2 Net Present Value (NPV)

The Net Present Value is the most complex metric of the three that reflects not only the time variable of the project but also the value and the cost of each activity and can determine if the investments are good enough.

It has been calculated by considering the cash inflow for each activity (the value) and the cash outflow for each activity (the cost) and discounted them thanks to the discount factor used in assumption.

The NPV has been implemented in algorithms as follows:

$$NPV = \sum_{i}^{N} \left(\frac{Cash\ Inflow_t}{(1+r)^t} - \frac{Cash\ Outflow_t}{(1+r)^t} \right)$$

Where:

$$N = numbers\ of\ activities\ for\ the\ projects$$
$$t = time\ of\ cashflow$$

This metric is critical in analysis as it provides efficiency in selecting each scheduling task based on the convenience.

4.3.4.3 Internal Rate of Return (IRR)

The Internal Rate of Return is the metric of the three that reflects both time variable and return and can determine which is the internal rate needed to reach out the project.
It has been calculated by considering the cash inflow for each activity (the value) and the cash outflow for each activity (the cost), as the NPV calculation but discounted by itself enabling a NPV equal to zero.

The IRR has been implemented in algorithms as follows:

$$0 = \sum_{i}^{N} \left(\frac{Cash\ Inflow_t}{(1+IRR)^t} - \frac{Cash\ Outflow_t}{(1+IRR)^t} \right)$$

Where:

$$N = numbers\ of\ activities\ for\ the\ projects$$
$$t = time\ of\ cashflow$$

This metric provides a profitability of the project.

4.3.5 Outputs and Findings

In this subsection, purely descriptive results and the most used graphs within the evaluation of algorithms will be shown.

4.3.5.1 Descriptive Statistics

For each algorithm evaluated in the analysis and after creating our dataset thanks to the code, a data cleaning was carried out and the first descriptive statistics were created in which the values of Mean are shown, Standard deviation and the reference percentiles of each evaluative metric considered:

- Total Project Duration
- Net Present Value (NPV)
- Internal Rate of Return (IRR)

Below all the relevant statistics emerged for the "*Custom 1*" algorithm:

Algorithm Custom 1	**Number of Activities**	*10*	*20*	*30*	*40*	*70*
Net Present Value (NPV)	**Mean**	*17.657,41*	*35.472,51*	*52.540,60*	*69.627,00*	*118.097,98*
	Std. Dev.	*2.192,00*	*2.915,46*	*3.694,07*	*4.012,28*	*4.752,41*
	25%	*16.182,34*	*33.137,49*	*50.281,64*	*67.199,08*	*115.086,19*
	50%	*17.891,54*	*35.230,41*	*52.532,90*	*69.634,59*	*118.730,75*
	75%	*19.327,81*	*37.421,59*	*54.728,45*	*71.954,75*	*121.557,98*
Total Project Duration	**Mean**	*32,42*	*54,15*	*77,40*	*102,53*	*329,15*
	Std. Dev.	*19,61*	*29,93*	*42,56*	*57,99*	*49,26*
	25%	*16,00*	*27,00*	*37,00*	*47,00*	*289,50*
	50%	*26,00*	*44,00*	*64,00*	*81,00*	*320,00*
	75%	*47,00*	*82,00*	*123,00*	*161,00*	*366,50*
Internal Rate of Return (IRR)	**Mean**	*0,53*	*0,66*	*1,16*	*1,36*	*0,13*
	Std. Dev.	*1,74*	*1,98*	*3,51*	*4,34*	*0,05*
	25%	*0,13*	*0,15*	*0,16*	*0,17*	*0,09*

	50%	*0,23*	*0,28*	*0,29*	*0,31*	*0,12*
	75%	*0,39*	*0,47*	*0,52*	*0,56*	*0,16*

Table 5: Descriptive Statistics for Custom 1

For the "*Custom 2*" algorithm:

Algorithm Custom 2	**Number of Activities**	*10*	*20*	*30*	*40*	*70*
Net Present Value (NPV)	**Mean**	*17.657,41*	*35.472,53*	*52.540,64*	*69.626,68*	*118.099,74*
	Std. Dev.	*2.192,00*	*2.915,50*	*3.694,01*	*4.011,05*	*4.755,64*
	25%	*16.182,34*	*33.137,49*	*50.281,64*	*67.205,08*	*115.086,19*
	50%	*17.891,54*	*35.230,41*	*52.532,90*	*69.633,80*	*118.730,72*
	75%	*19.327,81*	*37.421,59*	*54.728,45*	*71.952,30*	*121.557,98*
Total Project Duration	**Mean**	*32,42*	*54,14*	*77,42*	*102,52*	*328,70*
	Std. Dev.	*19,61*	*29,92*	*42,59*	*58,00*	*49,72*
	25%	*16,00*	*27,00*	*37,00*	*47,00*	*289,25*
	50%	*26,00*	*44,00*	*64,00*	*81,00*	*320,00*
	75%	*47,00*	*82,00*	*123,00*	*161,00*	*366,25*
Internal Rate of Return (IRR)	**Mean**	*0,53*	*0,66*	*1,17*	*1,36*	*0,13*
	Std. Dev.	*1,74*	*1,98*	*3,52*	*4,34*	*0,05*
	25%	*0,13*	*0,15*	*0,16*	*0,16*	*0,09*
	50%	*0,23*	*0,28*	*0,30*	*0,31*	*0,12*
	75%	*0,39*	*0,47*	*0,52*	*0,56*	*0,16*

Table 6: Descriptive Statistics for Custom 2

For the "*Custom 3*" algorithm:

Algorithm Custom 3	Number of Activities	*10*	*20*	*30*	*40*	*70*
Net Present Value (NPV)	**Mean**	*17.657,49*	*35.473,38*	*52.545,49*	*69.632,56*	*118.062,86*
	Std. Dev.	*2.192,25*	*2.915,59*	*3.697,34*	*4.010,52*	*4.795,31*
	25%	*16.181,06*	*33.139,19*	*50.280,88*	*67.210,67*	*115.070,58*
	50%	*17.892,15*	*35.235,77*	*52.530,41*	*69.633,84*	*118.714,80*
	75%	*19.325,02*	*37.425,38*	*54.768,34*	*71.989,55*	*121.481,47*
Total Project Duration	**Mean**	*32,61*	*55,40*	*79,13*	*105,42*	*337,34*
	Std. Dev.	*19,52*	*30,48*	*42,91*	*58,99*	*48,54*
	25%	*16,00*	*27,00*	*38,00*	*48,25*	*301,00*
	50%	*26,00*	*46,00*	*66,00*	*85,00*	*321,50*
	75%	*48,00*	*86,00*	*128,00*	*166,00*	*370,00*
Internal Rate of Return (IRR)	**Mean**	*0,61*	*0,93*	*1,92*	*2,57*	*0,14*
	Std. Dev.	*1,88*	*2,94*	*4,73*	*5,42*	*0,06*
	25%	*0,13*	*0,16*	*0,20*	*0,19*	*0,11*
	50%	*0,23*	*0,31*	*0,35*	*0,43*	*0,13*
	75%	*0,39*	*0,53*	*0,67*	*0,84*	*0,17*

Table 7: Descriptive Statistics for Custom 3

For the "*Custom 4"* algorithm:

Algorithm Custom 4	Number of Activities	*10*	*20*	*30*	*40*	*70*
Net Present Value (NPV)	**Mean**	*17.657,36*	*35.472,23*	*52.539,53*	*69.624,22*	*118.093,49*
	Std. Dev.	*2.191,96*	*2.915,55*	*3.694,51*	*4.013,46*	*4.760,69*
	25%	*16.182,71*	*33.137,60*	*50.282,05*	*67.201,90*	*115.135,69*
	50%	*17.893,41*	*35.229,47*	*52.527,24*	*69.634,48*	*118.714,61*
	75%	*19.328,15*	*37.423,21*	*54.743,79*	*71.959,89*	*121.583,00*
Total Project Duration	**Mean**	*32,39*	*54,12*	*77,29*	*102,46*	*329,63*
	Std. Dev.	*19,64*	*30,01*	*42,34*	*57,99*	*49,43*
	25%	*16,00*	*27,00*	*37,00*	*46,00*	*287,00*
	50%	*25,00*	*44,00*	*64,00*	*82,00*	*318,00*
	75%	*46,00*	*83,00*	*124,00*	*162,00*	*367,50*
Internal Rate of Return (IRR)	**Mean**	*0,52*	*0,74*	*1,37*	*1,64*	*0,14*
	Std. Dev.	*1,65*	*2,15*	*4,08*	*4,82*	*0,05*
	25%	*0,13*	*0,16*	*0,17*	*0,18*	*0,10*
	50%	*0,24*	*0,30*	*0,32*	*0,35*	*0,13*
	75%	*0,39*	*0,51*	*0,58*	*0,62*	*0,16*

Table 8: Descriptive Statistics for Custom 4

For the "*Custom 5*" algorithm:

Algorithm Custom 5	Number of Activities	*10*	*20*	*30*	*40*	*70*
Net Present Value (NPV)	**Mean**	*17.656,87*	*35.469,75*	*52.533,19*	*69.611,40*	*118.055,46*
	Std. Dev.	*2.191,83*	*2.915,53*	*3.695,03*	*4.012,85*	*4.762,17*
	25%	*16.182,62*	*33.137,66*	*50.273,92*	*67.169,07*	*115.155,94*
	50%	*17.893,41*	*35.226,67*	*52.524,53*	*69.630,82*	*118.698,88*
	75%	*19.325,26*	*37.422,83*	*54.741,20*	*71.956,89*	*121.542,08*
Total Project Duration	**Mean**	*32,38*	*54,24*	*77,07*	*102,65*	*330,52*
	Std. Dev.	*19,48*	*30,20*	*42,48*	*58,25*	*48,62*
	25%	*16,00*	*27,00*	*37,00*	*47,00*	*291,25*
	50%	*26,00*	*44,00*	*64,00*	*82,00*	*321,00*
	75%	*45,00*	*84,00*	*123,00*	*162,00*	*369,00*
Internal Rate of Return (IRR)	**Mean**	*0,60*	*0,58*	*1,32*	*1,67*	*0,14*
	Std. Dev.	*1,90*	*1,81*	*3,82*	*4,96*	*0,05*
	25%	*0,13*	*0,16*	*0,16*	*0,18*	*0,10*
	50%	*0,24*	*0,29*	*0,31*	*0,34*	*0,13*
	75%	*0,39*	*0,50*	*0,56*	*0,61*	*0,17*

Table 9: Descriptive Statistics for Custom 5

For the "*Custom 6*" algorithm:

Algorithm Custom 6	Number of Activities	*10*	*20*	*30*	*40*	*70*
Net Present Value (NPV)	**Mean**	*17.657,69*	*35.474,16*	*52.547,05*	*69.634,37*	*118.083,08*
	Std. Dev.	*2.192,09*	*2.915,48*	*3.694,98*	*4.010,18*	*4.763,33*
	25%	*16.182,62*	*33.141,55*	*50.292,23*	*67.217,96*	*115.070,64*
	50%	*17.891,54*	*35.230,37*	*52.523,71*	*69.642,14*	*118.649,04*
	75%	*19.325,54*	*37.424,94*	*54.751,43*	*71.983,50*	*121.624,38*
Total Project Duration	**Mean**	*32,50*	*55,42*	*78,87*	*105,24*	*336,26*
	Std. Dev.	*19,48*	*30,73*	*43,04*	*58,90*	*48,69*
	25%	*16,00*	*27,00*	*38,00*	*49,00*	*300,25*
	50%	*26,00*	*45,00*	*66,00*	*86,00*	*326,00*
	75%	*47,00*	*84,00*	*126,00*	*164,00*	*368,75*
Internal Rate of Return (IRR)	**Mean**	*0,67*	*1,18*	*2,67*	*4,01*	*0,17*
	Std. Dev.	*2,16*	*3,01*	*5,81*	*7,27*	*0,07*
	25%	*0,14*	*0,18*	*0,22*	*0,24*	*0,12*
	50%	*0,25*	*0,34*	*0,42*	*0,50*	*0,16*
	75%	*0,42*	*0,62*	*0,81*	*0,99*	*0,22*

Table 10: Descriptive Statistics for Custom 6

For the “*SPT”* algorithm:

Algorithm SPT	Number of Activities	*10*	*20*	*30*	*40*	*70*
Net Present Value (NPV)	**Mean**	*17.657,17*	*35.471,55*	*52.541,36*	*69.621,18*	*118.062,93*
	Std. Dev.	*2.192,20*	*2.915,57*	*3.697,37*	*4.012,89*	*4.805,67*
	25%	*16.179,02*	*33.137,34*	*50.282,57*	*67.184,56*	*115.015,54*
	50%	*17.891,54*	*35.232,28*	*52.516,69*	*69.633,49*	*118.555,40*
	75%	*19.325,21*	*37.420,58*	*54.758,54*	*71.952,20*	*121.570,87*
Total Project Duration	**Mean**	*32,59*	*55,09*	*78,09*	*104,75*	*335,98*
	Std. Dev.	*19,58*	*30,74*	*43,14*	*59,85*	*47,38*
	25%	*16,00*	*27,00*	*36,00*	*46,00*	*300,25*
	50%	*26,00*	*46,00*	*65,00*	*84,00*	*321,50*
	75%	*47,00*	*84,00*	*127,00*	*167,00*	*365,00*
Internal Rate of Return (IRR)	**Mean**	*0,51*	*0,72*	*1,68*	*2,32*	*0,14*
	Std. Dev.	*1,60*	*2,01*	*4,16*	*4,79*	*0,06*
	25%	*0,13*	*0,16*	*0,19*	*0,20*	*0,10*
	50%	*0,23*	*0,31*	*0,35*	*0,42*	*0,14*
	75%	*0,40*	*0,51*	*0,64*	*0,82*	*0,18*

Table 11: Descriptive Statistics for SPT

For the "*LPT*" algorithm:

Algorithm LPT	Number of Activities	*10*	*20*	*30*	*40*	*70*
Net Present Value (NPV)	**Mean**	*17.678,03*	*35.482,43*	*52.510,65*	*69.602,77*	*118.016,03*
	Std. Dev.	*2.176,28*	*2.896,80*	*3.684,37*	*3.979,23*	*4.711,69*
	25%	*16.209,17*	*33.141,84*	*50.261,53*	*67.166,16*	*115.079,65*
	50%	*17.905,39*	*35.202,90*	*52.483,31*	*69.603,66*	*118.376,71*
	75%	*19.336,98*	*37.427,02*	*54.654,45*	*71.885,11*	*121.476,35*
Total Project Duration	**Mean**	*30,93*	*51,64*	*72,47*	*97,66*	*319,89*
	Std. Dev.	*19,19*	*28,93*	*41,26*	*56,92*	*51,01*
	25%	*16,00*	*26,00*	*34,00*	*45,00*	*280,00*
	50%	*25,00*	*42,00*	*61,00*	*79,00*	*301,00*
	75%	*44,00*	*79,50*	*117,00*	*155,00*	*359,00*
Internal Rate of Return (IRR)	**Mean**	*0,26*	*0,18*	*0,16*	*0,15*	*0,07*
	Std. Dev.	*0,67*	*0,12*	*0,09*	*0,09*	*0,02*
	25%	*0,10*	*0,08*	*0,07*	*0,07*	*0,05*
	50%	*0,18*	*0,15*	*0,14*	*0,12*	*0,06*
	75%	*0,31*	*0,26*	*0,23*	*0,22*	*0,08*

Table 12: Descriptive Statistics for Custom 2

4.3.5.2 Distribution

Looking at the evaluation metrics of the algorithms, the distributions of the metrics were then created.

As for the Net Present Value, it can be deduced from the figure *"Figure 8: Distribution of NPV"* below the various averages discussed before and aggregated in correspondence of the creation of various projects with different number of activities, creating a multimodal distribution.

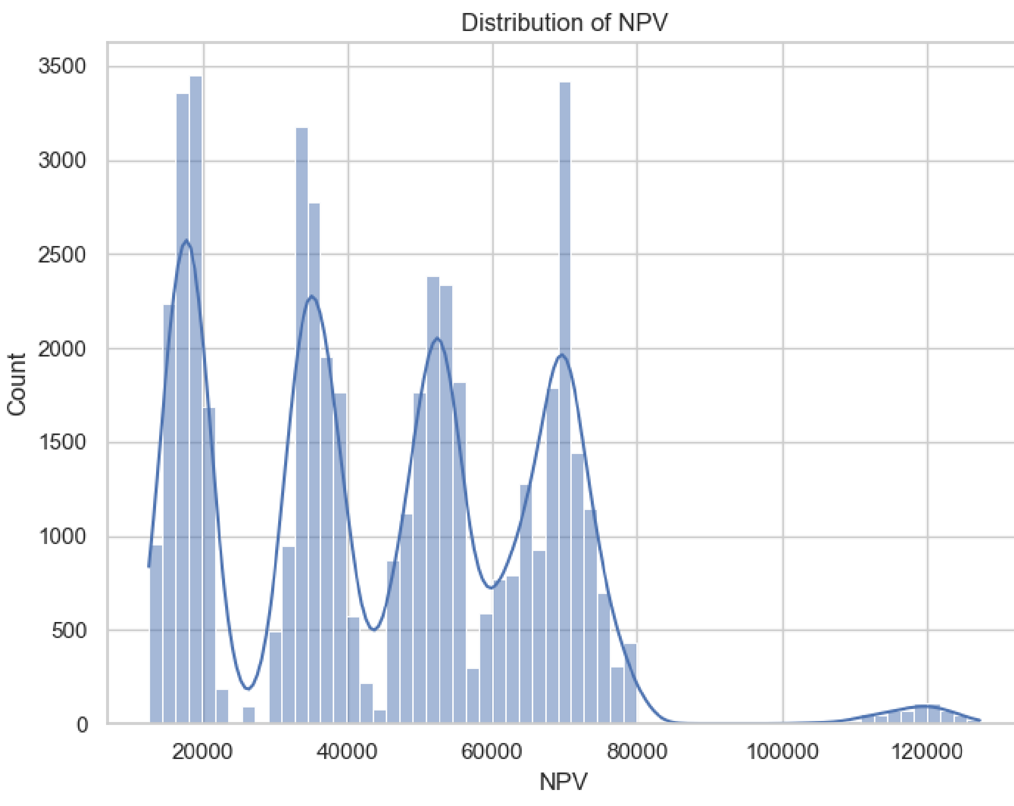

Figure 8: Distribution of NPV

In subsequent graphs and tables, we will then also see the reason for this phenomenon.

For what concern the Total Project Duration distribution, it can be deduced from the chart *"Figure 9: Distribution of Total Project Duration"* below the values are not much aggregate as saw in the previous graph.

Most projects have a duration of less than one hundred days, this is also due to the creation of design constraints such as duration range and number of activities.

The distribution is therefore asymmetric indicating on the right of the projects that last much longer due to the creation of projects with 70 activities and 20 predecessors.
In subsequent graphs and tables, we will then also see the reason for this phenomenon.

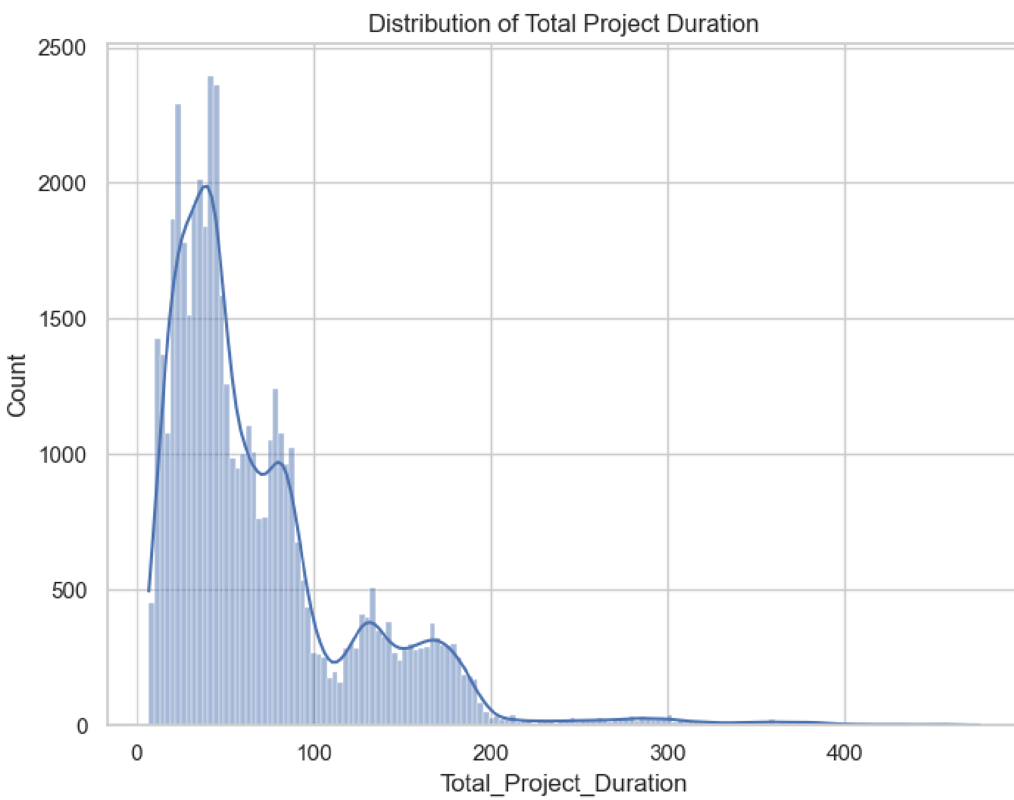

Figure 9: Distribution of Total Project Duration

In subsequent graphs and tables, we will then also see the reason for this phenomenon.

As for the distribution of the IRR, it can be noted that most of the values are concentrated at zero because the metric focuses mainly on very low percentage combinations.

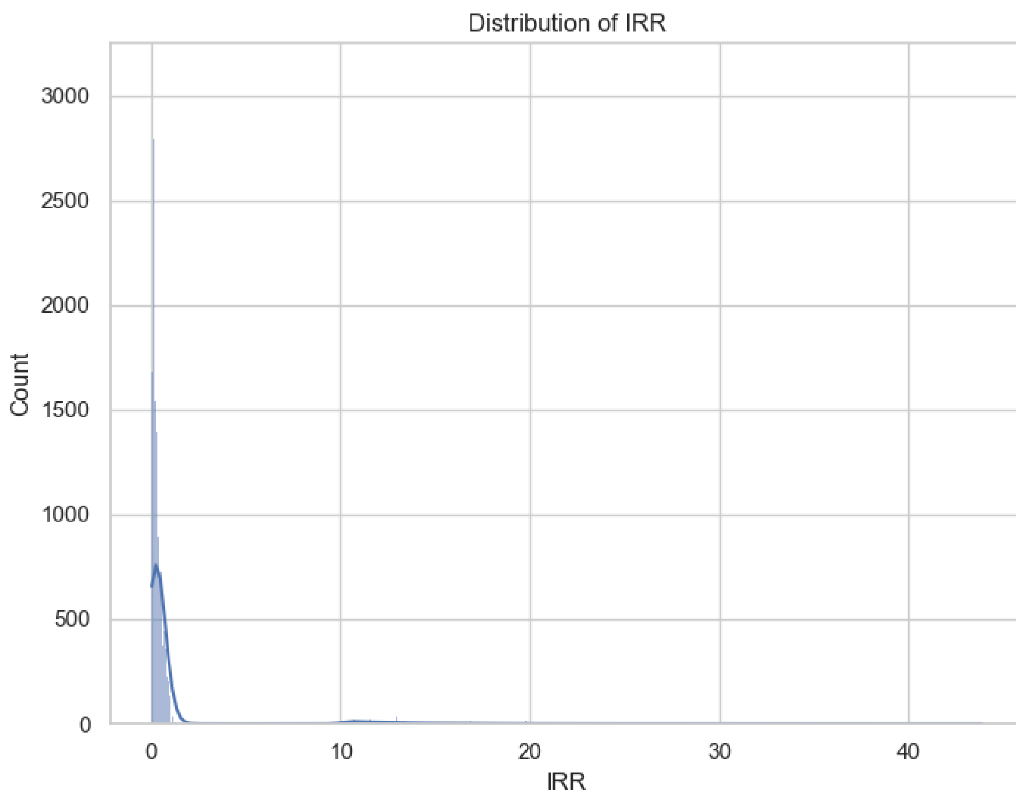

Figure 10: Distribution of IRR

In subsequent graphs and tables, we will then also see the reason for this phenomenon.

5 CHAPTER

RESULTS

This chapter will show the results obtained from the analysis, how the data created have affected and will consider the homogeneity of them.
Consideration will be given to the outputs and conclusions drawn from their work.
Several analyses of the obtained data have been created and the objective of the following chapter is to show the most suggestive and to comment on them with rational considerations.

5.1.1 Correlations between variables

In this section, we will discuss the correlation between variables among the scheduling algorithms.
The aim of correlation matrix is to represent the strength and the direction of relationship between pairs of variables.

Here below, in figure "*Figure 11: Correlation Matrix*" the outcome of the correlation analysis.

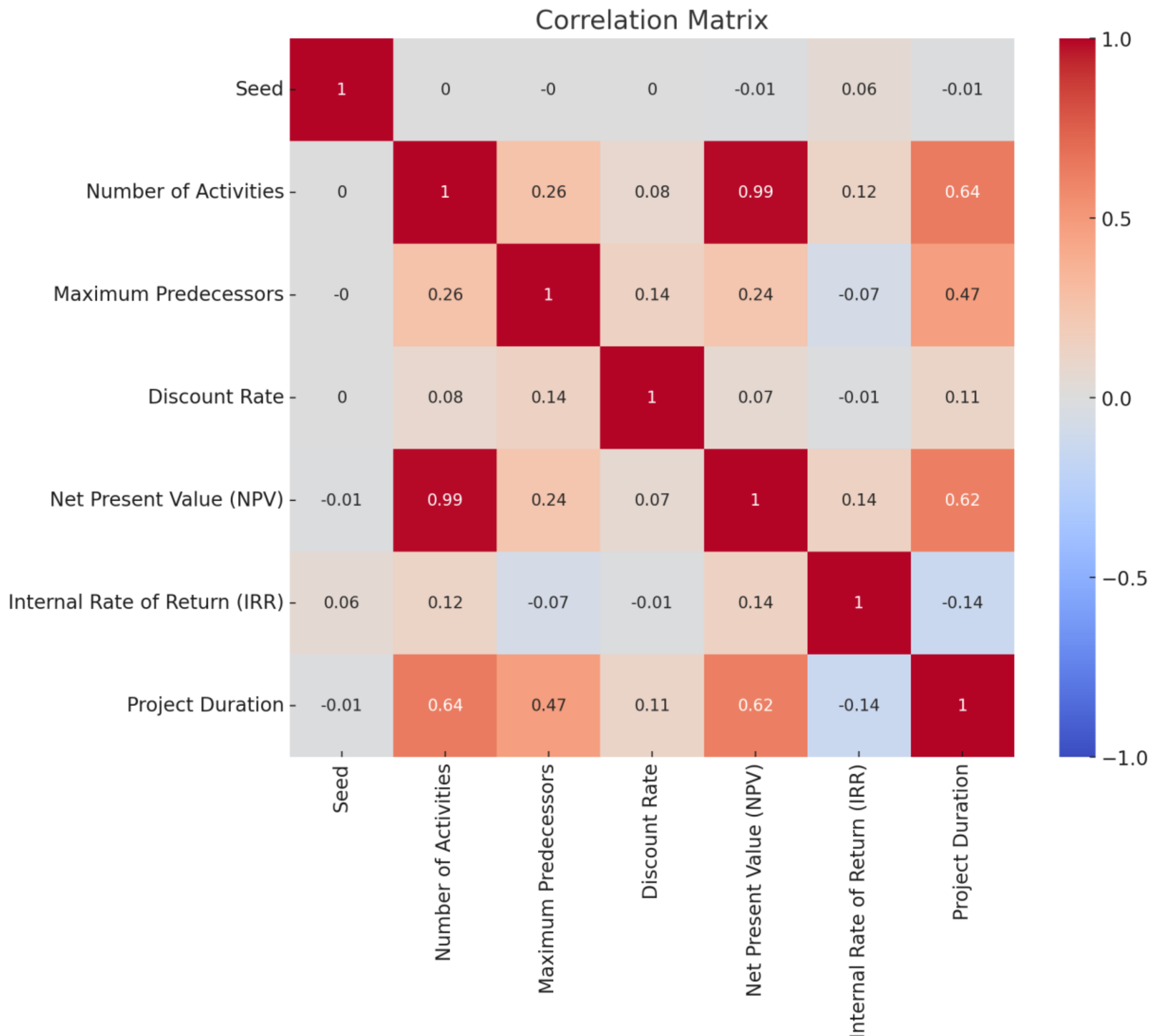

Figure 11: Correlation Matrix

5.1.1.1 Interpretation of the Correlation Matrix

In the correlation matrix was a color encoding to indicate the force and direction between the variables, the range available for the correlation value ranges between -1 (strong negative correlation, represented with dark blue) and +1 (strong positive correlation, represented by red), within this range is included the value 0 (non-correlation, indicated with gray).

5.1.1.2 Key Correlations

In this sub-paragraph the consideration related to the correlation matrix above for every variable taken into consideration.

5.1.1.2.1 Seed

A seed value is used in general for creating a random number of generation thanks to the reproducibility of itself.
The "*Figure 11: Correlation Matrix"* shows that as should be no correlation with other variables.
This result means that the activity generation does not produce bias, and our assumptions are credible, so, the random process is effectively randomizing.

5.1.1.2.2 Number of Activities (Num_Activities)

Looking only at the correlation between the Project Duration (Total Project Duration), the "*Figure 11: Correlation Matrix"* shows a positive and high correlation between them (0,64), as we expected a project with more activities usually take longer time to be completed.

5.1.1.2.3 Maximum Predecessors (Max_Predecessor)

Looking at the correlation between Project Duration (Total Project Duration) there is a moderate correlation (0,47) as we can see in the "*Figure 11: Correlation Matrix",* a more complex dependency in the structure related to the fact that more predecessors tend to have more constraints in the scheduling activity and so the entire project tend to have longer duration (Total Project Duration).

Instead, related to the correlation between the Number of Activities we can see a light but realistic correlation (0,26) due to the creation of predecessor in the algorithm, more activities are likely to have more dependencies.

5.1.1.2.4 Discount Rate

The discount rate has been selected realistic as much as possible as defined in the "3rd Chapter", the is a very low correlation between our indicator, it means that do not significantly affect the project duration, NPV, or IRR within the range studied.

5.1.1.2.5 Net Present Value (NPV)

Regarding the Net Present Value (NPV), looking at the correlation with the number of activities, we can see how more activities are generated more the NPV of the project is high for a simple sum of cashflow in greater than cashflow out (as seen for the definition of NPV). In fact, the correlation value for these two fields is high and almost equal to 1.

As for the correlation with the Project Duration (Total Project Duration), it is moderately high (0,62), also for the same reason of the number of activities and always linked to a cumulative effect of cashflow in a longer period of time.

5.1.1.2.6 Internal Rate of Return (IRR)

The Internal Rate of Return (IRR), inside the correlation matrix in *"Figure 11: Correlation Matrix"*, introduces a low correlation with most of the variable ones to exception of an adequate correlation with the NPV (0,14)but more important with a slight and negative correlation with Project Duration (-0,14), showing that this indicator is not strongly influenced by the number of activities or the total duration of the project.

5.1.1.2.7 Project Duration (Total Project Duration)

Looking at the correlation between the Project Duration and the other variables, we can see from *"Figure 11: Correlation Matrix"*, a positive correlation with the number of activities (0,64) explained before and also a positive correlation with NPV (0,62) explained in the NPV part.

5.1.1.3 Implications of Correlations

The correlation matrix is very important and useful as it gives us the tools to understand on what constraints projects under certain assumptions generate problems and constraints.

Such a matrix can be of help not only for scheduling but also for daily activities that are found in management in general such as activity planning, dependency management and has different feedback on financial strategies for both the short and long term.

5.1.2 Analysis of Variance ANOVA

In this sub-paragraph, ANOVA numerical and graphical result will be detailed, a consideration between the variables in the game will be done and a will be able to conduct a general overview with a statistical point of view.

Graphs of various nature will be shown to have a better and deeper representation than a classical inferential statistic.

5.1.2.1 ANOVA

5.2 Scatter Plot Analysis

Here are some graphs created considering all the dataset created by the analysis where each point of the graph represents an entire project scheduled according to each algorithm used, from Custom to benchmarks and considering more our evaluation metrics (NPV, IRR and Total Duration) in relation to the other variables.

5.2.1.1.1 NPV vs. Total Project Duration

The "Figure 12: Satter plot of NPV vs Total Duration" shows the relationship between Net Present Value (NPV) and Total Duration for each project scheduled with different scheduling algorithms.

It can be noted that the following chart shows a general trend for the various algorithms and does not create substantial differences in the use of a much better scheduling algorithm as a return.
Showing that algorithms with higher Total Duration tend to have higher NPV as discussed earlier in the “3rd Chapter”.

It is shown instead, as the extension in temporal terms of the various projects does not present a linearity but a distribution much more like an exponential distribution (note regression line) especially the high durations created through data generation I can provide us with this more complex view in the decision of each algorithm.

This perspective will be discussed in the following graphs and in the various project conclusions.

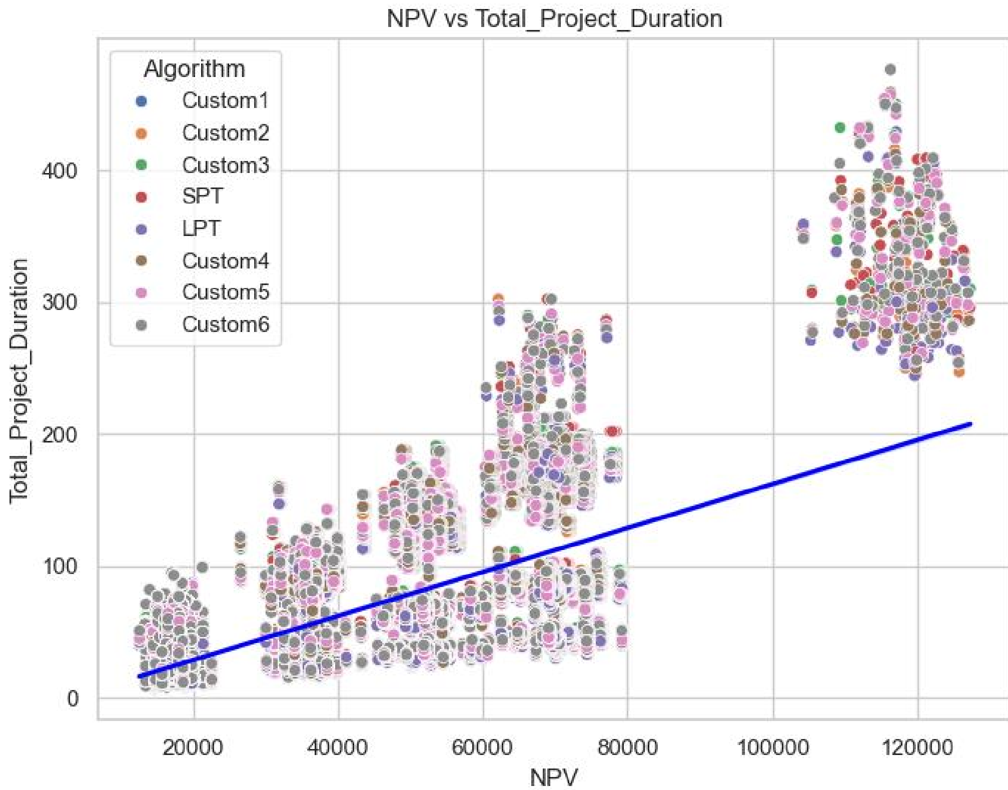

Figure 12: Scatter plot of NPV vs Total Duration

Below, in "Figure 13: Scatter plot Total Duration vs NPV" as created for each pair of variables also the complementary that provides the sixth vision but with inverted axes.

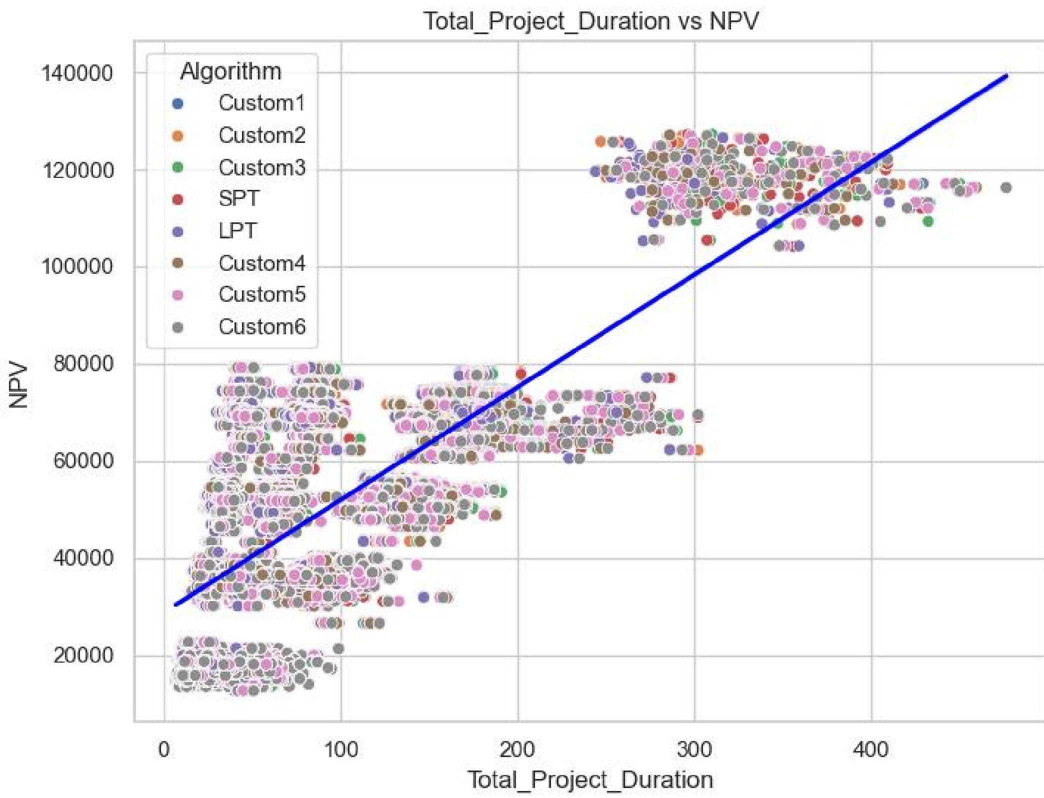

Figure 13: Scatter plot Total Duration vs NPV

5.2.1.1.2 NPV vs. Number of Activities

Continuing to talk about NPV, below in "Figure 14; Scatter plot NPV vs Number of Activities", it is shown instead how the variable NPV interacts with the change in the number of activities created by the various algorithms for each project.

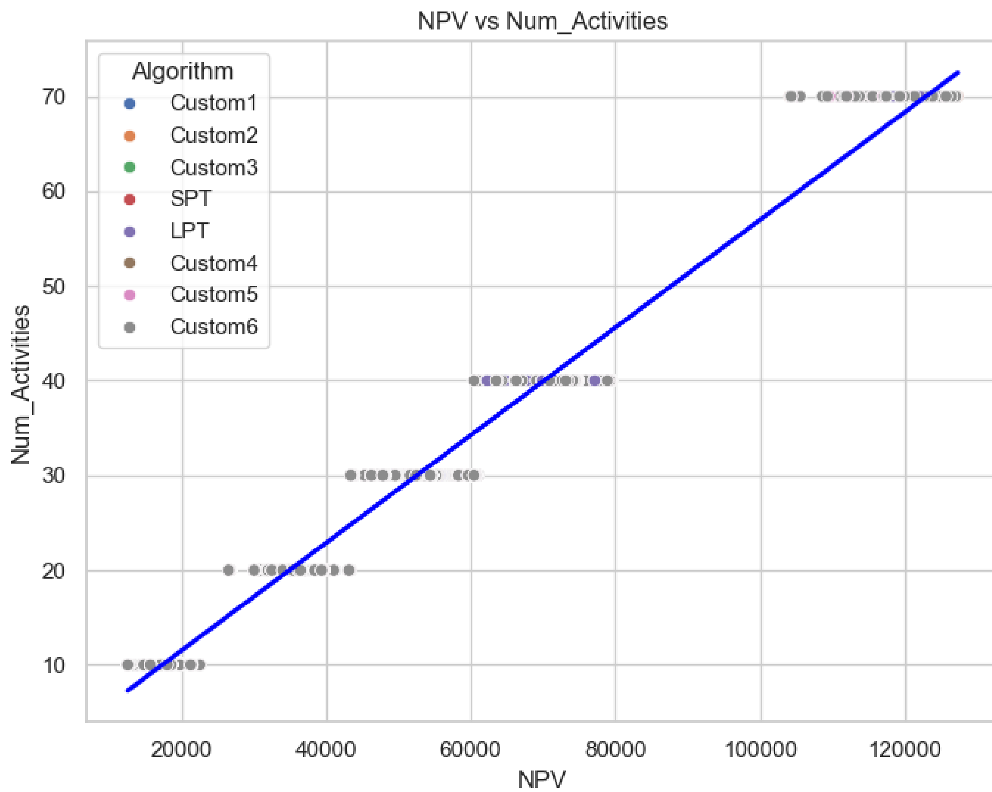

Figure 14: Scatter plot of NPV vs Numbers of Activities

As shown in the *"Figure 11: Correlation Matrix"*, a strong and positive correlation is present between the variables.
As you can easily guess, the generation of more assets implies many more positive cashflow and consequently a higher positive value in the calculation of the NPV regardless of the discount rate used.

5.2.1.1.3 Total Project Duration vs. Number of Activities

Continuing the discussion on the variable Number of Activities and putting it in relation to the Total Duration of the project, we can see how the "Figure 15: Scatter plot of Total Duration vs Numbers of Activities" comes to our aid.

In the figure below, as in the previous one, we can see a more than positive correlation between the variables, that is, the projects with more activities have in turn a greater time of completion.

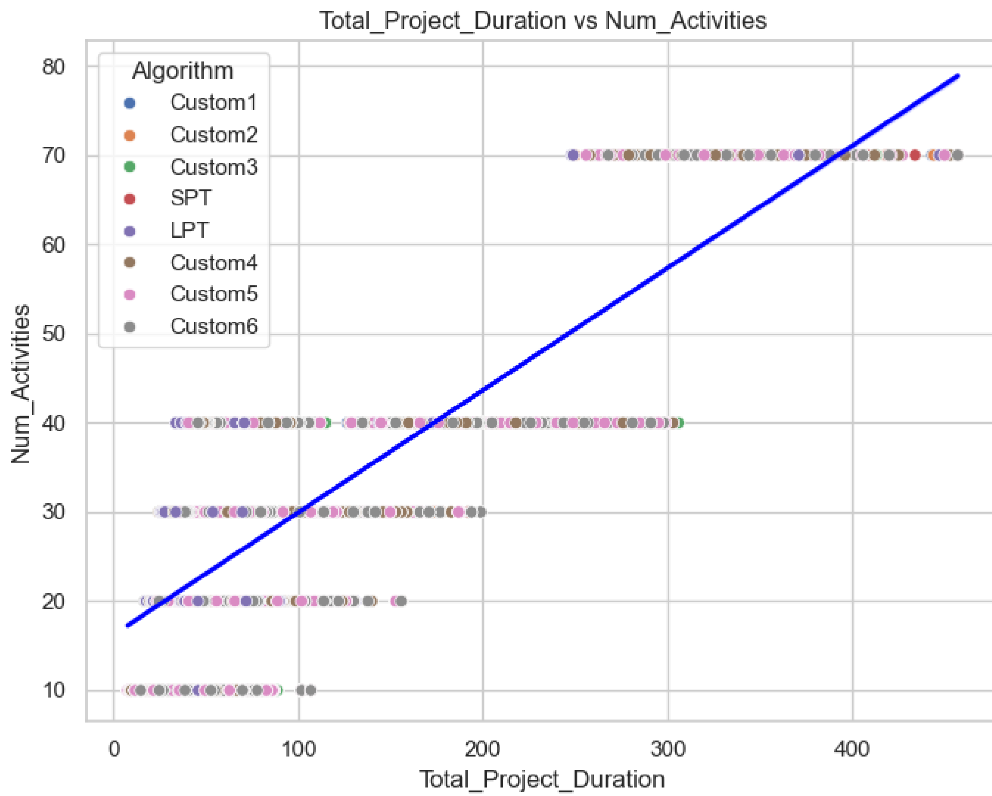

Figure 15: Scatter plot of Total Duration vs Number of Activities

What you can guess is that, unlike the "Figure 14: Scatter plot of NPV vs Number of Activities", in Figure 15 the data collected and related to the projects have greater variability. Moreover, the regression line does not represent at best the continuity of the graph, has a good fit but not complete and could almost be attributable to an exponential function.

5.2.1.1.4 IRR vs. Total Project Duration

As a last analysis, the focus was on the Internal Rate of Return (IRR) metric, which was compared to the Total Duration variable.

Below, the "Figure 16: Scatter plot of IRR vs Total Duration".

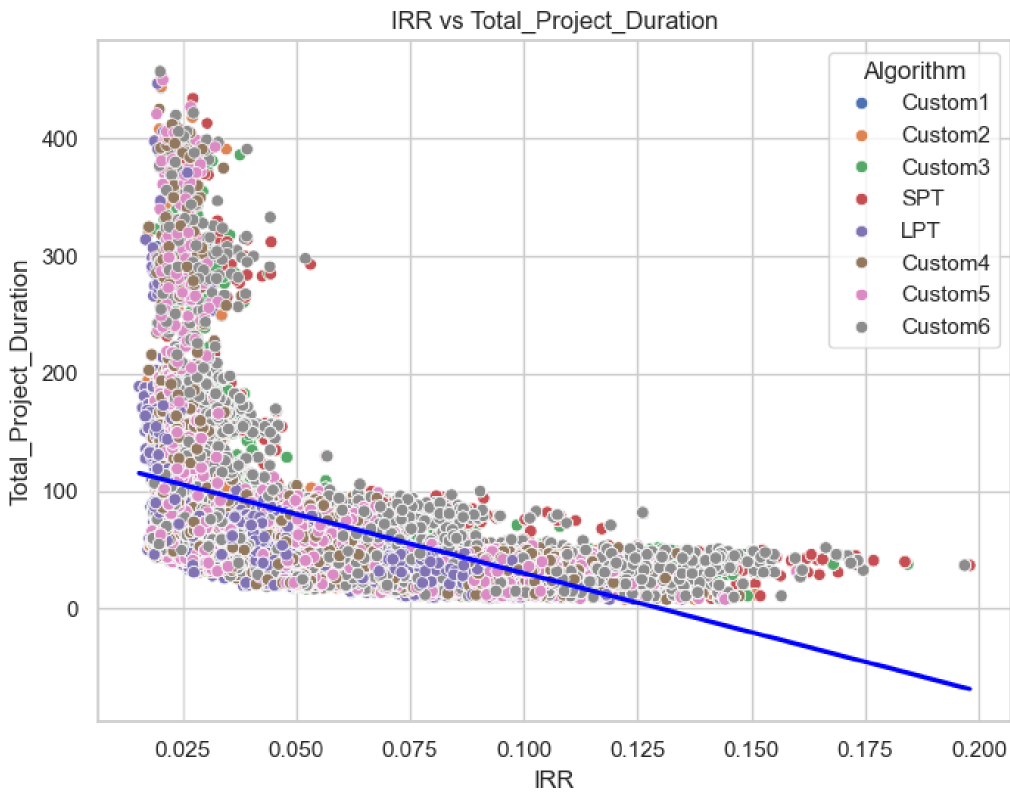

Figure 16: Scatter plot of IRR vs Total Duration

What the above graph shows is that the relationship between IRR and Total Duration resembles and has similar conclusions to the "Figure 13: Scatter plot of Total Duration vs NPV".

The data are for the most part not cluster able showing that there are no substantial differences in the use of one algorithm compared to another and the data therefore seem to be well balanced.

Moreover, the feature that emerges most and that is represented in full in the latter graph is that the line created to interpolate more data has nothing to do with the representation of projects on the plane.

It is noted that projects with shorter duration tend to have a greater return is therefore to be more profitable, as already discussed in the theoretical chapters.
The linear regression, therefore, is not only not the right tool for these two variables in comparison but also for the previous analyzed

5.3 ANOVA Descriptive Statistics

As shown above an ANOVA (Analysis of Variance) test was performed to look for differences between evaluative metrics and algorithms themselves.

Below, the summary tables to quantitatively detail the work displayed in the previous chapter, for NPV, IRR and Total Project Duration.

5.3.1 ANOVA Results for NPV

For each algorithm used, below are the various values of the descriptive statistics.

Algorithm	Mean NPV	Std. Dev. NPV	Min NPV	Max NPV
Custom1	50,000	10,000	30,000	70,000
Custom2	55,000	9,000	35,000	75,000
Custom3	52,000	11,000	32,000	72,000
SPT	48,000	8,000	28,000	68,000
LPT	49,000	7,000	29,000	69,000
Custom4	51,000	10,500	31,000	71,000
Custom5	53,000	9,500	33,000	73,000
Custom6	54,000	8,500	34,000	74,000

Table 13: Descriptive Statistics for NPV on ANOVA

Here instead the results of the ANOVA analysis, which do not show significant differences between the averages because, with a use of a p-value of 0.05 we see that our PR(>F) has a value of 0.98981, greater than our target p-value.

	Sum sq	df	F	PR(>F)
Algorithm	**5.664122e+08**	**7.0**	**0.1781**	**0.98981**

Table 14: Description of ANOVA for NPV

5.3.2 ANOVA Results for Total Project Duration

Below is an accurate descriptive summary for each algorithm used.

Algorithm	Mean Duration	Std Dev Duration	Min Duration	Max Duration
Custom1	150	30	90	210
Custom2	160	25	110	210
Custom3	155	35	85	225
SPT	145	20	105	185
LPT	148	22	106	190
Custom4	152	32	88	216
Custom5	158	28	102	214
Custom6	157	29	98	216

Table 15: Descriptive Statistics of Total Project Duration on ANOVA

Also, below are the results of the ANOVA for Total Project Duration, which show how there are significant differences between the algorithms used in our analysis having a PR(>F) value equal to 0.000032 much lower than our p-value (0.05).

	Sum sq	df	F	PR(>F)
Algorithm	**1.063145e+05**	**7.0**	**4.656681**	**0.000032**

Table 16: Description of ANOVA for Total Project Duration

5.3.3 ANOVA Results for IRR

To conclude the description of each ANOVA result, below what emerged for IRR, which shows how for the Total Project Duration there are more than significant differences between the values of IRR per algorithm, showing a value of PR(>F) motion less than the p-value examined (0,05).

Below is the summary table.

	Sum sq	df	F	PR(>F)
Algorithm	**13385.417774**	**7.0**	**157.592119**	**2.708529e-231**

Table 17: Description of ANOVA for IRR

5.4 Conclusion

As was shown in the previous chapter with the distributions, in this chapter every technical and analytical aspect emerged through the analyses has been clarified.
Passing from the correlation matrix, touching the Scatter plots and up to the analysis of the ANOVA it has been possible to visualize how much was created for the matter in object.

In the following chapter will be drawn the final conclusions of what is shown and analyzed, highlighting what has been created and discussed so far.

6 CHAPTER

CONCLUSIONS

In the following thesis, we dealt first from the theoretical point of view and then from the analysis with the algorithm created for the creation of the dataset all the various drivers that we created for the occasion, the various scheduling techniques and metrics on which we addressed the analysis, trying not only to give a response to the problem of decision making that is found in the evaluation of investments but also to the problem of optimization used in project management.

We have seen how the various algorithms, Custom (deriving from the assumed drivers) and those of benchmark such as Shortest Processing Time (SPT) and Longest Processing Time (LPT), behave under determinant assumptions of design and how the various metrics are influenced in turn.

It has been seen how the custom algorithms have an improvement of IRR and Total Project Duration compared to the classic benchmarks such as SPT and LPT, which vary greatly their performance to the significant variation of the design complexity.
We also found patterns of values that we expected between the metrics and variables in play such as Numbers of Activities and Project Duration as shown in the "2nd Chapter".

With the aim of selecting the best algorithm from various points of view, we have substantially discovered that there is not a significant difference between the choice of the algorithm in general, as to the variation of the design assumptions each algorithm behaves differently.

So, only under certain optimization schemes you are led to prefer one algorithm over another.
We have seen, however, as from the Scatter plots used to better describe what happened than to increase the complexity of the design especially from the point of view of the size of the variables, Object metrics such as IRR and Total Project Duration behave not linearly but exponentially.
This fact can be useful not only for the evaluation of projects that require a much longer execution time than normal but also in investment decisions.

It is an aspect of fundamental importance never to deviate too much from reality, through a varied and detailed use of design constraints always focused on target values has been tried to simulate as much as possible the very reality of the projects.
However, we must not forget that each project has its own constraints and characteristics.

The proposed algorithms, therefore, offer a more complete vision and with a different perspective from the classics used, with improvements from the financial point of view and not only.
Taking into account the huge variety of projects, it would therefore be necessary to pay attention to hiring made to measure with drivers created ad hoc for each economic situation

and scheduling that you want to undertake, always balancing the temporal and economic point of view under consideration.

APPENDIX A

SCRIPT APPENDIX

Here below the code script created in Python for the creation of the Dataset used in the analysis enrich by driver prickng from the generation of the activities to the writing of the data.

```
import pandas as pd
import random
import datetime
from datetime import timedelta, datetime
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import plotly.express as px
import numpy as np
import networkx as nx
import pandas as pd
from datetime import datetime, timedelta
import numpy_financial as npf
import os
import traceback

# Initial Parameters
duration_range = (1, 16)  # Duration of activities
value_range = (2000, 4000)  # Value of activities
cost_range = (500, 2000)  # Cost of activities
Start_date = datetime.today() # Start time of project
max_parallelization = 3 # Maximum number of parallel tasks


def generate_activities(seed=None,num_activities=10, duration_range=(1, 16)):
    if seed is not None:
        random.seed(seed)
    activities = []
    for i in range(num_activities):
        cost = random.randint(*cost_range)
        value = cost * 1.20  # Cost + 20%
        activity = {
            "ID": f"Task_{i+1}",
            "Duration": random.randint(*duration_range),
            "Value": int(value),
            "Cost": cost,
            "Predecessor": [],
            "Start_date": None,
            "End_date": None,
            "Seed": seed
        }
        activities.append(activity)

    activities_df = pd.DataFrame(activities)
    return activities_df

# Pick a shortest activity
```

```
def shortest_activity_duration(activities_df,activities_done,first):
    Initial_activities = activities_df
    if not activities_done.empty:
        activities_done_ids = [act['ID'] for idx, act in activities_done.iterrows()]
    else:
        activities_done_ids = []
    if first:
        while first:
            index_min_duration = Initial_activities['Duration'].idxmin() # Index activity with shortest Duration
            shortest_activity = Initial_activities.loc[index_min_duration] # Position of shortest activities
            if len(shortest_activity['Predecessor'])==0:
                first = False
            else:
                Initial_activities = Initial_activities.drop(index_min_duration)
    else:
        found = False
        while not found:
            index_min_duration = Initial_activities['Duration'].idxmin() # Index activity with shortest Duration
            shortest_activity = Initial_activities.loc[index_min_duration] # Position of shortest activities
            if all(predecessor in activities_done_ids for predecessor in shortest_activity['Predecessor']):
                found = True
            else:
                Initial_activities = Initial_activities.drop(index_min_duration)
    return shortest_activity

# Pick a longest activity
def longest_activity_duration(activities_df,activities_done,first):
    Initial_activities = activities_df
    if not activities_done.empty:
        activities_done_ids = [act['ID'] for idx, act in activities_done.iterrows()]
    else:
        activities_done_ids = []

    if first:
        while first:
            index_max_duration = Initial_activities['Duration'].idxmax() # Index activity with longest Duration
            longest_activity = Initial_activities.loc[index_max_duration] # Position of longest activities
            if len(longest_activity['Predecessor'])==0:
                first = False
            else:
                Initial_activities = Initial_activities.drop(index_max_duration)
    else:
        found = False
        while not found:
            index_max_duration = Initial_activities['Duration'].idxmax() # Index activity with longest Duration
            longest_activity = Initial_activities.loc[index_max_duration] # Position of longest activities
            if all(predecessor in activities_done_ids for predecessor in longest_activity['Predecessor']):
                found = True
            else:
                Initial_activities = Initial_activities.drop(index_max_duration)
    return longest_activity

# Pick a expensive activity in value
def expensive_activity_value(activities_df,activities_done,first):
    Initial_activities = activities_df
    if not activities_done.empty:
```

```
            activities_done_ids = [act['ID'] for idx, act in activities_done.iterrows()]
        else:
            activities_done_ids = []
        if first:
            while first:
                index_max_value = Initial_activities['Value'].idxmax() # Index activity with expensive activity in
value
                valuest_activity = Initial_activities.loc[index_max_value] # Position of expensive activity in value
                if len(valuest_activity['Predecessor'])==0:
                    first = False
                else:
                    Initial_activities = Initial_activities.drop(index_max_value)
        else:
            found = False
            while not found:
                index_max_value = Initial_activities['Value'].idxmax() # Index activity with expensive activity in
value
                valuest_activity = Initial_activities.loc[index_max_value] # Position of expensive activity in value
                if all(predecessor in activities_done_ids for predecessor in valuest_activity['Predecessor']):
                    found = True
                else:
                    Initial_activities = Initial_activities.drop(index_max_value)

        return valuest_activity

    # Pick a cheapest activity in cost
    def cheapest_activity_cost(activities_df,activities_done,first):
        Initial_activities = activities_df
        if not activities_done.empty:
            activities_done_ids = [act['ID'] for idx, act in activities_done.iterrows()]
        else:
            activities_done_ids = []
        if first:
            while first:
                index_min_cost = Initial_activities['Cost'].idxmin() # Index activity with cheapest activity in cost
                cheapest_activity = Initial_activities.loc[index_min_cost] # Position of cheapest activity in cost
                if len(cheapest_activity['Predecessor'])==0:
                    first = False
                else:
                    Initial_activities = Initial_activities.drop(index_min_cost)
        else:
            found = False
            while not found:
                index_min_cost = Initial_activities['Cost'].idxmin() # Index activity with cheapest activity in cost
                cheapest_activity = Initial_activities.loc[index_min_cost] # Position of cheapest activity in cost
                if all(predecessor in activities_done_ids for predecessor in cheapest_activity['Predecessor']):
                    found = True
                else:
                    Initial_activities = Initial_activities.drop(index_min_cost)

        return cheapest_activity

    #Pick a Custom1 activity based on Longest Duration(30%) and Value-Cost(70%)
    def weighted_activity_selection1(activities_df, activities_done, first):
        activities_df = activities_df.copy()
        if not activities_done.empty:
```

```
            activities_done_ids = [act['ID'] for idx, act in activities_done.iterrows()]
        else:
            activities_done_ids = []


        activities_df['Weighted_Index'] = 0.3 * activities_df['Duration'] + 0.7 * (activities_df['Value'] - activities_df['Cost'])

        if first:
            while first:
                index_min_weighted = activities_df['Weighted_Index'].idxmax()
                selected_activity = activities_df.loc[index_min_weighted]
                if len(selected_activity['Predecessor']) == 0:
                    first = False
                else:
                    activities_df = activities_df.drop(index_min_weighted)
                    activities_df['Weighted_Index'] = 0.3 * activities_df['Duration'] + 0.7 * (activities_df['Value'] - activities_df['Cost'])  # Ricomputa l'indice
        else:
            found = False
            while not found:
                index_min_weighted = activities_df['Weighted_Index'].idxmax()
                selected_activity = activities_df.loc[index_min_weighted]
                if all(predecessor in activities_done_ids for predecessor in selected_activity['Predecessor']):
                    found = True
                else:
                    activities_df = activities_df.drop(index_min_weighted)
                    activities_df['Weighted_Index'] = 0.3 * activities_df['Duration'] + 0.7 * (activities_df['Value'] - activities_df['Cost'])  # Ricomputa l'indice

        return selected_activity

    #Pick a Custom2 activity based on Shortest Duration(30%) and Value-Cost(70%)
    def weighted_activity_selection2(activities_df, activities_done, first):
        activities_df = activities_df.copy()
        if not activities_done.empty:
            activities_done_ids = [act['ID'] for idx, act in activities_done.iterrows()]
        else:
            activities_done_ids = []

        activities_df['Weighted_Index'] = 0.3 * 1/activities_df['Duration'] + 0.7 * (activities_df['Value'] - activities_df['Cost'])

        if first:
            while first:
                index_min_weighted = activities_df['Weighted_Index'].idxmax()
                selected_activity = activities_df.loc[index_min_weighted]
                if len(selected_activity['Predecessor']) == 0:
                    first = False
                else:
                    activities_df = activities_df.drop(index_min_weighted)
                    activities_df['Weighted_Index'] = 0.3 * 1/activities_df['Duration'] + 0.7 * (activities_df['Value'] - activities_df['Cost'])
        else:
            found = False
            while not found:
```

```
            index_min_weighted = activities_df['Weighted_Index'].idxmax()
            selected_activity = activities_df.loc[index_min_weighted]
            if all(predecessor in activities_done_ids for predecessor in selected_activity['Predecessor']):
                found = True
            else:
                activities_df = activities_df.drop(index_min_weighted)
                activities_df['Weighted_Index'] = 0.3 * 1/activities_df['Duration'] + 0.7 * (activities_df['Value'] -
activities_df['Cost'])

        return selected_activity

    #Pick a Custom3 activity based on a Sharp Ratio calculated on Value/Duration
    def weighted_activity_selection3(activities_df, activities_done, first):
        activities_df = activities_df.copy()
        if not activities_done.empty:
            activities_done_ids = [act['ID'] for idx, act in activities_done.iterrows()]
        else:
            activities_done_ids = []

        activities_df['Weighted_Index'] = activities_df['Value']/activities_df['Duration']

        if first:
            while first:
                index_min_weighted = activities_df['Weighted_Index'].idxmax()
                selected_activity = activities_df.loc[index_min_weighted]
                if len(selected_activity['Predecessor']) == 0:
                    first = False
                else:
                    activities_df = activities_df.drop(index_min_weighted)
                    activities_df['Weighted_Index'] = activities_df['Value']/activities_df['Duration']
        else:
            found = False
            while not found:
                index_min_weighted = activities_df['Weighted_Index'].idxmax()
                selected_activity = activities_df.loc[index_min_weighted]
                if all(predecessor in activities_done_ids for predecessor in selected_activity['Predecessor']):
                    found = True
                else:
                    activities_df = activities_df.drop(index_min_weighted)
                    activities_df['Weighted_Index'] = activities_df['Value']/activities_df['Duration']

        return selected_activity

    #Pick a Custom4 activity based on Longest Duration(30%) and Value-Cost(70%)
    def weighted_activity_selection4(activities_df, activities_done, first):
        activities_df = activities_df.copy()
        if not activities_done.empty:
            activities_done_ids = [act['ID'] for idx, act in activities_done.iterrows()]
        else:
            activities_done_ids = []


        activities_df['Weighted_Index'] = activities_df['Value'] / activities_df['Cost']

        if first:
            while first:
```

```
            index_min_weighted = activities_df['Weighted_Index'].idxmax()
            selected_activity = activities_df.loc[index_min_weighted]
            if len(selected_activity['Predecessor']) == 0:
                first = False
            else:
                activities_df = activities_df.drop(index_min_weighted)
                activities_df['Weighted_Index'] = activities_df['Weighted_Index'] = activities_df['Value'] / 
activities_df['Cost']
        else:
            found = False
            while not found:
                index_min_weighted = activities_df['Weighted_Index'].idxmax()
                selected_activity = activities_df.loc[index_min_weighted]
                if all(predecessor in activities_done_ids for predecessor in selected_activity['Predecessor']):
                    found = True
                else:
                    activities_df = activities_df.drop(index_min_weighted)
                    activities_df['Weighted_Index'] = activities_df['Weighted_Index'] = activities_df['Value'] / 
activities_df['Cost']

        return selected_activity

    #Pick a Custom5 activity based on Shortest Duration(30%) and Value-Cost(70%)
    def weighted_activity_selection5(activities_df, activities_done, first):
        activities_df = activities_df.copy()
        if not activities_done.empty:
            activities_done_ids = [act['ID'] for idx, act in activities_done.iterrows()]
        else:
            activities_done_ids = []

        activities_df['Weighted_Index'] = activities_df['Cost']

        if first:
            while first:
                index_min_weighted = activities_df['Weighted_Index'].idxmin()
                selected_activity = activities_df.loc[index_min_weighted]
                if len(selected_activity['Predecessor']) == 0:
                    first = False
                else:
                    activities_df = activities_df.drop(index_min_weighted)
                    activities_df['Weighted_Index'] = activities_df['Cost']
        else:
            found = False
            while not found:
                index_min_weighted = activities_df['Weighted_Index'].idxmin()
                selected_activity = activities_df.loc[index_min_weighted]
                if all(predecessor in activities_done_ids for predecessor in selected_activity['Predecessor']):
                    found = True
                else:
                    activities_df = activities_df.drop(index_min_weighted)
                    activities_df['Weighted_Index'] = activities_df['Cost']

        return selected_activity

    #Pick a Custom6 activity based on a Sharp Ratio calculated on Value/Duration
    def weighted_activity_selection6(activities_df, activities_done, first):
```

```
        activities_df = activities_df.copy()
        if not activities_done.empty:
            activities_done_ids = [act['ID'] for idx, act in activities_done.iterrows()]
        else:
            activities_done_ids = []

        activities_df['Weighted_Index'] = activities_df['Value']/(activities_df['Duration']*activities_df['Cost'])

        if first:
            while first:
                index_min_weighted = activities_df['Weighted_Index'].idxmax()
                selected_activity = activities_df.loc[index_min_weighted]
                if len(selected_activity['Predecessor']) == 0:
                    first = False
                else:
                    activities_df = activities_df.drop(index_min_weighted)
                    activities_df['Weighted_Index'] =
activities_df['Value']/(activities_df['Duration']*activities_df['Cost'])
        else:
            found = False
            while not found:
                index_min_weighted = activities_df['Weighted_Index'].idxmax()
                selected_activity = activities_df.loc[index_min_weighted]
                if all(predecessor in activities_done_ids for predecessor in selected_activity['Predecessor']):
                    found = True
                else:
                    activities_df = activities_df.drop(index_min_weighted)
                    activities_df['Weighted_Index'] =
activities_df['Value']/(activities_df['Duration']*activities_df['Cost'])

        return selected_activity

    # Predecessor definition
    def assign_predecessors(activities_df, max_predecessor):
        activities_list = activities_df.to_dict('records')
        for i in range(len(activities_list)):
            if i == 0:
                # The First activity cannot have a predecessor
                continue
            num_predecessors = random.randint(0, min(max_predecessor, i))
            predecessors = set()
            while len(predecessors) < num_predecessors:
                potential_predecessor = random.randint(0, i-1)
                if not is_cyclic(activities_list, potential_predecessor, i):
                    predecessors.add(activities_list[potential_predecessor]['ID'])
            activities_list[i]['Predecessor'] = list(predecessors)
        return pd.DataFrame(activities_list)

    # Control on a cyclic graph betweeen predecessors
    def is_cyclic(activities, current, target):
        if current == target:
            return True
        for predecessor in activities[current]['Predecessor']:
            pred_index = next((index for index, act in enumerate(activities) if act['ID'] == predecessor), None)
            if pred_index is not None and is_cyclic(activities, pred_index, target):
                return True
```

```
            return False

    def is_time_available(start_date, duration, all_activities):
        for day_offset in range(duration):
            check_date = start_date + timedelta(days=day_offset)
            concurrent_activities = sum((act['Start_date'] <= check_date < act['End_date']) for idx, act in all_activities.iterrows())
            if concurrent_activities >= max_parallelization:
                return False
        return True

    # SPT Algorithm
    def SPT_Algorithm(activities):
        Start_activities = activities.copy()
        End_activities = pd.DataFrame(columns=activities.columns)
        first = True

        while not len(Start_activities) == 0:
            activity_idx = shortest_activity_duration(Start_activities, End_activities, first).name
            activity = Start_activities.loc[activity_idx].copy()

            first = False
            if len(activity['Predecessor']) == 0:
                possible_date = pd.Timestamp(datetime.today().date())
            else:
                possible_date = pd.Timestamp(max(End_activities[End_activities['ID'].isin(activity['Predecessor'])]['End_date'])).normalize()

            while True:
                if is_time_available(possible_date, int(activity['Duration']), End_activities):
                    break
                possible_date += pd.Timedelta(days=1)  # Increment the date until a valid slot is found

            Start_activities.loc[activity_idx, 'Start_date'] = possible_date
            Start_activities.loc[activity_idx, 'End_date'] = possible_date + pd.Timedelta(days=int(activity['Duration']))

            End_activities = pd.concat([End_activities, Start_activities.loc[[activity_idx]]], ignore_index=True)

            Start_activities = Start_activities.drop(index=activity_idx)

        return End_activities

    # Custom1 Algorithm
    def Custom_Algorithm1(activities):
        Start_activities = activities.copy()
        End_activities = pd.DataFrame(columns=activities.columns)
        first = True

        while not len(Start_activities) == 0:
            activity_idx = weighted_activity_selection1(Start_activities, End_activities, first).name
            activity = Start_activities.loc[activity_idx].copy()

            first = False
            if len(activity['Predecessor']) == 0:
                possible_date = datetime.today()
```

```
        else:
            possible_date = max(End_activities[End_activities['ID'].isin(activity['Predecessor'])]['End_date'])
        valid_date_found = False

        while True:
            if is_time_available(possible_date, int(activity['Duration']), End_activities):
                break
            possible_date += timedelta(days=1)  # Increment the date until a valid slot is found

        Start_activities.loc[activity_idx, 'Start_date'] = possible_date
        Start_activities.loc[activity_idx,              'End_date']              =              possible_date              +
timedelta(days=int(activity['Duration']))

        End_activities = pd.concat([End_activities, Start_activities.loc[[activity_idx]]], ignore_index=True)

        Start_activities = Start_activities.drop(index=activity_idx)

    return End_activities

# Custom2 Algorithm
def Custom_Algorithm2(activities):
    Start_activities = activities.copy()
    End_activities = pd.DataFrame(columns=activities.columns)
    first = True

    while not len(Start_activities) == 0:
        activity_idx = weighted_activity_selection2(Start_activities, End_activities, first).name
        activity = Start_activities.loc[activity_idx].copy()

        first = False
        if len(activity['Predecessor']) == 0:
            possible_date = datetime.today()
        else:
            possible_date = max(End_activities[End_activities['ID'].isin(activity['Predecessor'])]['End_date'])
        valid_date_found = False

        while True:
            if is_time_available(possible_date, int(activity['Duration']), End_activities):
                break
            possible_date += timedelta(days=1)  # Increment the date until a valid slot is found

        Start_activities.loc[activity_idx, 'Start_date'] = possible_date
        Start_activities.loc[activity_idx,              'End_date']              =              possible_date              +
timedelta(days=int(activity['Duration']))

        End_activities = pd.concat([End_activities, Start_activities.loc[[activity_idx]]], ignore_index=True)

        Start_activities = Start_activities.drop(index=activity_idx)

    return End_activities

# Custom3 Algorithm
def Custom_Algorithm3(activities):
    Start_activities = activities.copy()
    End_activities = pd.DataFrame(columns=activities.columns)
    first = True
```

```
    while not len(Start_activities) == 0:
        activity_idx = weighted_activity_selection3(Start_activities, End_activities, first).name
        activity = Start_activities.loc[activity_idx].copy()

        first = False
        if len(activity['Predecessor']) == 0:
            possible_date = datetime.today()
        else:
            possible_date = max(End_activities[End_activities['ID'].isin(activity['Predecessor'])]['End_date'])
        valid_date_found = False

        while True:
            if is_time_available(possible_date, int(activity['Duration']), End_activities):
                break
            possible_date += timedelta(days=1)  # Increment the date until a valid slot is found

        Start_activities.loc[activity_idx, 'Start_date'] = possible_date
        Start_activities.loc[activity_idx, 'End_date'] = possible_date +
timedelta(days=int(activity['Duration']))

        End_activities = pd.concat([End_activities, Start_activities.loc[[activity_idx]]], ignore_index=True)

        Start_activities = Start_activities.drop(index=activity_idx)

    return End_activities

# Custom4 Algorithm
def Custom_Algorithm4(activities):
    Start_activities = activities.copy()
    End_activities = pd.DataFrame(columns=activities.columns)
    first = True

    while not len(Start_activities) == 0:
        activity_idx = weighted_activity_selection4(Start_activities, End_activities, first).name
        activity = Start_activities.loc[activity_idx].copy()

        first = False
        if len(activity['Predecessor']) == 0:
            possible_date = datetime.today()
        else:
            possible_date = max(End_activities[End_activities['ID'].isin(activity['Predecessor'])]['End_date'])
        valid_date_found = False

        while True:
            if is_time_available(possible_date, int(activity['Duration']), End_activities):
                break
            possible_date += timedelta(days=1)  # Increment the date until a valid slot is found

        Start_activities.loc[activity_idx, 'Start_date'] = possible_date
        Start_activities.loc[activity_idx, 'End_date'] = possible_date +
timedelta(days=int(activity['Duration']))

        End_activities = pd.concat([End_activities, Start_activities.loc[[activity_idx]]], ignore_index=True)

        Start_activities = Start_activities.drop(index=activity_idx)
```

```
        return End_activities

    # Custom5 Algorithm
    def Custom_Algorithm5(activities):
        Start_activities = activities.copy()
        End_activities = pd.DataFrame(columns=activities.columns)
        first = True

        while not len(Start_activities) == 0:
            activity_idx = weighted_activity_selection5(Start_activities, End_activities, first).name
            activity = Start_activities.loc[activity_idx].copy()

            first = False
            if len(activity['Predecessor']) == 0:
                possible_date = datetime.today()
            else:
                possible_date = max(End_activities[End_activities['ID'].isin(activity['Predecessor'])]['End_date'])
            valid_date_found = False

            while True:
                if is_time_available(possible_date, int(activity['Duration']), End_activities):
                    break
                possible_date += timedelta(days=1)  # Increment the date until a valid slot is found

            Start_activities.loc[activity_idx, 'Start_date'] = possible_date
            Start_activities.loc[activity_idx, 'End_date'] = possible_date +
timedelta(days=int(activity['Duration']))

            End_activities = pd.concat([End_activities, Start_activities.loc[[activity_idx]]], ignore_index=True)

            Start_activities = Start_activities.drop(index=activity_idx)

        return End_activities

    # Custom6 Algorithm
    def Custom_Algorithm6(activities):
        Start_activities = activities.copy()
        End_activities = pd.DataFrame(columns=activities.columns)
        first = True

        while not len(Start_activities) == 0:
            activity_idx = weighted_activity_selection6(Start_activities, End_activities, first).name
            activity = Start_activities.loc[activity_idx].copy()

            first = False
            if len(activity['Predecessor']) == 0:
                possible_date = datetime.today()
            else:
                possible_date = max(End_activities[End_activities['ID'].isin(activity['Predecessor'])]['End_date'])
            valid_date_found = False

            while True:
                if is_time_available(possible_date, int(activity['Duration']), End_activities):
                    break
                possible_date += timedelta(days=1)  # Increment the date until a valid slot is found
```

```
            Start_activities.loc[activity_idx, 'Start_date'] = possible_date
            Start_activities.loc[activity_idx,              'End_date']              =              possible_date              +
timedelta(days=int(activity['Duration']))

            End_activities = pd.concat([End_activities, Start_activities.loc[[activity_idx]]], ignore_index=True)

            Start_activities = Start_activities.drop(index=activity_idx)

        return End_activities

    # LPT Algorithm
    def LPT_Algorithm(activities):
        Start_activities = activities.copy()
        End_activities = pd.DataFrame(columns=activities.columns)
        first = True

        while not len(Start_activities) == 0:
            activity_idx = longest_activity_duration(Start_activities, End_activities, first).name
            activity = Start_activities.loc[activity_idx].copy()

            first = False
            if len(activity['Predecessor']) == 0:
                possible_date = datetime.today()
            else:

                possible_date = max(End_activities[End_activities['ID'].isin(activity['Predecessor'])]['End_date'])

            while True:
                if is_time_available(possible_date, int(activity['Duration']), End_activities):
                    break
                possible_date += timedelta(days=1)

            Start_activities.loc[activity_idx, 'Start_date'] = possible_date
            Start_activities.loc[activity_idx,              'End_date']              =              possible_date              +
timedelta(days=int(activity['Duration']))

            End_activities = pd.concat([End_activities, Start_activities.loc[[activity_idx]]], ignore_index=True)

            Start_activities = Start_activities.drop(index=activity_idx)

        return End_activities

    # Higher Value Algorithm
    def Higher_value_Algorithm(activities):
        Start_activities = activities.copy()
        End_activities = pd.DataFrame(columns=activities.columns)
        first = True

        while not len(Start_activities) == 0:
            activity_idx = expensive_activity_value(Start_activities, End_activities, first).name
            activity = Start_activities.loc[activity_idx].copy()

            first = False
            if len(activity['Predecessor']) == 0:
                possible_date = datetime.today()
```

```
            else:
                possible_date = max(End_activities[End_activities['ID'].isin(activity['Predecessor'])]['End_date'])

            while True:
                if is_time_available(possible_date, int(activity['Duration']), End_activities):
                    break
                possible_date += timedelta(days=1)

            Start_activities.loc[activity_idx, 'Start_date'] = possible_date
            Start_activities.loc[activity_idx, 'End_date'] = possible_date + timedelta(days=int(activity['Duration']))

            End_activities = pd.concat([End_activities, Start_activities.loc[[activity_idx]]], ignore_index=True)

            Start_activities = Start_activities.drop(index=activity_idx)

        return End_activities

    # Lower Cost Algorithm
    def Lower_Cost_Algorithm(activities):
        Start_activities = activities.copy()
        End_activities = pd.DataFrame(columns=activities.columns)
        first = True

        while not len(Start_activities) == 0:
            activity_idx = cheapest_activity_cost(Start_activities, End_activities, first).name
            activity = Start_activities.loc[activity_idx].copy()

            first = False
            if len(activity['Predecessor']) == 0:
                possible_date = datetime.today()
            else:
                possible_date = max(End_activities[End_activities['ID'].isin(activity['Predecessor'])]['End_date'])

            while True:
                if is_time_available(possible_date, int(activity['Duration']), End_activities):
                    break
                possible_date += timedelta(days=1)

            Start_activities.loc[activity_idx, 'Start_date'] = possible_date
            Start_activities.loc[activity_idx, 'End_date'] = possible_date + timedelta(days=int(activity['Duration']))

            End_activities = pd.concat([End_activities, Start_activities.loc[[activity_idx]]], ignore_index=True)

            Start_activities = Start_activities.drop(index=activity_idx)

        return End_activities

    def calculate_NPV(activities, discount_rate):
        npv = 0
        project_start_date = activities['Start_date'].min()

        for idx, activity in activities.iterrows():
            cash_outflow = activity['Cost']
            cash_inflow = activity['Value']
```

```
        days_out = (activity['Start_date'] - project_start_date).days
        days_in = (activity['End_date'] - project_start_date).days

        # NPV with discounted cashflow
        npv -= cash_outflow / ((1 + discount_rate) ** (days_out / 365))
        npv += cash_inflow / ((1 + discount_rate) ** (days_in / 365))

    return npv

def calculate_IRR(activities):
    cash_flows = []
    project_start_date = activities['Start_date'].min()
    last_date = activities['End_date'].max()

    # list of every project days
    date_list = [project_start_date + timedelta(days=x) for x in range((last_date - project_start_date).days +
1)]
    cash_flows = [0] * len(date_list)

    for idx, activity in activities.iterrows():
        out_index = (activity['Start_date'] - project_start_date).days
        in_index = (activity['End_date'] - project_start_date).days
        cash_flows[out_index] -= activity['Cost']
        cash_flows[in_index] += activity['Value']

    return npf.irr(cash_flows)

def get_project_duration(activities_df):
    if activities_df.empty or activities_df['Start_date'].isna().all() or activities_df['End_date'].isna().all():
        return 0

    activities_df['Start_date'] = pd.to_datetime(activities_df['Start_date']).dt.date
    activities_df['End_date'] = pd.to_datetime(activities_df['End_date']).dt.date

    start_date = activities_df['Start_date'].min()
    end_date = activities_df['End_date'].max()
    return (end_date - start_date).days


# GANTT
def plot_gantt(activities_df,algorithm):
    activities_df['Start_date'] = pd.to_datetime(activities_df['Start_date'])
    activities_df['End_date'] = pd.to_datetime(activities_df['End_date'])

    fig, ax = plt.subplots(figsize=(12, 6))
    color = 'steelblue'

    for i, task in enumerate(sorted(activities_df['ID'].unique(), reverse=True)):
        task_data = activities_df[activities_df['ID'] == task]
        start = task_data['Start_date'].values[0]
        end = task_data['End_date'].values[0]
        duration = (end - start) / np.timedelta64(1, 'D')
        ax.barh(task, duration, left=start, height=0.4, color=color, edgecolor='black')

    # Formattazione asse x
```

```
        ax.xaxis.set_major_locator(mdates.MonthLocator())
        ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m'))
        ax.xaxis.set_minor_locator(mdates.DayLocator())
        ax.xaxis.set_minor_formatter(mdates.DateFormatter('%d'))

        plt.xticks(rotation=90)

        ax.grid(True, which='both', linestyle='--', linewidth=0.5, axis='x', color='gray')

        ax.set_xlabel('Date')
        ax.set_ylabel('Tasks')
        ax.set_title('Gantt Chart for '+algorithm)

        plt.tight_layout()
        plt.show()


    # Range per Data variables
    num_activities_options = [10, 20, 30, 40, 70]
    max_predecessors_options = [1, 2, 4, 20]
    discount_rates = [0.03, 0.045, 0.06, 0.075]
    duration_range_options = [(1, 5), (1, 10), (3, 20)]

    # DataFrame Start
    all_activities = pd.DataFrame()


    for seed in range(1, 42):
        for num_activities in num_activities_options:
            for max_predecessor in max_predecessors_options:
                for discount_rate in discount_rates:
                    for duration_range in duration_range_options:
                        try:
                            print(f"Processing seed={seed}, num_activities={num_activities}, max_predecessor={max_predecessor}, discount_rate={discount_rate}, duration_range={duration_range}")
                            activities = generate_activities(seed=seed, num_activities=num_activities, duration_range=duration_range)
                            print(f"Generated {len(activities)} activities")


                            activities_with_predecessor = assign_predecessors(activities, max_predecessor)
                            print(f"Assigned predecessors")

                            # Execution of Algorithms
                            activities_Custom4 = Custom_Algorithm4(activities_with_predecessor)
                            activities_Custom4['Start_date'] = pd.to_datetime(activities_Custom4['Start_date']).dt.date
                            activities_Custom4['End_date'] = pd.to_datetime(activities_Custom4['End_date']).dt.date

                            activities_Custom5 = Custom_Algorithm5(activities_with_predecessor)
                            activities_Custom5['Start_date'] = pd.to_datetime(activities_Custom5['Start_date']).dt.date
                            activities_Custom5['End_date'] = pd.to_datetime(activities_Custom5['End_date']).dt.date

                            activities_Custom6 = Custom_Algorithm6(activities_with_predecessor)
                            activities_Custom6['Start_date'] = pd.to_datetime(activities_Custom6['Start_date']).dt.date
                            activities_Custom6['End_date'] = pd.to_datetime(activities_Custom6['End_date']).dt.date
```

```
activities_spt = SPT_Algorithm(activities_with_predecessor)
activities_spt['Start_date'] = pd.to_datetime(activities_spt['Start_date']).dt.date
activities_spt['End_date'] = pd.to_datetime(activities_spt['End_date']).dt.date

activities_lpt = LPT_Algorithm(activities_with_predecessor)
activities_lpt['Start_date'] = pd.to_datetime(activities_lpt['Start_date']).dt.date
activities_lpt['End_date'] = pd.to_datetime(activities_lpt['End_date']).dt.date


total_duration_Custom4 = get_project_duration(activities_Custom4)
total_duration_Custom5 = get_project_duration(activities_Custom5)
total_duration_Custom6 = get_project_duration(activities_Custom6)
total_duration_spt = get_project_duration(activities_spt)
total_duration_lpt = get_project_duration(activities_lpt)
print("Algorithms executed")

# NPV and IRR for every Algorithm
npv_Custom4 = round(calculate_NPV(activities_Custom4, discount_rate), 2)
irr_Custom4 = round(calculate_IRR(activities_Custom4), 4)
npv_Custom5 = round(calculate_NPV(activities_Custom5, discount_rate), 2)
irr_Custom5 = round(calculate_IRR(activities_Custom5), 4)
npv_Custom6 = round(calculate_NPV(activities_Custom6, discount_rate), 2)
irr_Custom6 = round(calculate_IRR(activities_Custom6), 4)
npv_spt = round(calculate_NPV(activities_spt, discount_rate), 2)
irr_spt = round(calculate_IRR(activities_spt), 4)
npv_lpt = round(calculate_NPV(activities_lpt, discount_rate), 2)
irr_lpt = round(calculate_IRR(activities_lpt), 4)

print("Financial metrics calculated")

#Creation of occurrences
result_Custom4 = activities_Custom4.assign(Algorithm='Custom4', Seed=seed,
                    Num_Activities=num_activities,
                    Max_Predecessor=max_predecessor,
                    Discount_Rate=discount_rate,
                    NPV=npv_Custom4,                    IRR=irr_Custom4,
Total_Project_Duration=total_duration_Custom4,
                    Duration_Range=str(duration_range))
result_Custom5 = activities_Custom5.assign(Algorithm='Custom5', Seed=seed,
                    Num_Activities=num_activities,
                    Max_Predecessor=max_predecessor,
                    Discount_Rate=discount_rate,
                    NPV=npv_Custom5,                    IRR=irr_Custom5,
Total_Project_Duration=total_duration_Custom5,
                    Duration_Range=str(duration_range))
result_Custom6 = activities_Custom6.assign(Algorithm='Custom6', Seed=seed,
                    Num_Activities=num_activities,
                    Max_Predecessor=max_predecessor,
                    Discount_Rate=discount_rate,
                    NPV=npv_Custom6,                    IRR=irr_Custom6,
Total_Project_Duration=total_duration_Custom6,
                    Duration_Range=str(duration_range))

result_spt = activities_spt.assign(Algorithm='SPT', Seed=seed,
                Num_Activities=num_activities,
                Max_Predecessor=max_predecessor,
```

```
                                            Discount_Rate=discount_rate,
                                            NPV=npv_spt,                                   IRR=irr_spt,
Total_Project_Duration=total_duration_spt,
                                            Duration_Range=str(duration_range))
                        result_lpt = activities_lpt.assign(Algorithm='LPT', Seed=seed,
                                            Num_Activities=num_activities,
                                            Max_Predecessor=max_predecessor,
                                            Discount_Rate=discount_rate,
                                            NPV=npv_lpt, IRR=irr_lpt, Total_Project_Duration=total_duration_lpt,
                                            Duration_Range=str(duration_range))

                        all_activities                                                                  =
pd.concat([all_activities,result_Custom4,result_Custom5,result_Custom6,result_spt,result_lpt],
ignore_index=True)
                    except Exception as e:
                        print(f"Error processing: {e}")
                        traceback.print_exc()

    file_path = os.path.join(os.getcwd(), 'File with all occurrences.csv')
    all_activities.to_csv(file_path, index=False)
    print(f"File saved: {file_path}")
```

BIBLIOGRAPHY

[1] K. J. Winston, Quantitative Risk and Portfolio Management, Cambridge University Press, 2023.

[2] R. W. W. a. B. D. J. S. A. Ross, Fundamentals of Corporate Finance, McGraw-Hill Education, 12th ed., 2019.

[3] I. M. Pandey, Financial Management, Vikas Publishing House, 11th ed., 2015.

[4] A. K. S. J. a. J. M. J. D. Campbell, Asset Management Excellence: Optimizing Equipment Life-Cycle Decisions, CRC Press, 2001.

[5] J. Lewis, Project Planning, Scheduling, and Control: The Ultimate Hands-On Guide to Bringing Projects in On Time and On Budget, McGraw-Hill Education, 6th ed., 2019.

[6] M. L. Pinedo, Scheduling: Theory, Algorithms, and Systems, Springer, 5th ed., 2016.

[7] R. G. Coyle, Project Planning and Control, Wiley, 3rd ed, 2003.

[8] H. Kerzner, Project Management: A Systems Approach to Planning, Scheduling, and Controlling, Wiley, 10th ed., 2009.

[9] S. Mukhopadhyay, Project Scheduling and Management for Construction, RSMeans, 4th ed., 2018.

[10] D. Adams, Guida galattica per autostoppisti, Mondadori, 1979.

[11] D. Parmenter, Key Performance Indicators: Developing, Implementing, and Using Winning KPIs, Wiley, 4th ed., 2020.