UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTA' DI INGEGNERIA MECCANICA

Dipartimento di Ingegneria Industriale e Scienze Matematiche

Corso di Laurea magistrale in **Ingegneria Meccanica**

Gestione delle Incertezze del Sistema MRP con un Approccio di Programmazione e Inferenza di tipo Fuzzy

A Fuzzy Programming and Fuzzy Inference Approach for MRP System Uncertainties Management

Relatore:

Prof. Ing. **Filippo Emanuele Ciarapica**

Tesi di Laurea di:

Eros D'Addazio

A.A. **2021 / 2022**

Contents

Compendio 1

1. Introduction 4

2. AIDEAS: AI Driven Industrial Equipment Product Life Cycle Boosting Agility, Sustainability and Resilience 7

2.1 **Objectives of the AIDEAS solution** 9

3. Theoretical References and State of the Art: MPS, MRP and key factors 12

3.1 **MPS** 13

3.2 **MRP** 19

3.3 **Key factors in MRP management: the safety stock and the bullwhip effect** 21

4. Procurement management strategies: “stock” based and “requirements” based 25

4.1 **EOQ & EOI as “stock” based strategies.** 25

4.2 **MRP as “requirement” based strategy**. 28

4.2.1 **BOM “explosion” for requirements evaluation** 28

4.2.2 **Capacity requirements planning (CRP)** 30

4.2.3 **“ABC” analysis as strategic support for MRP** 31

4.3 **Alternative methods for cases with a higher degree of complexity** 32

4.3.1 **Multi-stage Stochastic Optimisation** 32

4.3.2 **DDMRP** 33

5. The sphere of artificial intelligence: a brief overview 38

5.1 **Machine learning** 39

5.2 **Deep learning: the neural networks** 42

5.3 **AI applications for forecasting** 44

5.3 **Digital Twin** 45

5.4 **Fuzzy Logic** 46

6. Fuzzy Logic Theory 50
6.1 **The "Fuzzy Set"** 51
6.2 **Fuzzification / Defuzzification** 55
6.3 **FIS and Rules (Mahmdani/Sugeno and IF/THEN)** 59
6.3.1 **FIS (fuzzy inference system)** 60
6.4 **Defuzzification** (meaning and methods) 63
6.4.1 **CoG (center of gravity defuzzification)** 64
6.5 **Expert Systems and Fuzzy Logic** (FES: fuzzy expert systems) 67
7. The use of AI in production and logistics management: how fuzzy logic could play a key role 70
8. IIRA Architecture (wb – 6) 76
9. Project Business Viewpoint 82
9.1 **Business Viewpoint: Stakeholders** 82
9.2 **Business Viewpoint: vision, values, key objectives, fundamental capabilities** 85
10. Project Usage Viewpoint 88
10.1 **The Roles**: 88
10.2 **Activity Definition** 90
11. Project Functional Viewpoint 92
11.1 **Generic Features** 92
11.1.1 **How to ensure the features** 93
11.2 **Algorithm conceptual task flow & Architecture for Runtime** 95
12. Project Implementation Viewpoint 99
12.1 **Proposed AIDEAS Solution Technical Features** 106
13. Project proposal 107
13.1 **Problem Model** 111

13.2 **Model Equations** ... 114

13.2.1 **Equations for the Deterministic model** ... 114

13.2.2 **Equations for the Fuzzy model** ... 117

13.2.3 **How to manage coefficients calculation** ... 118

14. The Algorithm ... 123

14.1 **Problem's assumptions** ... 123

14.2 **Code building** ... 129

14.3 **Elasticisation Coefficients** ... 131

14.4 **Planning Constraints:** ... 138

14.5 **The Solver** ... 138

14.6 **The Output** ... 139

14.7 **How does the algorithm actually work?** ... 142

14.8 **Results evaluation** ... 148

15. Conclusion and possible future developments ... 151

16. INDEX ... 154

16.1 **References** ... 154

16.2 **Web pages index** ... 163

16.3 **Figures Index** ... 165

16.4 **Tables Index** ... 168

16.5 **Equations Index** ... 168

Compendio

Il presente lavoro di tesi è, a tutti gli effetti, un report del progetto portato a termine in sede di tirocinio presso il dipartimento di Ingegneria Industriale dell’Università Politecnica di Valencia (CIGIP). Il tema dell’elaborato è relativo alle direttive del progetto di matrice Europea denominato “AIDEAS”, nonché un framework del programma “Horizon Europe”. Tale progetto ha come focus principale la ricerca di possibili impieghi e lo sviluppo di toolkit che, attraverso l’uso dell’intelligenza artificiale, possano fornire un “boost” nell’esecuzione di attività legate all’intero ciclo di vita di una specifica categoria di prodotti, quella dei macchinari industriali, ricercando oltretutto un miglioramento della sostenibilità delle stesse. Il progetto non si dedica quindi esclusivamente alle azioni legate all’uso del prodotto ma divide i suoi sforzi su 4 “suites” contenenti attività di varia natura: dal Design, alla sfera del Manufacturing & Use e la ricerca di metodi volti a migliorare le performance in termini di durata del ciclo vita. Obiettivo del tirocinio in sé non è stato quello di vagliare tutte le Suites collegate al progetto ma di concentrarsi su un quesito in particolare tra quelli proposti: implementare una soluzione che, tramite l’uso dell’AI, potesse efficientare le attività di procurement di materie prime e componentistica, in modo da garantire la corretta pianificazione dei fabbisogni aziendali.

A seguito di approfondite ricerche in letteratura e confronti con esperti dell’area gestionale, si è potuto constatare come lo svolgimento di quest’attività profondamente strategica venga svolta normalmente attraverso sistemi di tipo MRP (material requirement planning), incorporati sottoforma di algoritmi all’interno dei tradizionali software gestionali utilizzati nelle realtà produttive di qualunque dimensione e orientamento di mercato (noti come sistemi di resource enterprise planning o ERP). A conclusione di quest’analisi preliminare, si è optato nel trovare un metodo per poter ottimizzare i metodi gestionali attualmente in uso piuttosto che rimpiazzarli del tutto. L’idea è stata quella di progettare, tramite linguaggio di programmazione Python e con l’uso di librerie open-source, un algoritmo che impiegasse simultaneamente sia dei concetti di logica “Fuzzy” che ben si sposano con il fatto che, in fase di pianificazione, goals e vincoli vengano spesso espressi in maniera verbale, sia un approccio matematico deterministico di programmazione lineare. Il primo step è stato quello di simulare un database (tramite Excel) che contenesse tutti i dati necessari alla pianificazione di ordini di acquisto e\o produzione necessari per poter soddisfare la domanda di mercato di 4 Items fittizi (costruendo delle ipotetiche "bill of materials", cioè dei database di ordini previsti\ricevuti ecc.) per un periodo di tempo tra 4 e 6 mesi, poi si è passati a ri-progettare un set di equazioni e funzioni obiettivo tradizionali, che potessero emulare l'approccio che un'azienda utilizzerebbe nel ricercare una minimizzazione dei costi gestionali relativi alla pianificazione dei requisiti al fine di ottenere il valore delle principali variabili decisionali; il tutto, sottoponendo il piano ad una serie di vincoli di programmazione che condizionano il risultato delle variabili (un vincolo potrebbe essere inteso come la disponibilità di capacità produttiva o l'entità della domanda, ad esempio). In ultimo, dopo uno studio approfondito della matematica tipica della logica Fuzzy, il problema dell'MRP ha visto un restyling completo con un'architettura che permette di operare una programmazione in grado di incorporare il fattore di "incertezza" che, nelle reali condizioni di lavoro (di natura non-deterministica), condiziona profondamente le scelte del material planner dal momento che esisterebbe un errore di fondo nel considerare "deterministiche" variabili come la capacità di risorse produttive a disposizione dell'azienda, l'andamento della domanda di mercato, i lead times.

I risultati ottenuti, seppur derivanti da un caso applicativo completamente simulato e valutati in termini di "livello di servizio assicurato" sono piuttosto soddisfacenti e, in alcuni casi, migliori rispetto a quelli ottenuti (con medie superiori al 92%) sullo stesso caso studio ma con un approccio tradizionale, a dimostrazione del fatto che un sistema di pianificazione deterministico, seppur affidabile e robusto, non garantisca un match ideale tra l'aleatorietà dei driver/variabili tradizionali di un piano MRP e le scelte adottate dal material planner. Match che invece si trova andando a progettando un rilassamento dei vincoli di programmazione. Possibili sviluppi futuri sono da ricercare in primis in un ampliamento del discorso di gestione del "linguaggio naturale" da parte dei calcolatori, nonché principale focus della logica Fuzzy, poiché potrebbe aprire le porte ad una realtà in cui la pianificazione e la gestione dei fabbisogni venga direttamente modellata in funzione della percezione del planner circa i principali fenomeni che ne condizionano le scelte e, in secondo luogo, progettando dei modelli matematici che meglio approssimino il suo modello di ragionamento in fase di Decision-Making.

1. Introduction

Artificial intelligence and Industry 4.0, a "collaboration" with infinite potential?

The emblematic Industry 4.0 paradigm needs no special introduction since, nowadays, the media exposure to which it is subjected, combined with the innovations to which it has opened up, has allowed everyone to experience its effects more or less directly, generating such an impact as to earn the title of "fourth industrial revolution". Its significance is demonstrated by the strong interest aroused in the scientific community which, in addition to studying its properties under a mere technological aspect, has also been engaged for years in evaluating its impact in the social sphere (1)(2), given that it is been visibly shaken by this encounter-clash: basically it is a completely normal reaction, which occurred similarly during the previous revolutions due to the new skills needed to compete on the market, the new habits introduced and the need for industries to find an alignment with the new technological trends.

Even today various names and concepts are attributed to this industrial revolution as if they were synonyms, in fact it is not uncommon to see "smart-manufacturing", "fourth industrial revolution", "cyber-physical industry", "smart factory" (3) used in its place, creating confusion and discouraging the attribution of a single concept; on the other hand, the nuances that such a broad theme can assume are innumerable, above all if they vary according to the point of view of the various industries, but it is unanimously agreed that wanting to create a flexible and digitised production system that allows a direct link between the human sphere and that of machines to guarantee the creation of personalised products and services is its primary objective (4).

The fact that we also speak of "services" is not obvious, as the term "Industry" could refer exclusively to the manufacturing sector of the production of physical goods when in reality all other activities not strictly related to manufacturing in the strict sense are visibly influenced by the innovations introduced (5), and this means that a joint effort is required by all the players in a supply chain so that they can acquire the necessary skills to better adapt and keep up with this process of change. But how do we adapt to this evolving scenario? Here too Germany was the first to create a sort of "guideline" that could be a point of reference (4), built on some key points:

- need for implementation of cyber-physical systems: the goal is to broaden the use and potential of computerised control systems in order to create sophisticated machines that can interact with humans in the performance of their activities (6) (7), increasing their security, efficiency, and performance. This trend of massive digitisation has raised many issues in terms of cybersecurity, as the transmission of sensitive data becomes an extremely frequent operation and consequently has required major investments by organisations to be able to ensure safe completion of certain operations, for which they were previously lacking, or nearly lacking, effective "protection systems" (8). The health of workers, from a psychophysical point of view, is also an issue that remains central to this revolution: one of the most futuristic and successful attempts in this respect could be, for example, the introduction of Cobots in the workplace (9).

- use of new enabling technologies (10) and creation of smart services linked not only to production but to all the activities that revolve around it, so much so that in addition to Industry 4.0, today there is also talk of Logistics 4.0 (11).

- Improved connectivity: we refer both to the real-time connectivity between the actors of a value-chain so as to improve their cooperation and reduce their reaction time, and to the network of connection and data transmission that needs to establish between the various resources of a manufacturing plant. The need to generate enormous masses of data so that it can be analyzed and continuously receive feedback for monitoring a production process or service results in another of the significant challenges of Industry 4.0, that of Big Data Analytics (12), (13).

- use of new materials and resources, optimisation, and adaptation of current manufacturing processes in order to be able to reduce environmental impact (14) (15) while improving energetic effectiveness.

Until now, only characteristics intrinsic to the world of Industry4.0 have been discussed, but one aspect that catches the eye is how "digital" becomes the common thread among all the needs that the new vision of industry has brought with it, requiring that an evolution takes place not only at an infrastructural level (necessary as a matter of fact) but also at a philosophical level, since in order to be able to embrace a change of this magnitude it is first necessary to understand all its potential in such a way as to optimise the result of the efforts that they follow. Never before can artificial intelligence take on a leading role within this transition by driving its rhythms, so much so that the economic efforts of organisations and governments in this regard have become anything but trivial.

Therefore, since the project in question is characterised by a strong "European" matrix (as it was born on the initiative and funding of the European Commission) enriched by an "innovative" imprint (as it is directly linked to the themes of Industry4. 0 and AI) it would be interesting to know the strategies that are inspiring European countries with respect to the evolutionary phenomenon that is redesigning the structure of strategic areas for the community, such as energy production, transport, welfare, and of course industry and infrastructures, with consequent radical changes (16) (17). To remain on more "engineering" and design topics, aspects closer to the industrial world and the interaction between man and productive resources have been discussed so far.

This document is a detailed account of an internship carried out within the CIGIP (Centro de Investigación en Gestión e Ingeniería de Producción), a research center that belongs to the "UPV-Universitat Politècnica de València", which joined the European project named AIDEAS and offered the chance to work on some of its framework. It will be structured as follows:

- firstly, the AIDEAS Project’s purposes will be explained, and particular attention is taken above a specific framework of its numerous cases
- the next step will be an in-depth literature review for some Industrial Engineering themes and AI technologies
- an exhaustive explanation of Fuzzy theory with a focus on operations on fuzzy set, fuzzification and defuzzification methods and Fuzzy Inference System types
- an attempt to build the project’s IIRA Architecture based of its specific conventions
- lastly it will be a theoretical description of the problem to be solved by the project development, how the tool designed has been obtained and how its mechanisms works. Then an evaluation of results will be carried.

2. AIDEAS: AI Driven Industrial Equipment Product Life Cycle Boosting Agility, Sustainability and Resilience

1 - HORIZON Europe Logo

Before introducing the specifics of the project that this text is intended to describe, it is worthwhile to give an overview of the reasons that led to its development and the boundary conditions that shaped its profile. The 'AIDEAS' project is the offspring of what is one of the few research and development programs for which Europe has earmarked such substantial funds, we are talking about the 'Horizon Europe' project, successor to 'Horizon 2020' (18): it is a 7-year program (2021-2027) with a total budget of around EUR 95.5 billion with which Europe aims to achieve a number of important goals, ranging from tackling climate change to achieving sustainable development goals and increasing commercial and industrial competitiveness ('UN's Sustainable Development Goals') while at the same time attempting to respond to various global challenges. This should not only incentivise research, but also stimulate the economic growth of countries precisely by seeking industrial competitiveness. And it is against this background that the AIDEAS project comes into play (wb-1).

2 - AIDEAS Logo

AIDEAS stands for “AI Driven industrial Equipment Product life cycle boosting Agility, Sustainability and resilience”, it is a project designed to fill certain gaps in the industrial machinery manufacturing sector with regard to the use of Artificial Intelligence (AI) tools, not least the objective described in the official documentation framework reads as follows "AIDEAS Project will develop AI technologies for supporting the entire life cycle of industrial equipment (design, manufacturing, use and repair/reuse/recycle) as a strategic instrument to improve sustainability, agility and resilience of the European machinery manufacturing companies" (wb-2).

The European machinery industry has experienced unprofitable growth for several years due to a few difficult circumstances, including declining productivity, rising material costs and increasing product complexity. The COVID-19 pandemic exacerbated the sector's difficulties and revealed a lack of resilience, but nevertheless, the machinery manufacturing sector continues to be an important source of employment for people (about 3.2 million people employed (wb-2)) and of growth and prosperity for its host country (19). Many of the inputs that have triggered various developments in the industrial landscape in general can be traced back to this sector, in which Europe has a historically strategic position. A common aspect that distinguishes the actors in this sector has been the lack of responsiveness (due to the sudden change and the lack of resources to invest) in taking the step that the industry4.0 revolution 'imposes' in order to take part in this transformation process, namely the use of those technologies recognised as enabling. The price of being able to follow this path of evolution entails for companies the use of very different technologies and working methods compared to those they have been used to up to now, and in fact one consequence of the change has been precisely that of the obsolescence: machinery, plants, production and management methods that cannot be reconciled with the line dictated by Industry4.0, especially from the point of view of digitalisation. Therefore, AIDEAS's main objective is to develop digital support tools for the industrial machinery and equipment sector in order to provide methods that can guide the company in making more robust decisions regarding the entire life cycle of the products and equipment in question, so that it can aim to increase the resilience, sustainability and agility of the supply chain; the latter understood as the acquisition of a chameleon-like ability to adapt to technological and market developments. In particular, there are four areas to which the project will devote its efforts (wb-2):

1) **Design**: AI technologies, integrated with CAD/CAM/CAE systems, for optimizing the design of industrial equipment structural components, mechanisms and control components.

2) **Manufacturing**: AI technologies for industrial equipment purchased components selection and procurement, manufactured parts processes optimisation, operations sequencing, quality control and customisation.

3) **Use**: AI technologies with added value for the industrial equipment user, providing enhanced support for installation and initial calibration, production, quality assurance and predictive maintenance for working on optimal conditions.

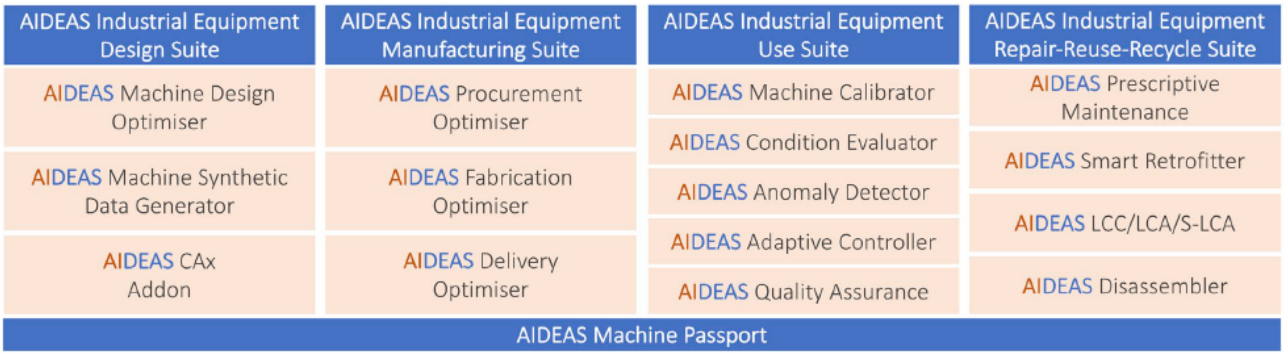
4) **Repair-Reuse-Recycle**: AI technologies for extending the useful life of machines through prescriptive maintenance (repair), facilitating a second life for machines through a smart retrofitting (reuse) and identification of the most sustainable end-of-life (recycle).

The AIDEAS is a project through which the academic world seeks to support the work of the industrial sector to provide solutions to drive the transition towards Industry 4.0. Therefore, in most cases, the solutions that will be developed or proposed will be entirely designed to adhere to real case studies based on both historical and current data of some of the companies involved in the project, on the one hand to favor their development and, on the other, in order to make sure that they benefit from the implemented solutions.

2.1 **Objectives of the AIDEAS solution**

The object of this study will be the search for methods to be able to fulfil one of the demands of the AIDEAS project and the attempt to realise a concrete solution that can be implemented and updated with future features. The main target is to design the structure of a toolkit that can assist company "X" in the management of its MRP (material requirement planning), facilitating the execution of all related operations and exploiting artificial intelligence as a carrier technology. The structure of the project is strongly characterised by the MRP management mode of those companies that will make use of the toolkit (whether they use a make to stock, make to order, assemble to order strategies), by how many stakeholders are involved in the supply chain in question (we are talking about manufacturers, suppliers, transport and stocking company managers), by the nature of the company (it may be a producer and at the same time a supplier for another entity in the supply chain). So, since the details previously highlighted, the solution may allow a high grade of personaliation to meet the organisation’s specific requirements.

An overview of the project’s key characteristics is exploited here but a detailed description of its function will be provided in the IIRA architecture description, after some theoretical explanation and a literature reviewing of the context of application, which are essential to justify the taken choices.



3 - AIDEAS Machine Passport

4- AIDEAS procurement optimiser

Of the 4 AIDEAS suites developed, this project aims to cover the needs of one of the 15 related projects, specifically the one called "AIDEAS Procurement Optimiser"

The driving idea that motivates the essential features of the proposed solution is to provide a method of optimizing the management of material requirement planning (MRP) that can enable those involved in this planning activity to better interface with the high variability of the data that constitute the planning input. Being able to introduce the factor "uncertainty" within a planning method such as this is neither simple nor immediate but, however, it allows the evaluation of alternative scenarios that do not necessarily have to be considered as a reference but can be seen as an object of comparison with the planning output obtained by alternative methods utilised by the company.

The solution in question consists of an algorithm that exploits linear programming contextually with fuzzy logic; the idea behind it is to provide a method to manage uncertainty (that could have a very large number of different sources) by an inference engine that by the use of fuzzy logic could take control of the problem's “elasticity”: this conceptual way to look at the problem is justified by the fact that some strategic data used during the planning couldn’t be known precisely since they are subjected to multiple fluctuations but it could be accepted think to them as capable to assume different values in respect to well know intervals of variation. In the light of this, it could be assumed that those data are not precisely known during the planning as, instead, the uncertainty linked to their fluctuation is. Thus, the algorithm aims to establish, within a certain planning horizon, what is the value of certain decision variables (considering the traditional outputs of MRP planning) that could create a reduction in operating costs. Cost reduction is seen as the main driver of planning. A further idea would be to embed the algorithm within an expert system that can encapsulate that knowledge necessary not only to suggest to the user what data constitutes the input to the scheduling but also how, if at all, to manage and modify the parameters of the algorithm itself according to the application context.

3. Theoretical References and State of the Art: MPS, MRP and key factors

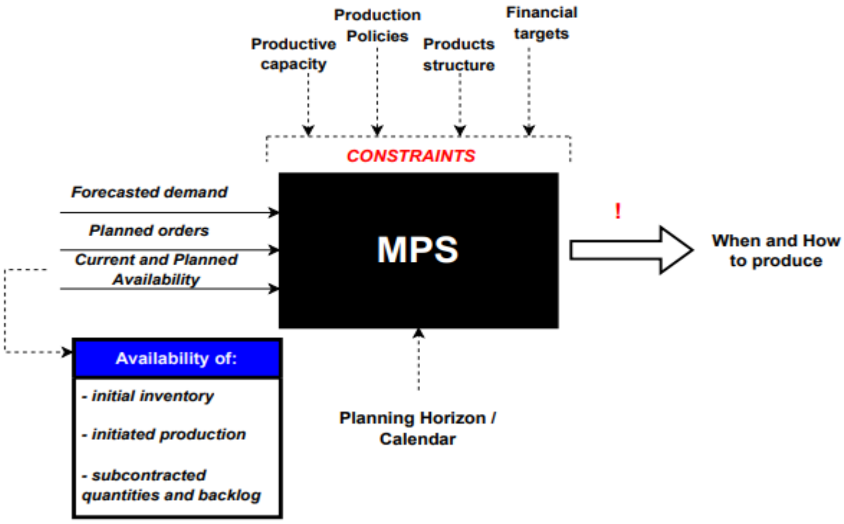
The information described in the current section is adapted from university teaching material (20).

Observing production management, using a top-down approach, one could imagine it as a pyramidal structure of business plans that flow into each other with a chain mechanism, generating step by step an increase in the degree of detail of the information and the reduction of the coverage period. Beginning with the most generic of the company's strategic plans, that draws its economic and financial objectives over a broad spectrum, this will go on to determine the architecture of what is known as the 'aggregate production plan': that's the one intended to meet market demands within an annual time window and to translate the sales plan designed by the organisation's top management in such a way as to guarantee the production capacity required to meet the matching point. In a nutshell, then, the purpose of the Aggregate Plan is to reconcile production capacity requirements (traceable to market demand) with actual productive capacity availability (expressed in man-hours or machine hours). Once a set of operational levers has been defined (ranging from the production rate, the level of stock in the assessment period, the availability of labour force and the willingness to take precautionary measures such as overtime hours or subcontracting), scheduling methods (such as Bowman's) are used to minimise the total costs of producing finished products by seeking the most efficient allocation of resources and, in combination with these, also PSH rules (production switching heuristic) or Magee's model are used with the same purposes.

The output of the aggregate plan is transformed into the input of the MPS which, for its part, covers a limited time horizon (3-4 months) and has as its object the individual finished products. In particular, the MPS aims to size production batches by 'exploding' the information from the plan that precedes it. A very high number of factors undermines the stability of the MPS output since the production environment is particularly stochastic and far from deterministic and has an indubitable importance in the successive detailed planning of material requirements and scheduling of production capacity. The last two issues just cited lead us to the MRP (material requirement planning) as well as the protagonist of this discussion and first responsible to the allocation of the resources available to the production of the various items in terms of raw materials and components generally (the Final Assembly Schedule "FAS"). Scheduling, revision and management of the master production schedule together constitute a highly articulated process that is strategic in manufacturing companies, with the aim of guaranteeing the production of those finished products that have been specifically commissioned, on time and in the required quantities, while also seeking to minimise costs (thus allowing for choosing the correct production strategy and the sizing of production batches given the imposition of a certain number of boundary conditions). This function is performed by looking at the best possible match between the production capacity that can be allocated in a specific time interval and the finished product demand that the market requires for the same period. Given its fluctuations, market demand can significantly interfere with the plans of the company that doesn't necessarily always have the production capacity necessary to satisfy it; edge cases could arise when this capacity is fully saturated or, conversely, under-used and both could generate supply chain disruptions and potential damage to the company's cash flow depending on the entity. The situation could be further complicated by the disorganisation in the management of material stocks (and this could lead the company to delay the production of some batches, generating dissatisfaction in its supply chain) and the failure to optimise the production process; a set of actions which, if optimised and used as drivers for improvement, would contribute significantly to the success of production. We could therefore define the MPS plan as a bridge that connects the company's purchasing area with the production area, as the former must ensure that orders for raw materials and components are processed to ensure that confirmed orders and finished product plans are produced correctly. The product code of the raw materials and components can be found in the BOM of the items in production (bill of materials) and, by integrating the data from this with the actual availability in the warehouse, a dataset is obtained and it practically constitutes the first input for the MRP which then will generate purchase orders once the MPS has analyzed and approved the production ones (knowing what the production capacity is at the time of the evaluation): in the light of those information it is not a mistake to define the MPS as the main driver of the MRP.

3.1 **MPS**

Quoting the MPS again, to get an idea of the quantity and origin of the data flowing into it, we could imagine it as a black box that receive a flow of information made by planned and confirmed customer orders, orders currently in production, forecast demand (the using of forecasting algorithm is discussed later), availability in the warehouse and, on the production side, it is important to know the characteristics of the individual item (BOM and the optimal sequence of production tasks that minimise the total execution time), the production capacity over the planning horizon (usually an interval with an extension of four/six months is considered) and the work schedule.



5 - MPS Scheme by optiptoerp.com

We talk about the planning horizon with reference to the period the MPS cover by guaranteeing the completion of the order. Must be specified that it is an intermediate planning between the other two, because basically there are 3 plans based on the level of detail of the information that populate them and the extension of the reference period:

1. The "**Aggregate Production Plan**": covers a rather extended time horizon (1 year) and has as its planning object very aggregated quantitative parameters (number of parts that can be produced, tons of finished product, etc., at an overall level or by product families). The purpose of the Aggregate Plan is to reconcile production capacity requirements (traceable to market demand) with actual availability (expressed in man-hours or machine hours).

2. The "**Master Production Schedule (MPS)**"

3. The “**Final Assembly Schedule (FAS)**”: covers a very short time horizon (1 month) and has as its object the assembly-configuration of the finished product. A typical task of the Final Plan is the choice of how orders are to be fulfilled.

The fundamental characteristic of the entire production planning process is its cyclicality, whereby the same period is the subject of several successive plans with an increasing degree of definition. Having reached this point, there is no need to specify how easy it can be to make mistakes by 'manually' managing such large quantities of data that such extended time horizons can be planned, nor how expensive these can be for the company, relative to their magnitude. Given that modern supply chains continue to evolve, becoming more and more intricate and generating complex dependencies (even for small and medium-size companies), the fact that people have to be responsible for ensuring their very subtle balance is enough to justify why there is an increasing move towards entrusting such operations to artificial intelligence systems cause they can be used to solve critical issues that undermine the sustainability of a supply chain (optimisation of utilisation and resources with reduction of waste, logistics, order scheduling) and to reduce both time and effort with the execution of those tasks. Once the characteristics of an MPS are outlined, it is necessary to know what its management strategy will be. The intrinsic characteristics of the type of product, its structure, the category to which it belongs, and the type of processes involved in its realisation condition plan management. It is not just a question of defining the strategy for production, whether Push or Pull type, but one must understand what will be the one used for the entire planning because it conditions a rather long-time window.

Today, in most production realities, the aim is to minimise stockholding costs and thus reduce the risk of remaining with unsold lots but, at the same time, maximise finished products variants; this normally goes along with the "Pull" type of production philosophy that, unlike the "Push" one, seeks to reduce as far as possible the need to anticipate the entry of materials into the warehouse (cause they generate either stockholding and maintenance cost proportionals to the duration of the stationary positioning in warehouse). If it is not necessary to play anticipation, then we could say that in such a system the effort to make accurate forecasts of future demand and production capacity is not even necessary. But in reality, it is customary to encounter very different problems: speaking of 'pure' systems of the pull type is not possible in today's context because the delivery times imposed by the market do not allow companies to work with this strategy. It is much more likely to find blended push-pull systems since the order portfolio is not made up exclusively of the ones formalised by the customer but is also enriched by those obtained as output from market forecasts (there are orders already processed and more or less extensive forecasts that determine the planning of how much, when and what to produce). Forecasts made on finished product sales inevitably affect the changes to be made in the production plans of all suppliers involved in the production of the individual item. The longer the forecasting horizon, the smaller, given the same forecasting tool used, is the goodness of the forecast: the smaller the window, the lower the probability of making a forecasting error. If we were working with forecasting algorithms which, with well-defined time steps, receive input datasets to predict forecasts, then we need to create a bridge between those estimates and the MPS, in order to automate the latter's update and make visible the modification to the subjects who carry out activities directly affected by this new alteration within the supply chain. At this point, we can say that the specific product category could directly conditionate the relationship between PLT (production lead time) and DLT (delivery lead time) and together with its demand are the major drivers influencing the MPS management strategy.

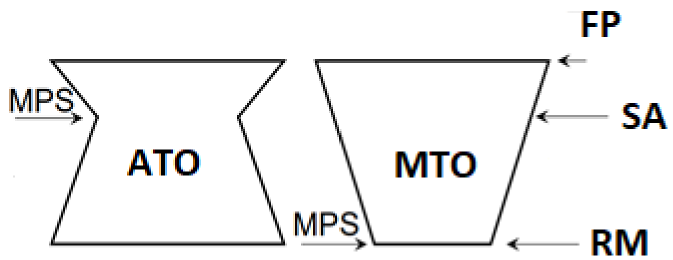
Referring to the scope of the project, which is aimed at the manufacturing context for the production of industrial machines, we can make an analysis of the particular scenery and product category. Normally, companies that do this type of production have a portfolio of finished products that is not very large, i.e., they produce products that do not have an exaggerated number of variants as these are limited at the top, while the BOMs (bills of materials) are very articulated. They are a category of products distinguished for having many sub-assemblies that can be mounted the same on the different variants (think for example of basic sensors, motor assemblies or small metallic parts, so their demand will be conditioned by the total demand for finished product received from the market) and other components for which the possibility of customisation is left to the customer (obviously in this case the customer has decision-making power over items that are more like optional). The manufacturing company from where the industrial machines shipped to the customer comes mainly operates with assembly operations carried out on pre-assemblies items. Assuming that the variety of finished products is not very high, MTO (make to order) and ATO (assemble to order) production strategies could be evaluated.

In the case of the MTO, the possible management scenarios are reduced to two:

1- DLT > PLT (ratio >1): then we can afford to make forecasts directly at the level of the finished product

2- the ratio is < 1: then the use of forecasting tools becomes indispensable to meet market requirements

Obviously, this is a generalisation since a company will not always have products for which only condition 1 or only condition 2 applies. The mixes that could occur in reality are certainly more complex.



6 - Assemble to Order / Make to Order strategies.

The picture [5] posed for full demonstration purposes shows the substantial differences between the two strategies. In its simplicity it is illustrative in showing how in the MTO case MPS planning is done at the raw material level (RM), while in the ATO case at the sub-assembly level (SA).

If the MPS is fed with highly reliable forecasts, it can minimise inventory costs while ensuring a high level of service for its customers. A revision is necessary whenever the forecasts are updated and new orders are received: in these cases, the time-phased record is one of the most frequently used techniques to revise the plan, because if a number of orders higher than what the production capacity can carry out is accepted, additional costs are incurred but they could be prevented in some manner. The time-phased record allows production to be planned in this way:

1- at the beginning there is a ready MPS program for a certain time window

2- for that window there are four variables that we must consider for each week: confirmed orders, planned orders, available products and quantity of products to be manufactured

3- therefore, given the forecasts and confirmed orders, it is known what the customer's request to be fulfilled is, therefore production planning will also take place based on the availability of products in the warehouse for that same week

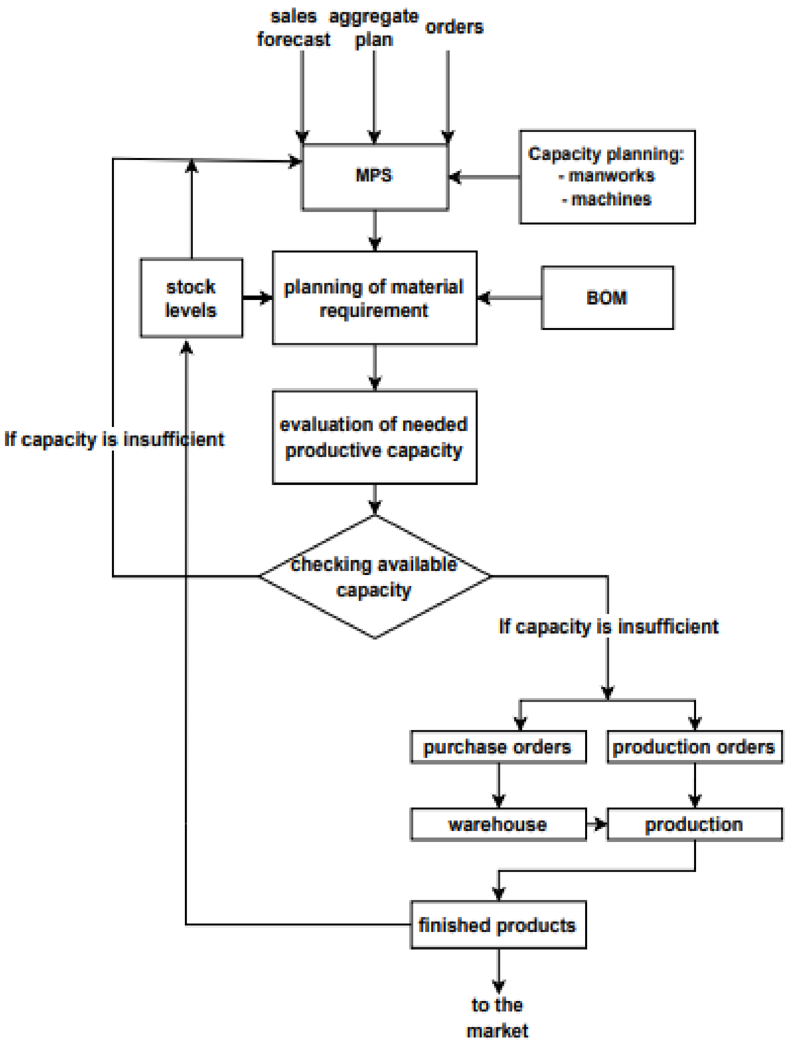
4- if there is a fluctuation in demand and consequently an alteration both in the number of orders and in the size of the forecasts, then it must be possible to review the plan (perhaps automatically by entering the new data on the variation in the software) and reprogram the quantity to be to produce

If there were certainty about the accuracy of the forecasts (absolute certainty does not exist), it would be easier to think of using specific management techniques: for example, if it was possible to evaluate the trend in demand in the coming weeks with a certain precision, it would be possible to study the oscillation pattern and understand that in the first X weeks from now there is a certain relatively low demand but that it grows from a certain point on (perhaps because the product has a certain seasonality or there have been changes in the market) and remains at those levels for Y weeks. Under these conditions, a possible reaction could be to produce batches of constant size in both weeks X and Y so as not to alter the overall production capacity: then, to date, production would be higher than necessary but maintaining the same rate of production, sufficient inventories would be generated to satisfy the demand for the product when it will be higher thus the output of the MPS concerning "how much to produce" is not enough to be able to plan production capacity. In order to be able to translate this data into requirements for the production environment, there are multiple applications, such as Rought Cut Capacity Planning. The RCCP method makes it possible to avoid day-by-day planning and to prevent inconvenient events that can lead to the interruption of production (e.g. a lack of sufficient capacity), so in broad terms it makes it possible to verify that sufficient capacity is available to meet the requirements of the MPS. In the event that the verification does not lead to positive results, then action could be taken by taking a step back and taking up the MPS plan again: if, for example, the default is generated by a lack of production capacity due to the size of orders, then one response could be to add overtime or outsource production batches to third parties. All this is always aimed at maintaining a certain level of service to the customer.

3.2 **MRP**

Referring to what is written in (21) , the definition that can be adopted for the Master Production Schedule (MRP) is: "a set of techniques that uses BOM data, inventory data and the Master Production Schedule to calculate material requirements recommended to release material replenishment orders". On the other hand, we traditionally consider as 5 the fundamental inputs of this plan: Master Production Schedule (MPS), Bill of Material, item master, orders, requirements (22). From what one reads is simple to perceive how strategically relevant MRP can be within the organisation, since its ability to meet market demands while ensuring a certain level of service is determined by the meticulous management of components and raw material availability and the control of the productive capacity during a certain time window. Although this planning is associated with a push-type management philosophy, it is still widely used today albeit contaminated by more 'lean' objectives and pull strategy actions (this underlines the fact that modern production environments have adapted to using a more blended management mode), one of them being the reduction of product units in warehouses or in process in the plant. The costs of management, stockholding, and the risks related to the obsolescence problem are not only attributable to the finished product units that the company has in its warehouse, but also the individual components, semi-finished products and raw materials utilised for their construction. The ideal condition that organisations would like to have been the minimisation not only of finished products levels in stock but also the one of items used in their production leading to a reduction in the associated costs and the likelihood of having unsold items. The problems associated with this perspective have already been introduced earlier and also pointed out the need to use forecasting tools to be able to play in advance the entry of certain product categories into one's own warehouse, given the impossibility of acquiring them once the confirmation of the order from the customer has been received.

Over the years, many researchers have tried to find methods to make the MRP architecture more robust through the using of more effective forecasting tools, but the lack of determinism in combination with the highly aleatory nature of forecasts contribute to making the approach not always 'safe' especially in particular production contexts. Significant studies on the uncertainty sources in material requirement planning management have been achieved by many researchers (23) but they have usually hypothesised as deterministic the production environment, and this doesn't find consistency with the structure of modern supply chains, where there are many sources and a complex "multi-echelon" structure, since they are built with multiple stocking points (usually with different items for each one) and nodes. As reported in (24) it is precisely the uncertainty related to the veracity of the forecast that makes traditional MRP management methods not 100% effective. The study also focused on understanding how the effect of varying parameters such as safety stock and lead times, re-planning frequencies, frozen periods could protect against demand variability. Other research such as that of (25) and (26) focused on investigating the effects caused by the choice of whether to operate safety stock management on finished products or on the components used in their manufacture, respectively. Having reached this point, it is easy to realise how many factors simultaneously contribute to the instability of the MRP, but also how important it is to intervene in research in order to increase its reliability. The underlying theory of MRP inevitably recalls the "stock management techniques" since it plays a fundamental role in stock control and flow of materials in the context of large production plants. It requires ad hoc management methods since it is not so rare to work without a single warehouse for raw materials and components in a real production environment, but it is more realistic to find multiple intermediate structures downstream of the machining centers (so they are smaller warehouses for semi-finished products) having a content capable to determines the level of work for the successive centres, and thus realise an equilibrium being as much more efficient in the productive sense as it is unstable. Using a pull management strategy, the intermediate subjects of the production chain do not know the value of the demand for the finished product, but work to restore the level of the immediately downstream warehouse above a certain threshold. When the commercial office receives an order for a certain quantity of items, it triggers a mechanism by which they are taken from the warehouse of finished products and "pulling" the production chain. In push management, on the other hand, everyone in the production system works 'knowing' the value of the demand received or, in any case, the demand forecast and evaluated through forecasting systems; this means that there is no actual production activation order. Hypothetically, intermediate warehouses would not be needed here since everyone knows with certainty what and how much to produce, while in the first case where we speak of warehouse management, the intermediate subject of the production line does not need to know what the various lead times are, but only has to check the next warehouse level to proceed with the start of activities.



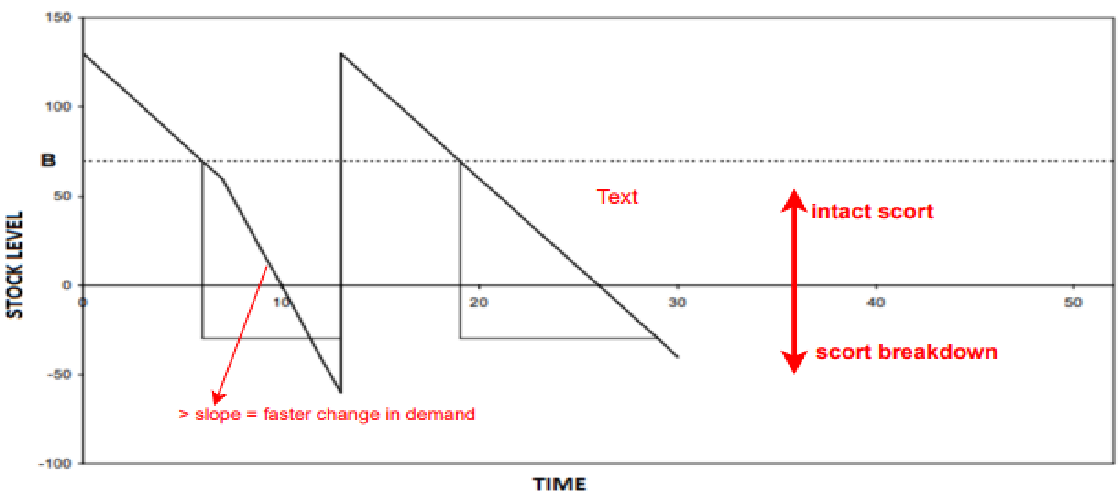
7 - MRP logic scheme

3.3 **Key factors in MRP management: the safety stock and the bullwhip effect**

The decision to discuss these two aspects jointly lies in the fact that they have a kind of mutually influencing capacity.

The issue of security escorting is a topic that provokes many debates among industry experts involved in optimizing management processes in supply chains, and this is evidenced by the fact that among the various existing production management methods they always, or almost always, play a key role. Research has always focused quite intensively on studying solutions for the problem of sizing, positioning and management of safety stocks (27), as they are seen as one of the principal ways of ensuring the fulfilment of market demand when this varies from the expected size. On the other hand, it is unreasonable to think that stocks alone can be entrusted with the task of managing the fluctuations and intrinsic properties of the modern market: the possibility given to consumers to have access to increasingly personalised products has fueled the number of variants of products/materials/components in circulation in the production process, so much so that the need has arisen to have expert figures for the sole management of warehouse and material flow problems since the risks that can arise from a wrong strategy are those of having large quantities of unsold products or products insufficient to satisfy market demand, respectively determining the two extreme cases of overstock and stock-out (28)(25).

The concept of "uncertainty" associated with the large number of variables that jointly can affect supply chain and market characteristics has always been considered as the major driver in determining safety stock dimensions and management choices adopted as a countermeasure to this issue (29). To emphasise how intricate and numerous the risks for a supply chain loaded with 'uncertainty' can be, one can cite a branch of applied research that was created precisely to find new solutions for these, the 'supply chain risk management' (SCRM) (30). Designing methods for correct safety stock sizing is one of the best-topics in the supply-chain world, in the article (31) and in (32) is given, respectively, an overview of research methods for the determination of safety stock properties and the study of the topic focused on production environments where MRP is used.



8 - safety stock effect

The most traditional method for determining the size of the safety stock of a certain item requires first of all that the level of customer service to be ensured is determined, i.e., the percentage of cases the stock in the warehouse must be able to cope with changes in demand and/or lead time. Regarding to the last two factors mentioned, it is assumed that these can vary according to a "normal" distribution law. If, for example, we were to look only at the case where it is the demand that varies but the LT remains constant, we have the situation explained above.

The consumption of product during the lead time is a variable constructed as a sum of identically distributed normal variables LT (demand, e.g., daily or weekly). The withdrawal of product from the warehouse during the lead time is then defined as:

$$D_{LT} \Rightarrow N(LT \cdot \overline{D}; LT \cdot \sigma_D^2)$$

Equation I - withdrawal of product

Associating the product consumption during the lead time with the normalised variable Z (the standard normal curve Z has a normal trend and mean and variance value equal to 1 and 0 respectively), we will have:

$$z = \frac{D_{LT} - \overline{D} \cdot LT}{\sigma_D \cdot \sqrt{LT}}$$

Equation II - normalised variable Z

from which we can derive the value of the reorder point as follows:

$$D_{LT} = \overline{D} \cdot LT + z_{LS} \cdot \sigma_D \cdot \sqrt{LT}$$

Equation III - Reorder point

By choosing the level of service (e.g. 95%), the tables of the standard normal curve give the value of Z_{LS}, i.e. the value of the standard normal variable corresponds to a probability equal to the chosen level of service (the probability of ensuring in time the requested demand of products made by the clients). At this point, we can understand that the safety stock value (ss) corresponds to that of the following equation:

$$ss = Z_{LS} * \sigma_D * \sqrt{LT}$$

Equation IV - safety stock

Turning now to the concept of the "bullwhip effect", and how this may relate to the security stock effect, it can be said that: the bullwhip effect, also known as demand information amplification or the Forrester effect, is a phenomenon whereby a small variation in end-customer demand leads to a significant fluctuation in orders that the upstream supplier receives in the supply chain system (33). The fundamental causes of the phenomenon can be attributed to the error that the subjects of a supply chain (customers, suppliers, manufacturers, salespeople) make in assessing the trend of demand using forecasting systems; all of them cannot have visibility of the demand for the entire extension of the supply chain but, at the same time, a mistake made in their evaluation can have repercussions that are amplified along its entire length. Although the phenomena connected to demand are certainly the most impactful, fluctuations and interruptions of various kinds can be generated in each node of a supply chain:

- variations in costs that determine a change in the trend of purchases
- logistical delays leading to production disruptions
- communication errors between components of the chain
- incorrect evaluation of lead times

The Bullwhip effect, regardless of how it manifests itself, generates economic outlays. With reference to the production environment, an excess of products would turn into waste and unsold products, while a lack of them would cause chain delays with customers and suppliers. The use of adequate forecasting systems and the correct sizing of safety stocks can significantly reduce the possibility of manifestation of the phenomenon that is extremely sensitive to the quality of the MRP management (in an indirect way).

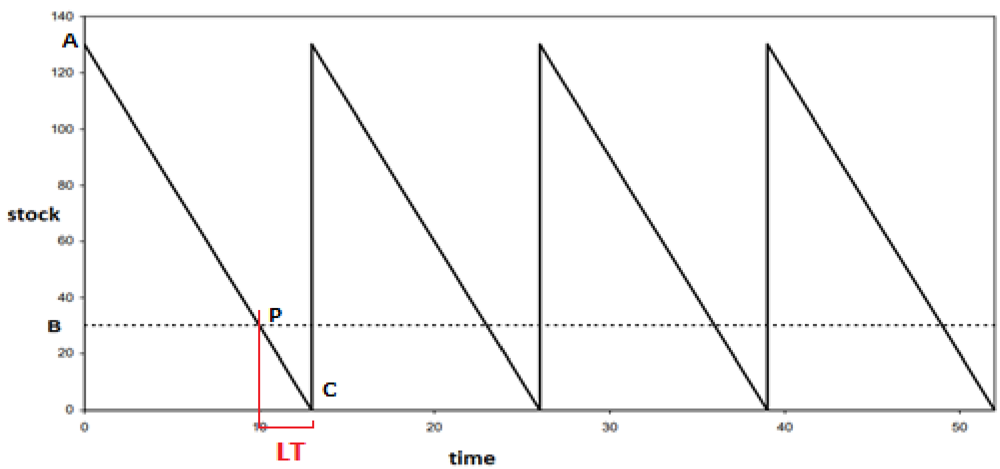
4. Procurement management strategies: “stock” based, and “requirements” based

Although the MRP management model can be traced back to a "requirement" type procurement management method, it is also necessary to know other strategies to understand how these combine to create more intricate models in modern production environments.

4.1 **EOQ & EOI as “stock” based strategies.**

- **Reorder point strategy**: production (or the order of an article) is only triggered when the respective stock has fallen below a certain level so as to avoid stock breakage situations. It turns out to be the least economical control method because it is difficult to implement a system that keeps track of stock. For this technique, an index named EOQ is introduced, allowing us to find the constant quantity to be reordered each time it falls below a certain limit, minimizing management costs.

- **Reorder period strategy**: the acquisition of material takes place at regular intervals, but as the consumption of stock is not regular (due to the variability of orders), this can lead to stock breakage situations. By periodically checking the stock level to reorder only the quantity necessary to restore the designed stock is risky because of not knowing what happens between one level of restoration and the next.



9 - EOQ strategy

If at the point “P” where the dotted line from "B" intersects the descending line $\overline{AC}$ a perpendicular is drawn on the X axis then the distance from the point of intersection to the next vertical line is the lead time: when reaching the crossroads an order is issued, but arriving at the vertical instead would mean that the order is available; therefore the lead time is the time between the issue of the order and its arrival.

This model provides us with what is the constant quantity to be ordered allowing management costs minimisation on an annual basis:

$$TC = P \cdot D + C \cdot D/Q + P \cdot F \cdot Q/2$$

Equation V - Total Cost

The first term indicates the purchase cost, the second term indicates the cost of issuing orders and the third term indicates the cost of maintaining stock.

P denotes the unit purchase price, D is the quantity ordered in a year, F is the unit stock price, Q/2 the average stock, C is the cost of issuing orders, D/Q is the number of orders placed.

Deriving with respect to Q we obtain:

$$EOQ = \sqrt{(2 * C * D)/(P * F)}$$

Equation VI - EOQ

at this point, we easily derive the reorder level as B:

$$B = (D * LT)/N$$

Equation VII - reorder level

The one described above is probably the most traditional way of planning the size of batches of products to be ordered as a function of certain factors and making possible to find the quantity that minimises the annual running costs, valid and under conditions of:

- Continuous stock control

- Constant order quantity

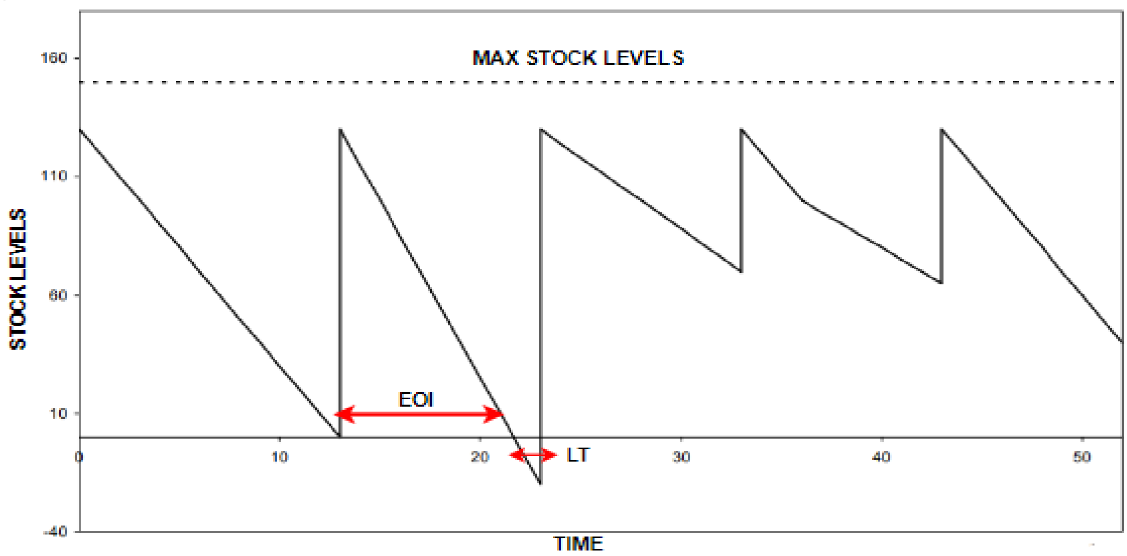
- Variable time between two successive orders

This basic model can be enriched with increasingly stringent constraints, resulting in other models such as "EOQ with variable purchase prices", "EOQ back order" assuming the possibility of extending delivery times, and the "EOQv".

The EOI model, used as another stock management technique, determines the amount of time that elapses between the issue of two successive purchase orders, operating under conditions of:

- Periodic check on stock
- Variable ordered quantity
- Time between two successive orders constant

The model determines the time interval between two orders that minimises management costs on an annual basis. By operating in these conditions, a quantity of items will be ordered that is sufficient only to restore, up to a certain level, the extent of warehouse stocks, trying to minimise annual management costs.



10- EOI strategy

4.2 **MRP as “requirement” based strategy**.

Theoretically, a production environment that uses MRP is typically characterised by a push type perspective (use of forecasts) with the entry of materials into the production line anticipated with respect to the actual customer demand.

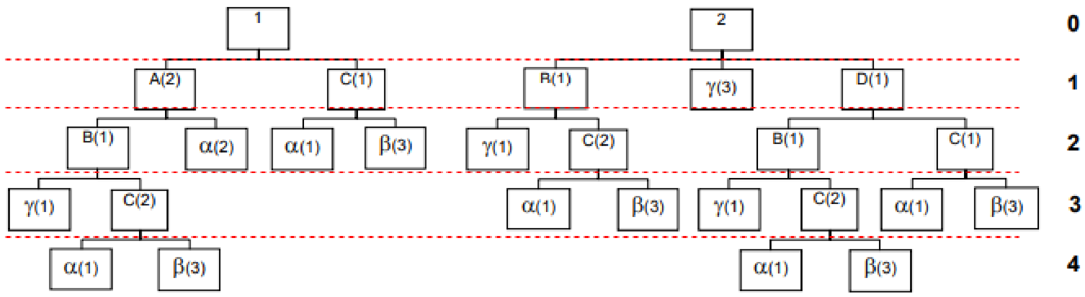
Unlike stock management (pull environment) for which each activity has visibility only of the upstream and downstream quantities in the intermediate storage buffers, here instead, the whole supply chain is fully aware of the actual input demand, so the management of the various activity follows different paths.

This type of management goes well with products whose demand is difficult to predict or highly variable, products that have a very complex bill of materials and whose unit economic value is high, therefore keeping them in stock could be very expensive as even in the case of companies that have long procurement lead times. A product description that mirrors, for example, that of an industrial machinery or equipment. The MRP (born as an American response to the Japanese just in time) has the principal objective of keeping inventory levels under control and planning the acquisition of raw materials and components together with the production of parts when there is an effective request of the same; MRP usually uses the upfront production of versatile functional assemblies (sub-assemblies that might be the same on different finished product models) so they could be assembled to obtain the final product when needed; this would make it possible to avoid risky capital investments.

4.2.1 **BOM “explosion” for requirements evaluation**

A case study is presented to give an idea of how needs are calculated in an MRP programming. Assuming the case of a company which has planned the production of two finished products (with identifiers "1" and "2" and unitary request) it is highly probable that these incorporate a certain quantity of identical items (just think of even of small metal parts, electronic wiring, sensors) therefore in planning the requirement of raw materials it is necessary to consider all the products that are found at level 0 of the BOM and the quantity of items required on each of the subsequent levels. The BOMs represented in the image [11] shows the identifier of the subassembly or components (α, β, γ etc.) and in brackets the number of units necessary for the realisation of the level where they are reported in; as far as the content of the two matrices is concerned, the first

([B]) shows the coefficients of use of the elements at the different levels of the bill of materials, while the second ([I-B]) is the result of subtraction from the identity matrix of order 9 of the first one.



	1	2	A	D	B	C	γ	α	β
1			2			1			
2				1	1		3		
A					1			2	
D					2	1			
B						2	1		
C								1	3
γ									
α									
β									

	1	2	A	D	B	C	γ	α	β
1	1		-2			-1			
2		1		-1	-1		-3		
A			1		-1			-2	
D				1	-2	-1			
B					1	-2	-1		
C						1		-1	-3
γ							1		
α								1	
β									1

11- BOM and requirement matrix

The key to understand the diagram, with reference to the same image, is that for which 2 units of the item with identification code "A" are needed (in turn "A" is a functional group of several items) to produce a finished product with identifier "1". By figuratively placing the BOM diagram of the two finished articles on level 0 side by side and "exploding" their bill of materials for all subsequent levels, we proceed with a series of calculations:

1) The first step (matrix [I-B]) constitutes, as already underlined, the difference between the first matrix and that identity of the same order.

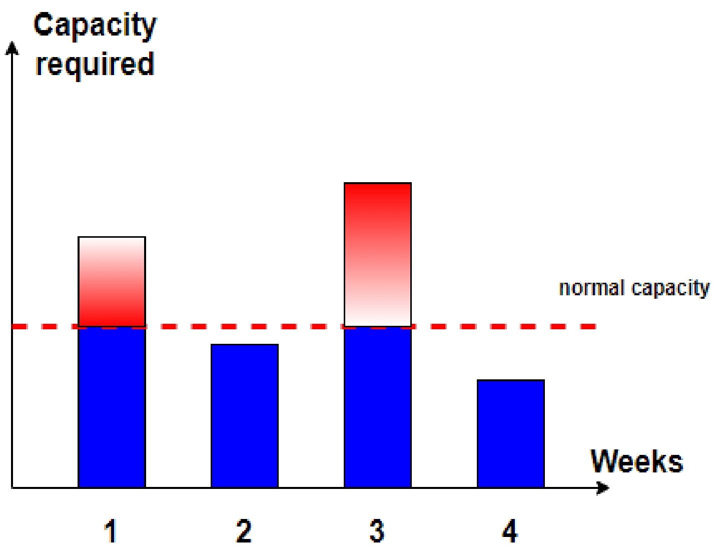
2) A matrix inversion operation is applied to the obtained matrix (obtaining a further matrix [I-B]-1)

3) The vector of finished product requests [d] is then calculated (in the example considered unitary for both item "1" and "2")

5) The last step involves calculating the product between the vector of the previous step and the matrix of step 2: this would allow to obtain the vector of needs [x] for the articles considered ([x] = [d] *[I-B]-1)

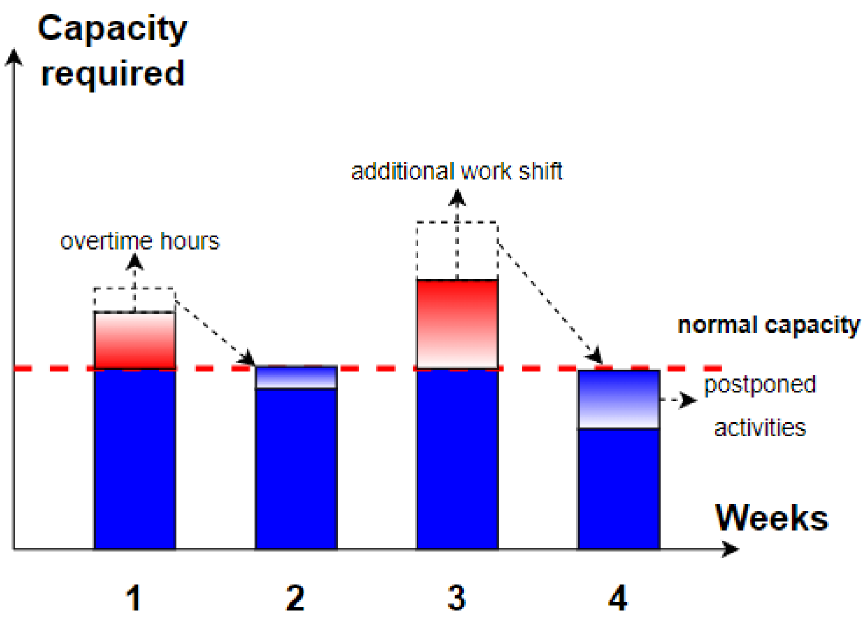
4.2.2 **Capacity requirements planning (CRP)**

Once the MRP provides its result, it is necessary to validate the feasibility of the same in terms of realisation, i.e. it is necessary to verify that there is sufficient production capacity to be able to satisfy the production demand (assessment of the load profiles of the work centres, time actually available to production in a certain interval, number of hours assigned to be able to carry out a certain activity, scheduled maintenance interventions, etc.) otherwise corrective actions would be triggered to fill any gap.



12 - CRP before corrective actions

Knowing, for example, on the one hand, what is the number of operators in a certain department available to carry out a production step and, on the other hand, knowing the efficiency of the i-th work centre together with the actual number of hours expendable for production (excluding the hours for meals and daily maintenance/cleaning) a calculation is made of the total hours available for processing in a certain period of time, for example a week; this amount is compared with the total hours actually necessary to carry out the production of a certain article (calculated according to the efficiency and type of machines/equipment of the plant) and countermeasures are adopted and based on this difference (as it could be either positive or negative). The reaction strategies could be many, from the introduction of overtime work shifts (depending on the extent of this lack of capacity over time) to the evaluation of the introduction of a new workforce or investment in new production units in the plant but also resort to subcontracting or a reorganisation of the MPS. A less drastic solution would be to spread any surplus capacity over those days that are at a lower level (down time) to standardise daily capacities. If, on the other hand, the workload was to be lower than the one faced under normal conditions of employment, then one could also think, if possible, to anticipate activities foreseen in subsequent periods.

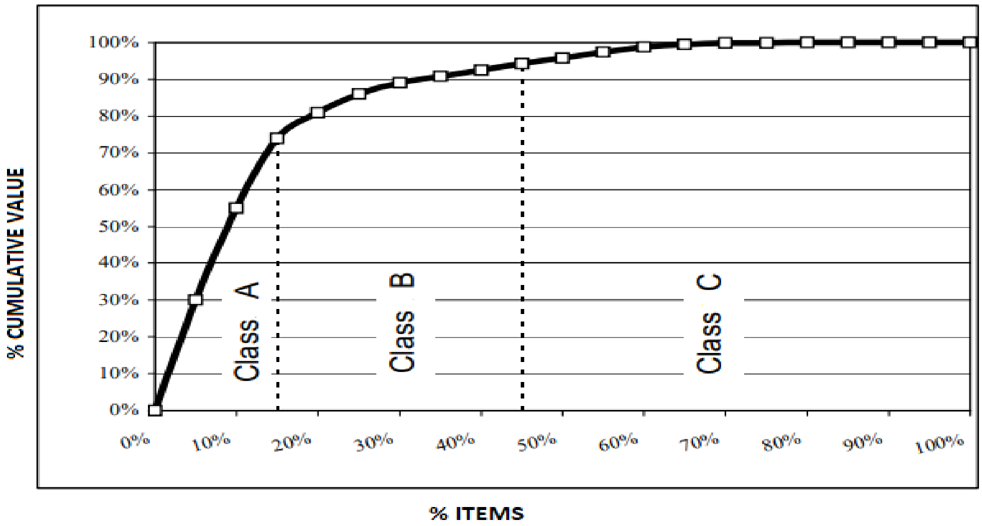


13 - CRP after corrective functions

4.2.3 **“ABC” analysis as strategic support for MRP**

Clustering methods such as ABC analysis for selective stock management could come into play to support the MRP strategy. This type of analysis aims to divide a certain set of subjects into three groups (group A, group B, group C) according to a basic criterion governing the clustering. The use of these methods is already well established within the manufacturing and industrial sectors in general, especially in the context of logistics and inventory management, that’s why a link is made with material requirement planning. With respect to inventory management the ABC analysis done at the level of turnover suggests that there are about 15% of items that together contribute about 70% of the revenue (so they must be carefully monitored) and constitute the group A, while in B fall about 20% of items having a contribution of 20% and in the last one 65% (so a large part) whose contribution to revenue is very low (so the efforts will not be particularly concentrated on controlling this category).

For example:



14 - ABC Analysis

Several studies have tested the good versatility of the method in manufacturing environments and for both purposes (in the article (34) is discussed the use of the methodology applied to automotive products, while for example in (35) it is applied to those in apparel) but only in (36) its using in conjunction with MRP planning is tested.

In addition to 'stock' management methods, there are other scenarios, such as those where product demand is discrete, and a set of algorithms could be applied. Among the best known and most applied are the Part Period algorithm, Wagner Whitin's algorithm and finally the Silver Meal algorithm.

4.3 **Alternative methods for cases with a higher degree of complexity**

4.3.1 **Multi-stage Stochastic Optimisation**

In the other hand, for scenario based multi-stage stochastic optimisation for capacitated multi-echelon MRP systems with lead times and stochastic dynamic demand there are very few studies about the implementation of analytical method to determine safety stock dimension, thus in the light of that supply chains are so different and customised by their actors and the products which pass through.

In (37) was investigated a way in order to find a multi-stage stochastic optimisation method for MRP, as there were no studies in the literature that used an approach such as the one in the cited paper, that eliminates the boundaries between two strategic items in

MRP planning: production batch size (this has directly dependencies with MPS planning) and safety stock. Batch and stock sizing had previously been treated as a combination of two separate issues, with the former directly dependent on orders received and expected and the latter chosen based on the level of service to be provided to the consumer and on the experience (38) (39). The novelty proposed by the researchers of the study is based on a MMCLP approach ("multi-echelon multi-item capacitated lot-sizing problem"), studied by assuming different possible demand trends, simulating thousands of scenarios and on two different frameworks:

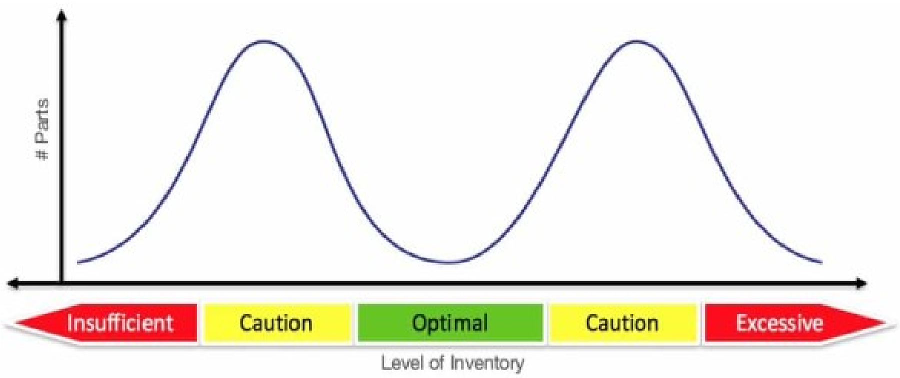
- 1st framework: production capacity is frozen for a certain period and the quantities to be produced together with the set-ups are decided at time zero and fixed

- 2nd framework: set-ups and quantities to be produced are reviewed and changed periodically

The choice of designing a method for evaluating batch size and safety stock jointly via stochastic optimisation showed high flexibility as it considers many constraints but at the same time, especially for framework 2, required high computational efforts. This presents itself as a viable alternative to more traditional methods such as lot-for-lot, EOQ/EPQ, Silver Meal.

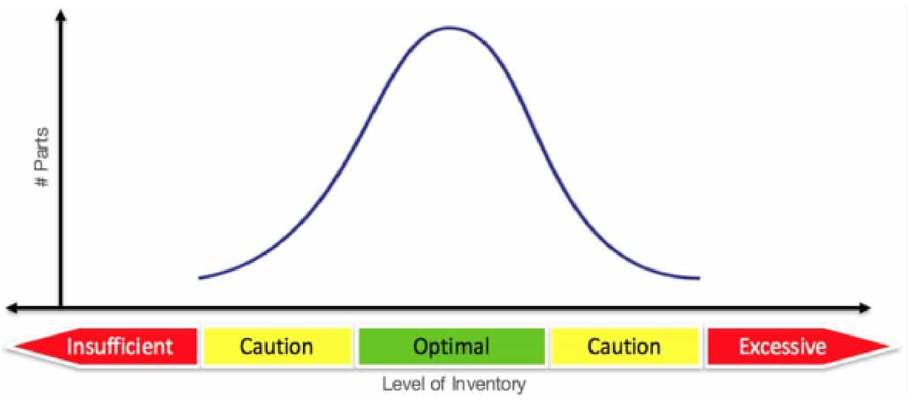
4.3.2 **DDMRP**

Other research, such as (40) and (41) focused on the DDMRP (Demand Driven Material Requirement Planning) methodology by providing both quantitative and qualitative results that can justify the transition from traditional MRP to this version as well as real application cases, bridging the gap due to the mere presence of theoretical descriptions. "In DDMRP, demand is not defined by the statement "what we can and will build" but the statement "what we can and will sell" (42). The DDMRP was first discussed as early as 2011 (42), aims to centralise demand-driven requirements planning and not inventory sizing; this should make the organisation more agile with respect to the dynamism of the market and replace the concept of safety stock so dear to the MRP planning method with that of strategic buffers, which positioning should make the system less dependent on forecast-based values and promote dynamism. "Traditional MRP focuses on answering for how much inventory to hold, and when to release order, while DDMRP focuses on answering for where to position the work-in-process inventory (WIPI)" (43).



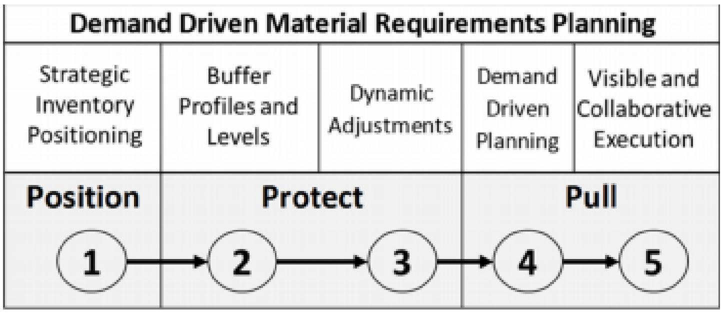
15: DDMRP 1 (44)

The complexity of modern supply chains combined with the increasing complexity of products and the reduction of LTs required by the market has led to the development of a bimodal distribution of inventory levels (image [15]), causing organisations to very often suffer from stock out faults caused by the fact that stock sizing in most cases is not properly designed. The effect that the MMRP is intended to bring is precisely that shown in the figure below [16], i.e. optimal stock sizing, which is positioned in a more "strategic" manner, also in accordance with the Lean and JIT philosophy whose pivotal aspects are merged with those of the MRP (wb-3):



16- DDMRP 2 (44)

The method is presented as a combination of the classic MRP with push philosophy (which suffers from the volatility of forecasts) and the rules of JIT (just in time; pull) and structured of 5 phases:



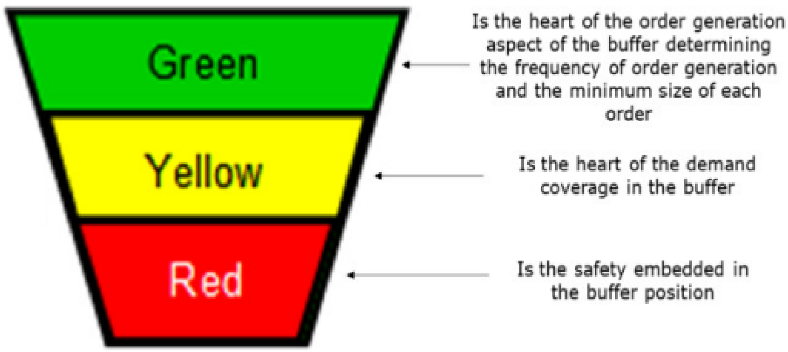
17 – DDMRP Logic (42)

1) Strategic inventory positioning

The first aims to strategically position decoupling points in the product structure or supply chain architecture that will have a dampening effect on demand variability throughout the SC. In most cases, where the DDMRP is used, it is the decoupled lead time that is the most frequently used factor for this step, considered to be the largest cumulative lead time that could occur relative to the BOM of a product.

2) Buffer profiles and levels

The second step is to define critical levels of buffer size in such a way as to avoid both over-sizing and under sizing (the former would lead to high maintenance costs and obsolescence risks as well as excessive space occupancy while the latter could result in stock breakage). Critical levels are identified by 3 colours, whereby we have:



18- DDMRP colored “zones”

Green represents an inventory position that requires no action, yellow represents a part has entered its rebuild or replenishment. Red represents a part that is in jeopardy may require special attention. Each colored area dimension is customised for the specific item by using specific factors. It means that decoupling points could stop the BOM "explosion" in a lower level when the order is made on a higher (45). The three zones are not all one-third the size of the total buffer but are purposely sized according to the size of the product. The area of each is calculated mainly based on the next 3 parameters: DLT, Average Daily Use (ADU) and Minimum Order Quantity (MOQ). The information that these zones provide is strategic: they indicate the amount of stock required to be held at each of the decoupling points and, depending on the colour, there is a prioritisation mode for restocking; the absence of decoupling points can create, the more complex and longer the supply chain along with the variability of products, a distortion of demand (see bullwhip effect) leading to problems such as stock out. The solution presented would, instead, go to dampen this variability. This would prove to be an effective tool that would solve a typical problem of MRP management: the cascading effect of defect propagation. Scheduling the entire MRP without decoupling points inevitably results in longer scheduling windows, and a variation downstream would have repercussions on more and more variations upstream (cascade effect), so we would like to have a management mode that divides the SC into a sort of area "independent" from the variations occurring elsewhere.

3) Dynamic adjustment

For the 3rd point, the use of dynamic buffers is requested in order to have as much reactiveness to adapt to the change of market and working conditions.

4) Demand driven planning

This step is guided by an equation called Net Flow Position (NFP) applied with high frequency precisely to keep the level of the various strategic buffers under control and to generate replenishment orders (resulting in purchase or production orders depending on whether the item in question is manufactured internally or purchased from external suppliers).

5) Visible and collaborative execution

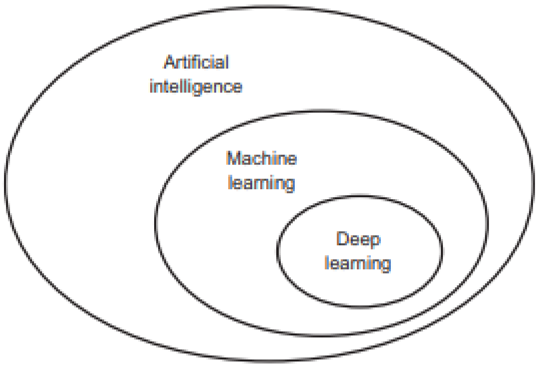
Together with point 4, the latter belongs to the application sphere of the DDMRP method, distinguishing between two phases, a planning phase when, based on the NFP, recovery orders are generated and an execution phase, when the order is dispatched.

5. The sphere of artificial intelligence: a brief overview

Starting from a semantic analysis, the term 'artificial intelligence' is used when talking about machines that simulate human behaviour (from a cognitive point of view) in approaching to perform certain functions and solve problems of various nature (learning and problem solving approach), and this is somewhat the definition that the founding father of this branch of science, John McCarthy, tried to provide, as reported in the article (46). The same article already cited explains how the categories of technologies that can be accepted under the definition of AI are 16: 'These are reasoning, programming, artificial life, belief revision, data mining, distributed AI, expert systems, genetic algorithms, systems, knowledge representation, machine learning, natural language understanding, neural networks, theorem proving, constraint satisfaction, and theory of computation'. Among those mentioned, there are some whose potential has proven to be more valid for use in the logistics sector and the procurement phase: more obsolete solutions are expert systems (47), while today machine learning and neural networks are certainly more considered given the recent developments of I4.0, that have contributed to the procurement phase not only acquiring even more importance from a strategic point of view (e.g. with regard to the objective of minimising stocks and forecasting orders), but also to the formation of a sort of study area dedicated to the application of the leading technologies applied in that specific step; area which in (48) is named as 'Procurement 4.0' and encompasses all activities such as purchasing, leasing, renting, or otherwise acquiring supplies, services, or construction from external suppliers whose execution is optimised by the use of 'technologies that can help procurement managers to deal with purchasing challenges are the Internet of Things (IoT), Cyber-Physical Systems (CPS), Cloud Computing (CC), Big Data Analytics (BDA), Robotics, Blockchain, Artificial Intelligence (AI), Smart Manufacturing, ERP systems, simulation, Internet, procure-to-pay systems, Contract Management Systems (CMS), Expert System, third-party providers, etc.

It is easy to create lists of technologies by harvesting the fruits of previous experiments, but one must ask how in practice one can come up with an AI-based solution that can meet the project needs described above.

5.1 **Machine learning**



19 - Machine Learning in Artificial Intelligence (49)

Machine Learning certainly presents itself as the most promising field of AI to be able to disrupt today's supply chain structure. It is a set of 'tools' capable of evolving their behaviour and computational capabilities through learning steps, to arrive at the condition of being able to emulate the human approach when making certain decisions but with greatly improved calculation and response times. What really makes this tool 'revolutionary' is the fact that it is not simply the result of mere programming but also undergoes training. Therefore, an ML algorithm receives large amounts of data as input to output usable information according to the context; training the algorithm means monitoring its performance to use it when its output has a high level of confidence and reliability, and this must be perceived through the emission of feedback.

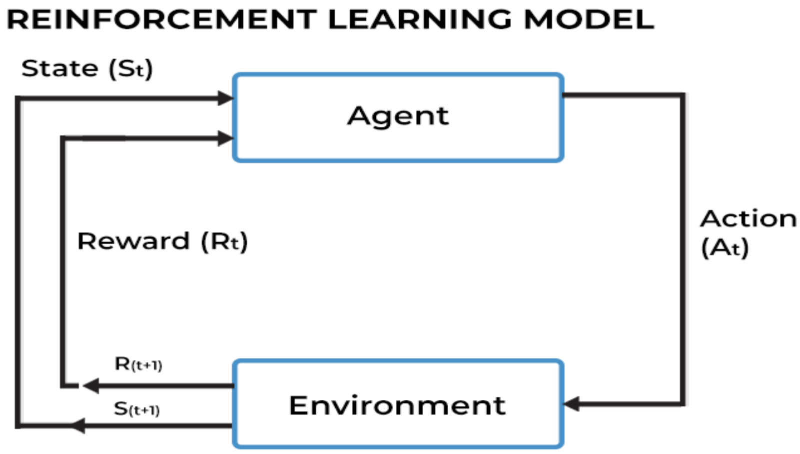
The capabilities of ML, remaining in the logistics sector, make it easy to use for forecasting future orders and sales, to create systems capable of providing the buyer with purchasing 'advice' based on the buyer's habits, to optimise stock quantities based on order forecasting (and this leads, in turn, to having a method of optimizing warehouse organisation, for example). It is usual to distinguish machine learning techniques into 3 categories:

Supervised learning

This category includes algorithms such as the Naive Bayes Classifier, regression/logistic regression, random forest. These are algorithms capable of learning patterns through the processing of historical data (and this is a determining factor especially from the supply chain point of view because large repositories of data significant for business trends can be fed to a human operator) and then use them to provide future forecasts with a more or less high degree of randomness. When using this learning mode, densely populated datasets are required. During training, the model will not know the entire dataset, but the programmer will have to split it in such a way that a robust portion of it (usually around 70-75%) is used to instruct the model to recognise certain hidden patterns/behaviours and, to validate the quality of the instruction, the second, smaller portion is used. Since the second is equally populated with known data, it will be possible, through comparison with the result obtained by the model, to understand how capable it has been of accurate prediction and, in the event of a negative result, to manage its hyperparameters to refine its deductive capacity. Belonging to this category are classification algorithms, linear regression algorithms (useful when one wants to verify a possible linearity, and how robust this is, between the value of a prediction and one of the properties possessed by the dataset) or Hidden Markov models, which can be implemented with the Python programming language (50) (51) (52).

Reinforcement Learning

This class of algorithms is used to train the machine to develop a trial-and-error approach. It performs a series of computations and then evaluates the results and learns from the best ones obtained. So, the machine in the future will be able to make decisions based on the fact that it has learned how to recognise them from past mistakes, this is why it is also known as 'feedback-based learning' (53). In a rather general way, a description can be made of the recurring aspects in algorithms of this class: the presence of an agent that learns because of an interaction with an environment is first and foremost. Unlike in the case of supervised learning, here we do not have input datasets, but there must be an environment capable of throwing feedback to the agent, following the actions it tries to conclude. This 'agent' can decide the action to 'learn' according to the reward given by the environment, thus gaining a certain autonomy. A mode of action such as the one described would seem to be applicable above all to what are called 'sequential decision problems', i.e., situations in which a certain result can only be achieved by choosing the right set of combinations.



20- Reinforcement learning behaviour - schematic figure (wb-4)

The one presented would seem to be a model that can be traced back to a video game. It is no coincidence, in fact, that the industry linked to this sector has been one of the main financiers for the studies, determining a large part of their developments (especially since Google Mind successfully applied this type of algorithm to Atari video games). One could imagine the result of a reinforcement learning algorithm as an artificial neural network that, by observing from the screen what is happening within the video game, generates as output the next action that must be performed in order to maximise the game's score (54) (55).

Unsupervised Learning

Falls into the category of clustering algorithms such as K-mean clustering. They are distinguished by the absence of human intervention in the learning phase why there is no instruction to recognise significant patterns but, instead, the entire action happens in an autonomous manner. They are useful to be able to create, within a certain dataset, separate groups of data whose members are united by characteristic features (56).

As shown in the image below, machine learning does not only meet the needs of Supplier Management and Inventory Management (which are the most characteristic from a logistical point of view) but is also used in other steps of the product life cycle such as its development phase or even the management of the customer's experience of its use.

5.2 **Deep learning: the neural networks**

Neural networks are the ultimate representation of the 'Deep Learning' concept, and the association of these with the structure of the human brain isn't a random choice. The use of this term very close to the field of neuro-sciences, is not intended to indicate that the mechanism that governs this artificial instrument emulates what occurs in the transmission of signals and information between the human brain nodes (given that there is not yet sufficient know-how to know perfectly how this occurs) but, in a very illustrative manner, a neural network takes this name because it is composed of a large number of nodes/neurons that are connected by a series of links that constitute the intricate network that performs the task of signal transport (59). Very common is the technique called feed-forward error back-propagation through the neuron X receives an input signal that depends on the weighted average of the output signals from the neurons linked to it.

The realisation of what we call neural networks can also occur by the use of supervised algorithms and with the aid of traditional programming languages such as the Python one. By writing in Python and accessing libraries such as TensorFlow (one of the first to be born and made open source, this library developed by Google is certainly the point of reference for programmers involved in projects that require high computational efforts, in order to perform particular functions such as prediction) or Pytorch (in this case, a younger library designed by Facebook developers), it is possible to use vast sets of tools and methods to design artificial neural networks: for instance, it is possible to create models capable of recognizing, after training, the belonging class for a certain unit of data, restricted to a range of them, providing as input a vast available dataset (as in the case of DNNs - deep neural networks - classifiers) and obtaining as output both the 'predicted' class and an estimate of the goodness of the result (calculated by precision, accuracy or a series of other metrics) (60) (61)

From an "informatic" point of view a neural network isn't much more than a layered data structure and the name "deep neural network" has been thought because of the high number of layers which could be reached. The major characteristic of those networks is related to the fact in order of the data is transformed with a step-by-step approach during the transition trough the layers from the input to the output one. Between the input and the output layers there could be a certain number of "hidden layers" (taking this name just because it is not possible to see concretely what is happening inside each of them) while, for the each other's two, we could know the data form in input and how it is changed trough the transformations just looking to the output.

In literature is possible to find a very large number of researches about the "Math" behind neural networks behaviour, which are the pillar for the modern knowledge (63) (64).Since the model initially has no instruction regarding the characteristics of the dataset, it is normal to obtain very poor predictions in the preliminary stages. It would be ideal if exist a method for understanding if the model is making enough good predictions as the training phase proceed: this can be realised since the input dataset is populated with known and accessible data, so that a comparison can be made between the expected data measure and the one possessed. Functions that perform this task could be Mean Squared Error, Mean Absolute Error, or Hinge Loss.

Without much detailed explanations it would be goof empathise that a data pre-processing of input data is recommended (in order to feed the model with something prepared appositely for a specific prevision type) and that's the same for the output too (pre-processing and post-processing). It's too easy having over-fitting issues during forecasting process, an example of this situation is when, evaluating the accuracy of the forecasting system, we obtain a value that is greater in the training phase than in the application phase on "unknown" data: if we monitor the performance of the forecasts during the deployment of the network we see that at each training epoch (by epoch is meant the phase of administering the input data and normally when working with large datasets we pass smaller batches, so at each epoch we pass a batch of data) the "accuracy" value of the forecast increases, but we are not trying to create a system that is reliable on known data, but must have this performance on those that are not. Over-fitting occurs when the network learns, for example, patterns specific to that training dataset that are then not as true for a test dataset. To remedy the defect, one must act on the hyperparameters, but their setting requires a certain amount of experience.

5.3 **AI applications for forecasting**

Material Requirement Planning is an operation that requires the use of forecasting systems to ensure that market demands are met. Companies cannot afford the risk of being 'unprepared' during any fluctuations in demand, since they could occur for a wide variety of reasons but, on a one-off basis, this can be resolved by agreements with the customer agreeing for a late delivery. Surely this cannot be an effective method of ensuring a high level of service for the customer because it severely undermines the company's position in the market. If we look at the time-phased record mode as the one to be adopted for updating the MPS then we can realise how the goodness of the forecast plays a key role: normally the forecast has an annual horizon but is 'revised' on a monthly or even weekly basis given its high volatility, incentivised by the not exceptional reliability of the forecasting tool adopted. The data used by these come from the company's history of previous years; we speak of 'time-series data' when dealing with this type of data, that can be useful in identifying significant patterns or recurring seasonal trends in product demand. Complicating the situation further is the large number of factors that could generate a variation in customer demand entity, e.g. the quantity of a certain item in stock, the availability of substitutes and alternative suppliers that the customer may rely on, the success of its market campaign and, on a broader spectrum, the political, social and economic factors of the country where operates (65).

Turning now to the methods that find greater appreciation in everyday practice, depending on the reliability of the forecasting they offer, we can say that the use of "traditional" tools such as those based on the experience of human resources in charge of production planning have been overtaken by more efficient algorithmic-based ones. Some more obsolete methods find their way into manufacturing companies (e.g. linear regression, simple exponential smoothing, moving averages), albeit rarely (66) but 'modern' ones include artificial neural networks (ANN), decision tree techniques, Random Forest, the application of ML as in the case of 'unsupervised' algorithms such as KNN. Depending on the type of industrial reality, there are tools that have shown more or less effectiveness when the forecasts were compared with the actual verified data, but ANNs have aroused, in most sectors, strong interest in production planning by virtue of their high efficiency and low error (67). Making the process even more inconvenient is the fact that companies must apply these forecasting systems to many products at the same time, generating a very heavy data stream.

ANN is a very effective method for various sectors in the manufacturing environment. Many studies have demonstrated the results of its application and justified its goodness by comparing the forecasts obtained firstly with the actual values that became apparent and then with those obtained with other methodologies. Normally, the calculation of the prediction error is taken as a meter for judging its effectiveness. Applications of the ANN method are discussed in the article (68) in which a real case study of an Ecuadorian textile company is reported and the conclusions carried out were all in favor of the MLP (Multilayer Perceptron - belongs to the ANN category) algorithm, that was applied to predict the sales of 5 different SKUs for the following year as a function of those that occurred in the previous 4 years. In (69) a method is applied that is very similar to the context of a manufacturing company in the metal mechanical industry, which, unlike the textile industry, produces items with a higher degree of customisation and a much lower tendency to manifest seasonal patterns.

The peculiar aspect of artificial neural networks is that, unlike many statistical methods that make good predictions when there is a linear relationship between the data, they can be used in particular when there is a 'non-linear' relationship between the items that influence demand trends. And this is the situation most cases.

5.3 **Digital Twin**

The digital twin paves a way to cyber-physical integration. “Digital twin is to create virtual models for physical objects in the digital way to simulate their behaviours” (70). It is probably one of the most promising challenges in terms of making something 'digitised'. As long as we remain within the concept of digitising objects, this might not come as a surprise as in that case, design and simulation software which has been assisting the engineering world for years, has already made great strides. It is when it comes to the digitisation of a process, a system, or a supply chain that the music changes. The concept of a digital twin could be simplified as much as possible by trying to summarise it in 3 distinct classes (71):

- The first is surely that of the real object/process/system (physically tangible) to be digitised.

- The second is the digitised twin: this could be imagined as a perfect copy of the first. By using the word copy, one does not want to allude to the fact that it is simply a snapshot that allows to see the characteristics of the first one at an exact moment in time, but the twin is an entity that 'lives' and, to all intents and purposes, has a connection with its real correspondent on which data travels and is exchanged, even in real time. On the "twin" is possible to make behaviour tests without affecting the real counterpart, so it can continue its working process undisturbed.

- The third, inevitably, is the mode of communication that connects the two entities.

From the point of view of the management sector, this can open a world for those involved in the programming and control of industrial processes and services. In those cases, knowing that one can test the characteristics of a process without having to interrupt its execution, monitor its behaviour after collecting data (even in real time, albeit with computational efforts that require significant investment) and simulate alternative scenarios makes it easier to achieve important advances that would otherwise have required a trial-and-error approach in the 'real' world, where missteps can be followed by huge economic losses.

5.4 **Fuzzy Logic**

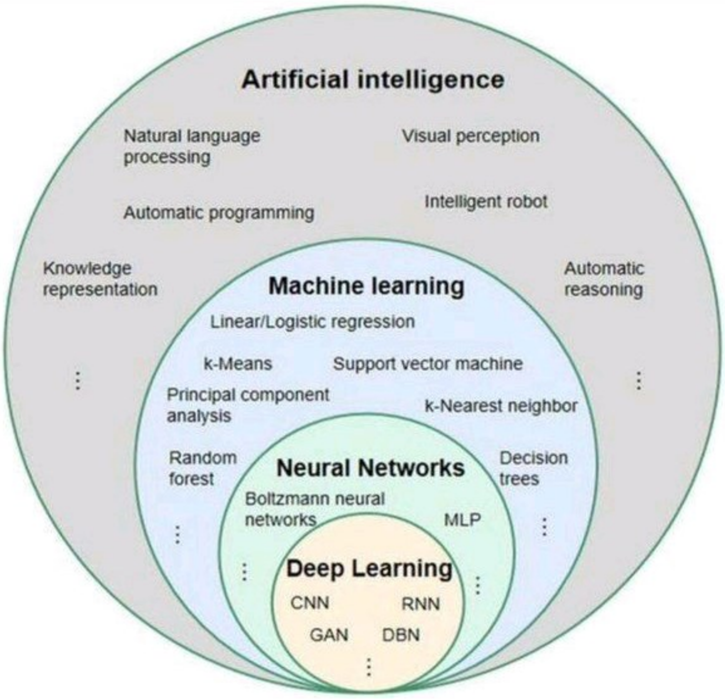
"Fuzzy systems may be less precise than conventional systems but are more like our everyday experiences as human decision makers. We tend to talk in fuzzy terms such as "tall," "large," and "rarely." These terms are not precise, but they are meaningful and allow us to describe real-world situations and reason about them" (72)

The theory that governs fuzzy logic is the result of the pioneering research of the mathematician Lofti Zadeh (73), who is defined as the real father of this branch of mathematics. It was Zadeh who introduced the concepts of fuzzy sets and fuzzy logic, arousing the interest of many researchers from all over the world, with an article published in 1965.

What are the aspects that distinguish this type of logic, defined as "blurred", from the more traditional conception of reasoning? It was the need to make a computer capable of emulating the way of reasoning of the human being in approaching the resolution of a problem, the spark that triggered Zadeh's studies: one aspect that distinguishes the computer from the human being is the its precision in defining the value of a variable through Boolean logic, which intrinsically makes possible the existence of only two conditions, such as 1/0, True/False, Yes/No, It belongs/It does not belong. On the other hand, however, man has never been able to adopt the same vision of the world used by the computer since it is enriched by the need to classify even the infinite shades of grey that lie between black and white, making his intrinsically "approximate" way of reasoning.

In the classic example of having to define the belonging of an element to a certain specific set, the traditional bivalent logic would admit the existence of only two possible conditions, i.e. "total belonging" or "total non-belonging", while Fuzzy would resort to the existence of "Fuzzy Sets" to solve the same problem, answering however that each element can belong to the initial set but with a different "degree of belonging" with respect to another: this would therefore lead this logic to be able to see not only the extremal values of a variable but all the possible nuances between 0 and 1, resorting to the use of the "membership function" as a measure of the membership level, therefore a value equal to 1 of the membership function would indicate that an element has all the cards in good standing to be defined as belonging to that specific set with the maximum absolute certainty, while a value of the same of 0.6, for example, will indicate had that the degree of belonging is "fair" or "good". The reason why this logic is also known by the term "fuzzy" lies in the fact that its intent is to manage inputs that are imprecise and vague, and it is precisely this that makes it the most suitable in simulating the way of reasoning of the person who, with the thousands of nuances that he would be able to give to the same concept, or in describing a phenomenon in quantitative terms only by changing the words and adjectives with which this is described, would not be as emulated by the Boolean one.

All this has introduced a new problem solving approach especially in those contexts in where the "linguistic variables" have such a strategic value that approximating their meaning to 0 or 1 as if they all pointed without distinction to the same meaning could have dangerous repercussions. Zadeh himself has not dedicated his studies only to the mathematical aspect of the logic that governs "fuzzy systems" and "fuzzy sets", but has also tried to evaluate their possible practical implications, especially in the context of the development of decision-making systems that could then be employed in many sectors (74) (75) (76). With these premises it is possible to connect the concepts of fuzzy logic to one of the pillars of artificial intelligence that has always aroused the interest of research, the "natural language processing". With "natural language processing" we indicate the ability of a computing system to interface with the linguistic variables used by human beings without approximating their value with a Boolean approach but by grasping the difference in meaning and, given that fuzzy logic is in itself structured on linguistic variables, then this type of link is anything but forced (77).



21- AI's Sphere (AI4Diversity - LinkedIn)

Given its possible implications, fuzzy logic is often compared to neural networks. These two artifacts of artificial intelligence can generate comparable results relating to the same problem but operating in a more or less decipherable way: due to their way of "hiding" the reasoning process that leads them to provide a certain output, neural networks are defined as black boxes as opposed to systems that operate with fuzzy logic, which instead are considered grey boxes by virtue of the fact that it is possible to trace the steps that lead them to a certain result, making these "rule-based models" more understandable than early (78) and this is mainly the result of the fact that the scaffolding of rules that supports the decision-making system of a fuzzy inferential module is painstakingly previously built by a programmer. For the concepts of fuzzy set, membership function, inferential and rule-based system, theoretical explanations will be provided making their meaning clearer, so as to reduce the ambiguities that could arise when describing the true intent of this study, that is to study the effects and implications of integrating the use of a fuzzy inferential system with a linear programming approach.

6. Fuzzy Logic Theory

In solving optimisation problems using linear programming, mathematics teaches that given a function subjected to a certain set of specific constraints (whose characteristics depend on the field of application) it is possible to optimise it, i.e., by looking for its minimum or that of maximum (79) . “A linear programming problem (also termed linear program or LP) is an optimisation problem to minimise or maximise a linear objective function subject to linear equality/inequality constraints” is the definition that (80) attributes to this methodology. In an industrial/logistics decision making context, linear programming is often used to model problems regarding operations involving economic disbursements and expenses whose entity is to be minimised (industrial planning and scheduling), as in the case of MRP.

But how are linear programming and fuzzy programming different? When the entity of the constraints and targets is known precisely and with certainty, then traditional linear programming is the best way to trace the value of the variables that generate the optimum point (this could coincide with a minimisation or maximisation of the objective function) but when these logics are to be applied to a real context, which can be the manufacturing one, one comes up against an environment that is anything but deterministic where target and constraint values are not known a priori or, in any case, are subject to randomness. In a context such as that of "decision making", the solutions that mathematics offers, under the hypothesis of a deterministic environment, may not be sufficient without the introduction of an "uncertainty" factor (81). And it is precisely the fuzzy logic that fulfills this need.

The theory that governs Fuzzy logic is quite complex and therefore, the intent of this text is not to provide an exhaustive overview of it but rather to clarify the meaning of some strategic concepts that will be taken up later in order to gain a small but well-focused set of knowledge for the purpose of better understanding. In the sections where the more technical features of the project will be explained, reference will often be made to concepts such as Fuzzy-set, Membership Function and Membership Value, Fuzzification, Fuzzy Inference System (FIS), “rules” and the Defuzzification activity.

The most exhaustive way to explain these concepts is to refer to publications by Zadeh himself. First of all, it should be clarified that fuzzy logic has nothing to do with the concepts of probability or randomness, as the first measures the likelihood that a future event will occur, the second has to do with the uncertainty about belonging or not of an object to a non-fuzzy set while Fuzzy logic, as we have already said, has to do with classes where there may be intermediate degrees of belonging and in contrast to probability this measures the ambiguity of an event that has already occurred.

6.1 **The "Fuzzy Set"**

Zadeh explains how a "fuzzy set" is nothing more than a class where it is not possible to define clear boundaries for which it can be said that an object belongs to it completely or does not belong to it at all, and this is useful precisely for modeling the classes of the real world that present this clear contrast of the concept of belonging with respect to the definition to which mathematics has accustomed us. All classes of objects characterised by commonly used adjectives such as big, small, simple, precise, approximate, hot, cold, etc. are fuzzy sets to all intents and purposes. To better highlight the differences with probabilistic concepts, the following example can be proposed: John's degree of belonging to the class of tall men is "0.7" it is a non-probabilistic statement regarding John's belonging to the fuzzy class of men high, while "The probability that John will marry within a year is 0.7" is a probabilistic statement concerning the uncertainty of the occurrence of an event that is not fuzzy, namely marriage.

The development of this logic began right here, with the futuristic goal of giving the computer the same ability to manage these concepts and respond to fuzzy instructions.

When we talk about "Fuzzy Sets" to understand what they consist of, we make a comparison with another category of sets, which is "Crisp Sets". Assume that we have a set that we'll call "Universal set X" consisting of all the natural numbers, then we could define that a Crisp set of this Universal set, for example, the sub-set of all the odd natural numbers (or even, it is indifferent).

X = {1, 2, 3, 4, 5…}

A = set of even numbers = {2, 4, 6, 8, 10…}

B = set of odd numbers = {1, 3, 5, 7, 9…}

A and B are two crisp sets of X. So a "crisp set" is a collection of unordered distinct elements derived from the "Universal set" and it consists in all possible elements taking part in any experiment. A practical example would be to examine a class of students as the "Universal set" and separate them into those who, according to the law, can be considered adults or not. All those who have therefore turned 18 are assigned a value of 1, while 0 for the rest. We have thus created two subsets of the universal set and, in according the Boolean logic, there can be either complete membership or complete non-membership. Things would be different if, instead, the scope is to create a group within the same class where the members can be entered or not according to a physical characteristic: for example, the height. In this case, anyone could be part of the group of "tall people" with a certain degree of membership since a person, according to the expert who builds the inference system which assigns him to the group of "tall people", could be "short", "medium high", "rather low", "definitely high" etc. but all this according to very subjective and non-delineated boundaries based on experience. If the idea of a person's maximum height is 2 meters, then a person of 1.60m would belong to the set of tall people with a much lower degree of membership (not zero) than a person of 1.95m in height, who would a degree of belonging close to 1 (therefore at most).

At this point, we can arrive at the concept of "membership function": it is a function that assigns, to the elements of a set, the corresponding "membership value". Accordingly, the "membership value" represents the degree of membership of an element to a certain set characterised by a certain "membership function". Talking about the degree of membership of an element with respect to a crisp set is not so sensible, since an element with respect to these has only two possibilities i.e., "to be within the set" or "not to be within the set" thus binary logic alone (TRUE/FALSE) would be exhaustive for modeling. When, on the other hand, one enters the context of fuzzy logic, one seeks to define the "degree of truth" of a proposition since, as already anticipated in the previous sections, this logic allows to model uncertainty.

Some formalisms are now given from (82) , (83) and (84)

Given a crisp universe of discourse X, "A" is a fuzzy set characterised by its membership function "μ", where x belongs to X and the number μA (x) could be seen as the degree of membership of x in the fuzzy set A or, equivalently, as the truth value of the statement:

$$\mu_A : X \rightarrow [0; 1]$$

Equation VIII - membership value

The examples given above also highlight some limitations of fuzzy logic; one above all, for example, is the choice of the membership function which, given the same input, could be chosen with very different shapes, thus affecting the mapping of the "membership values" making this process too conditionable and subjective. As for the "mathematics" that models the operations between fuzzy sets, it is necessary to explain the existence of some of these which will then be taken up again in the explanation of the algorithm. With reference to (85) the operations and definitions of greatest interest are the following:

- Normality of a fuzzy set

When do we use to say that a fuzzy set A is normal? It happens if and only if Sup μA (x) = 1, that is, the supremum of μA (x) over X is unity. On the other hand, a fuzzy set is named as "subnormal" if it is not normal. A non-empty subnormal fuzzy set can be normalised by dividing each μA (x) by the factor Sup μA (x).

- **Support of a fuzzy-set**

The support of a fuzzy set A is a set S (A) such that x ∈ S (A) ↔ μA (x) > 0. If μA (x)= constant over S (A), then A is non-fuzzy. Note that a non-fuzzy set may be subnormal.

- Intersection of Fuzzy sets

The intersection of A and B is denoted by A∩B and is defined as the largest fuzzy set contained in both A and B. The membership function of A∩B is given by:

$$\mu A \cap B(\mathrm{x}) = \mathrm{Min}\ (\mu A(\mathrm{x}), \mu B(\mathrm{x}))$$

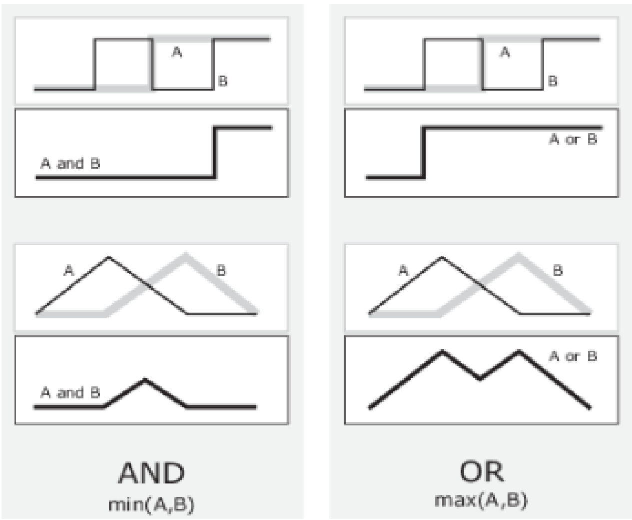
where Min (a, b) = a if a < b and Min (a, b) = b if a > b. t. We should intent the "and" in a "hard" sense and that's mean that we do not allow any tradeoff between μA (x)and μB (x)so long as μA (x) > μB(x) or vice-versa. For example, if μA (x) = 0.7 and μB (x)= 0.2, then μA ∩ B(x) = 0.2 so long as μA(x)> 0.2.

- Union of Fuzzy sets

Union and Intersection of Fuzzy sets are two topics related by a very strong relation because one is the dual of the other and vice versa. The Union of two fuzzy sets is itself a fuzzy set, more precisely the smallest set which contains both A e B (the Union is denoted with the following symbolism "A U B").

$$\mu A \cup B(x) = Max(\mu A(x), \mu B(x))$$

where Max (a,b) = a if a >= b (or b if a < b). As the Intersection has a close relation with the connective "and", the notion of Union has a relation with "or". We should differentiate between "hard or" and the "soft or", because the second one corresponds to the algebraic sum of set A and set B.

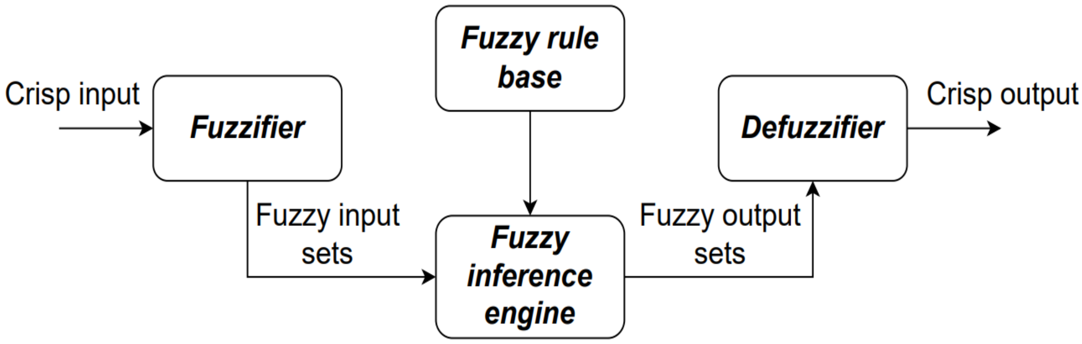


22 - *AND / OR applied on fuzzy sets.*

In the figure on the left, it is possible to get an idea, through a plot of two fuzzy sets, of what applying operations of union and intersection ("and" / "or") means in order to be able to generate a third fuzzy set.

6.2 **Fuzzification / Defuzzification**

The “fuzzification” and “defuzzification” phases take on concrete meaning when fuzzy logic is applied to solve problems in the real world.



23- Fuzzification - Inference - Defuzzification

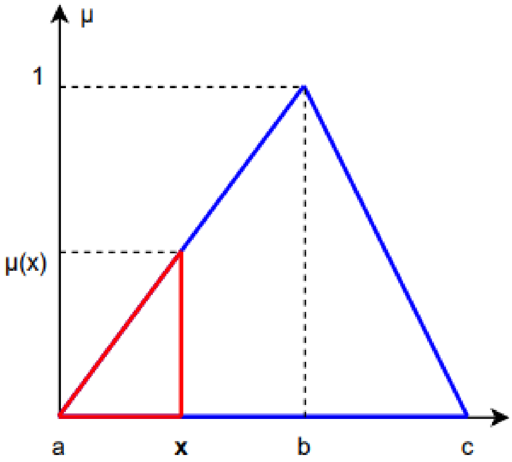
From a theoretical point of view, the first refers to the activity of converting a crisp input (a real scalar value) into its fuzzy equivalent according to a mapping function (the membership function) whose choice, as anticipated, can greatly condition the result of this "translation". Therefore, we can agree on the following definition: “Fuzzification could be defined as the process of converting a crisp input value to a fuzzy value and that is obtained by the use of the information in the knowledge base”.

Thinking of an application case such as the one presented in (86), for which a method is studied for the realisation of a thermoregulation system using this logic, one must think of the "fuzzification" activity as that which allows to allocate input " clear" (in this case temperatures) within fuzzy sets defined through linguistic variables that describe the way humans use natural language to express their perception (e.g. "too cold", "cold", "hot ", "hot" , “too hot” and so on).

This mapping takes place with the choice of a membership function, which strongly conditions the belonging of the input to one of the sets. In the fuzzy theory there are different categories of membership functions, but normally the most used are the following (because they can be easily implemented in control and decision-making systems).

Triangular Membership function

The construction of this function requires the definition of three parameters a, b, and c. The difference between the values of "c" and "a" (c - a) forms the base of the triangle, while the parameter "b" in controls the height.



24 - Membership value on a Triangular MF

The "fuzzification" of the input "x" assumes a value, looking to the exemplifying case showed in the image, that varies between "a" and "b" according to the following relationship which allows to obtain the corresponding "membership value":

$$\mu(x) = (x - a) / (b - a), \quad a \leq x \leq b$$

Equation IX: membership value - Triangular MF 1

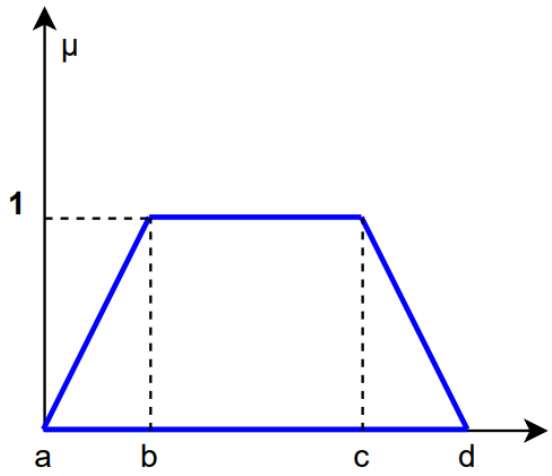
This is just one of the possible scenarios that could occur and can be seen in a combined way according to the report:

$$\mu_{triangle}(x; a, b, c) = \begin{cases} 0, & \text{if} \quad x \leq a \\ \frac{x-a}{b-a}, & \text{if} \quad a \leq x \leq b \\ \frac{c-x}{c-b}, & \text{if} \quad b \leq x \leq c \\ 0, & \text{if} \quad c \leq x \end{cases}$$

$$= \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$$

Equation X - membership value - Triangular MF 2

Trapezoidal Membership function



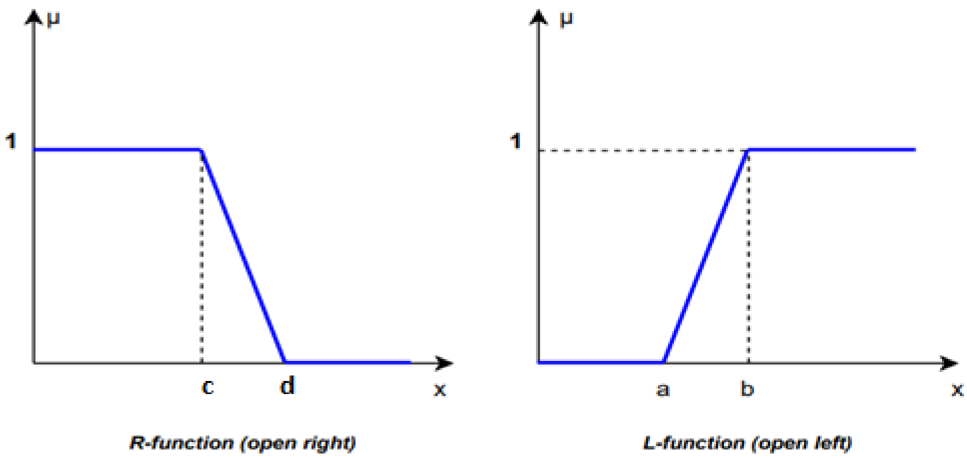
25- Trapezoidal MF

In the case of the trapezoidal function, the necessary parameters to define become four: "a", "b", "c", "d".

$$\mu_{trapezoidal}(x; a, b, c, d) = \begin{cases} 0, & \textit{if} \quad x \leq a \\ \dfrac{x-a}{b-a}, & \textit{if} \quad a \leq x \leq b \\ 1, & \textit{if} \quad b \leq x \leq c \\ \dfrac{d-x}{d-c}, & \textit{if} \quad c \leq x \leq d \\ 0, & \textit{if} \quad d \leq x \end{cases}$$

$$= \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right)$$

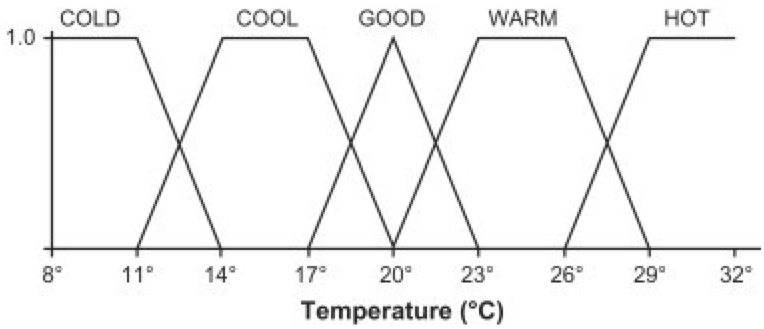
Equation XI - membership value - Trapezoidal MF



26 - Open right and open left MF

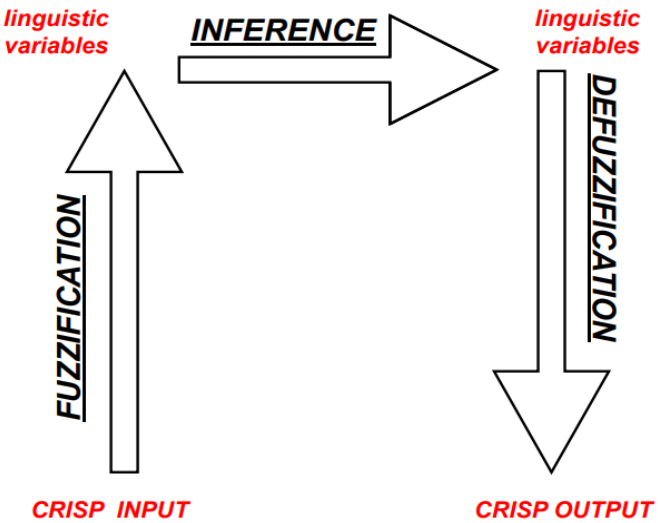
Examples of trapezoidal membership functions are the open-Right and the open-Left functions [26], for which in the first case we have a condition of the type “a” = “b” = - ∞, and “c” = “d” = +∞ for the second one.

Very often, especially for the creation of control or decision-making systems, the use of a combination of several membership functions occurs, for example, in the case of the previously introduced temperature controller one could think of a combination of the triangular and trapezoidal as in the following image:



27 - Temperature feeling as fuzzy sets.

Once the crisp inputs have been mapped into fuzzy values, it is necessary to proceed with their "defuzzification", a process coinciding with re-converting the fuzzy entity back to a crisp value, that is obtained through aggregation methods of the outputs from various rules and fuzzy sets that have been created. The need for this step lies in the fact that a controller/computer always needs to work with a numerical "something" and therefore the output must necessarily have certain characteristics.



28- inference logic process

6.3 **FIS and Rules (Mahmdani/Sugeno and IF/THEN)**

For the concept of "inference" in the context of a system that uses fuzzy logic, the reference is to its ability to draw deductions based on a comparison between the value of the inputs and the "directives" that a series of rules prescribe constructed ones require evaluating before outputting a conclusion. These rule systems, based in most controllers on IF - THEN type chains, constitute to all intents and purposes the major repository of knowledge of the domain expert who built them.

The inputs supplied as input are "compared" with linguistic variables which, as previously anticipated, constitute nothing more than fuzzy sets with very specific limit values: what is done is therefore to verify that the input value of a certain variable is contained or not within the set and consequently take a "decision" in output.

For example:

IF "x" is A AND "y" is B THEN …

or

IF "X" is A, THEN "y" is B…

To clarify the concept, we'll use a traditional example for the theory of fuzzy logic, known as "The tipping problem" (87). If one seeks to create a decision-making system capable of accurately calculating the value of the tip that a restaurant customer must give to the staff who served him, one could decide to evaluate its size on the basis of two observations:

A) the perceived quality of the service

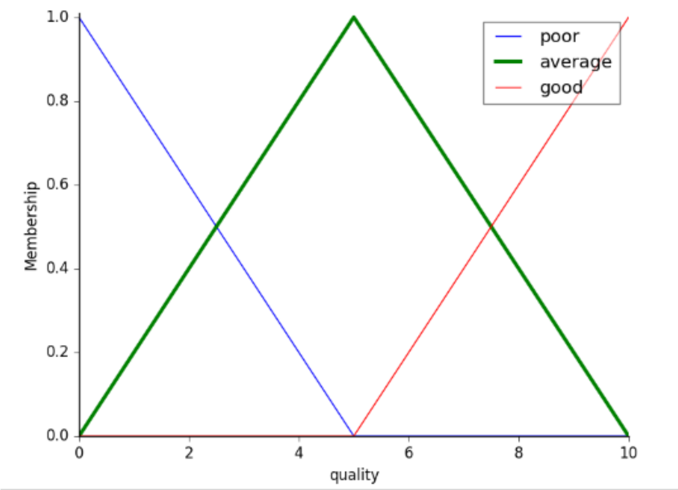
B) the perceived quality of the food

Each of the perceived qualities would be evaluated, for example, with a score from 1 (low) to 10 (maximum). At this point, one could think of creating a quality evaluation meter based on 3 linguistic variables:

- "POOR" quality

- "AVERAGE" quality

- "HIGH" quality

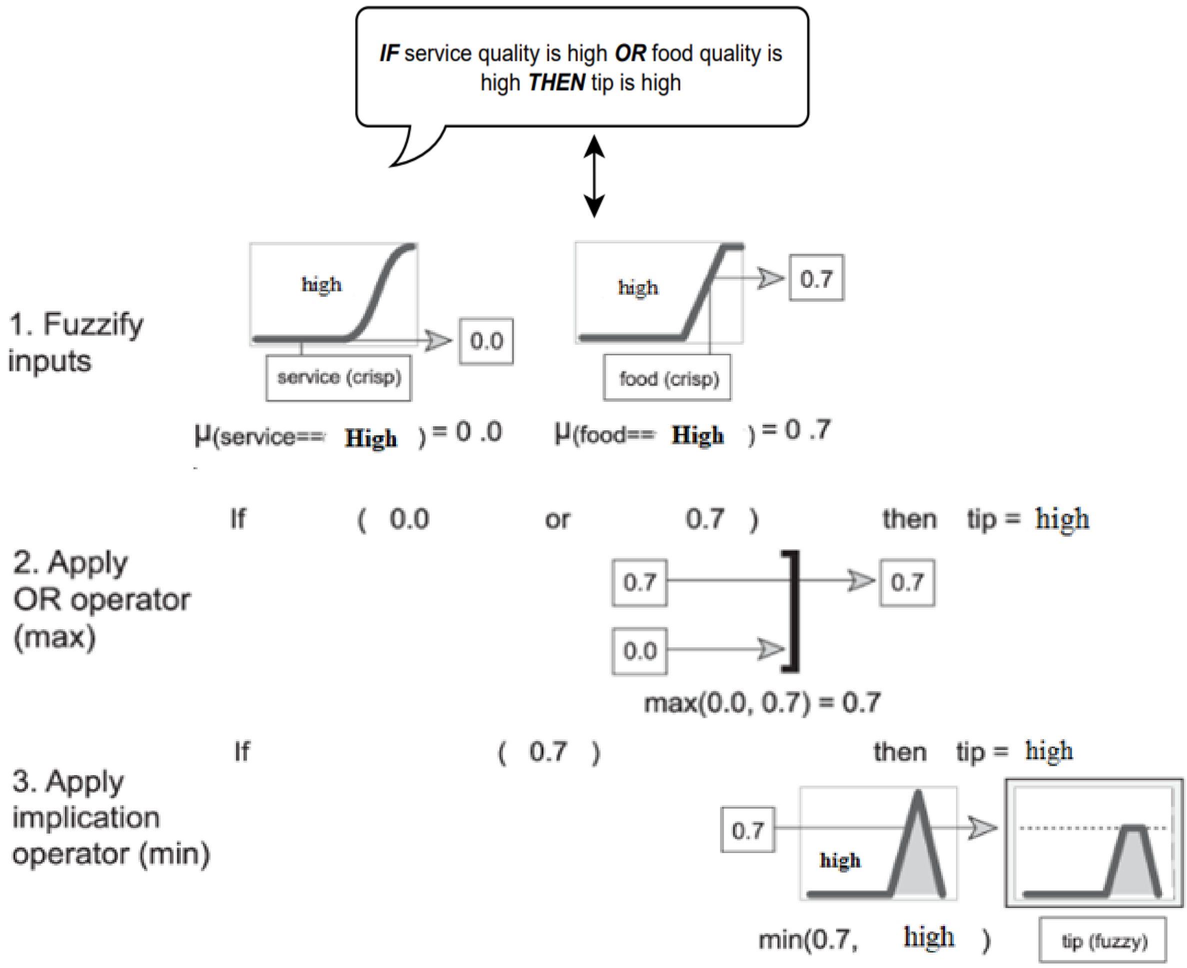


29- Three triangular MF for quality representation

Each of these 3 terms is represented by a membership function whose terms (for example "a", "b", "c" in the case of use of the triangular type function) have very specific assigned values. From this point on, the FIS (which stands for fuzzy inference system) must intervene.

6.3.1 **FIS (fuzzy inference system)**

Assuming to consider a rule like [***IF*** service quality is “high” ***OR*** food quality is “high” ***THEN*** tip is “high”] (image [30]) then the system performs the “fuzzification” of the quality input values of the service and the food (i.e. the marks chosen by the customer on a scale from one to ten) and maps their value by giving them an evaluation between 0 and 1 with respect to that specific fuzzy set: i.e. if the quality of the food had a value of membership equal to 1 would mean that it is maximum, therefore it can be deduced that the customer will have assigned it a vote equal to 10; while in the event that the grade is not equal to one of the extremes (0 or 10) but, for example, equal to 8 then this would mean that within the "HIGH quality" fuzzy set, whose extremes range from 5 to 10 (image [30]), a rating of about 0.6 would be get.



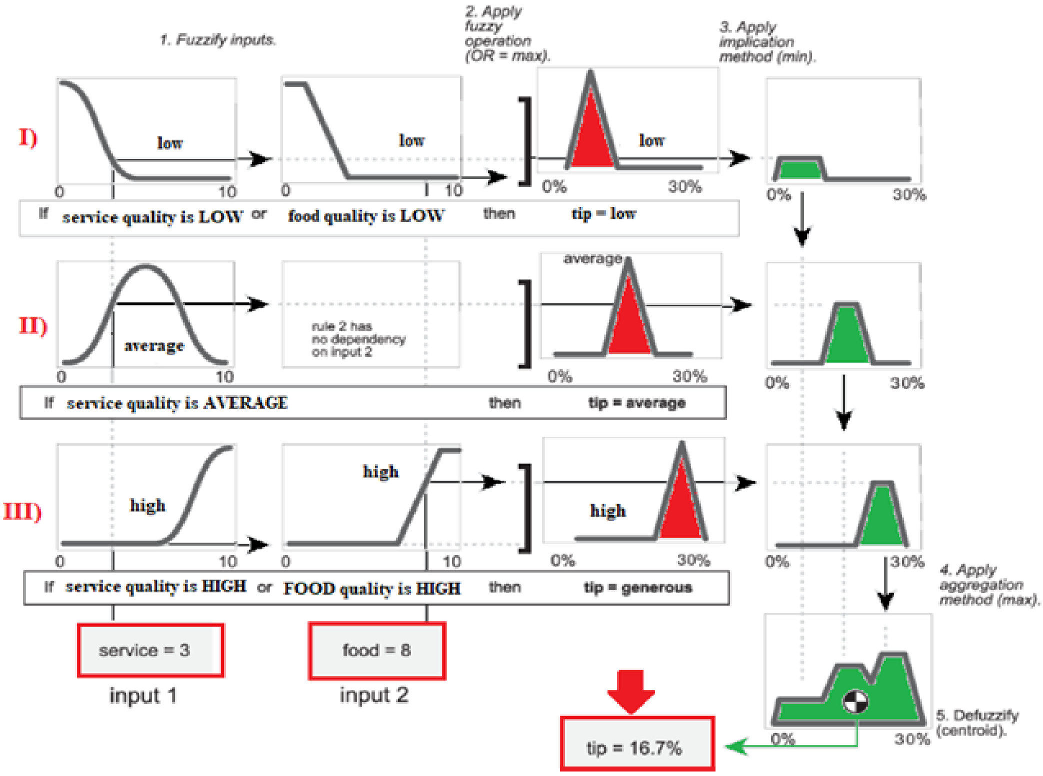
30- Fuzzy resolution for the "tipping problem" - 1

The inference system may use a different defuzzification approach depending on the "model". Normally, in the literature, is used to make reference to three macro typologies of FIS corresponding to that of "Mamdani" (the first model of FIS tested and above all the same one used for the purposes of the project; thus greater attention will be devoted), the model of "Takagi - Sugeno-Kang" (or simply "Sugeno") and that of "Tsukamoto"; the first two mentioned are those that find greater application. The differences between the two types reside not so much in the way they provide an output but precisely on the type of output since it corresponds to a fuzzy set in the case of the Mamdani-type FIS while it would be a linear function for the "Sugeno" one. Obtaining the result, according to (88) takes place according to a very precise series of steps:

1. the first step consists in converting the input crisp into the corresponding "membership-value" according to a specific fuzzy set (fuzzification).

2. obtaining an output from each of the applied rules.
3. aggregation of the outputs obtained.
4. Defuzzification.

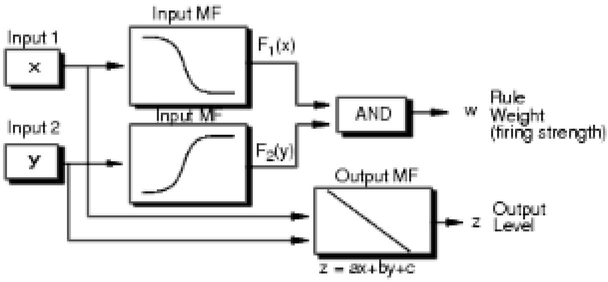
Step 3 is the one that highlights the main differences between the various methods. In the case of the Mamdani model, the outputs are obtained with the application of the "Max - Min" operator; its effect is the same as that previously introduced in the discussion about the possible operations in fuzzy logic.



31- Fuzzy resolution for the "tipping problem" - 2

The fact that the output obtained from the Mamdani model is easily transformable into its equivalent "linguistic" form makes it to all intents and purposes the most intuitive one to use in decision making contexts albeit with lower computational performance than that of Sugeno (89).

As far as the Sugeno model is concerned, on the other hand, an expression of a linear type dependent on the input values will be obtained in the output. A schematic representation of how it works is as follows:



32 - Takagi Sugeno FIS

Being more efficient in terms of performance, this model is more used in the construction of controllers. The fundamental differences between the two FIS are summarised in table (1):

DIFFERENCES BETWEEN MAMDANI-FIS & SUGENO-FIS	
MAMDANI	**SUGENO**
Intuitive and Interpretable	More computationally efficient
Low Accuracy	High Accuracy
Output = Fuzzy Set	Output = Linear Function
Well Suited for Human Input	Work well with optimisation
Less flexibility for system design	More flexibility for system design
Expressive power	Low interpretability

Table 1 - Mamdani & Sugeno FIS differences

6.4 **Defuzzification** (meaning and methods)

The last step that sanctions the achievement of the output is that of "defuzzification". The meaning of this operation has already been described previously but it is good to have an idea of how this happens and what are the methods to carry it out. Also, in this case, there are categories which in turn encompass different variants, the main ones being:

- Centroid methods

- Center of gravity method (CoG)
- Center of sum method (CoS)
- Center of area method (CoA)

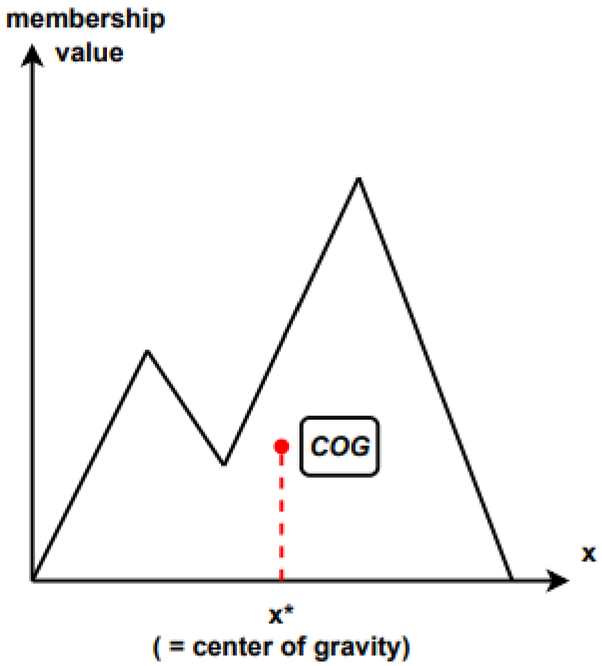
- Lambda Cut Method

- **Maxima Methods**

- Height method
- First of maxima (FoM)
- Last of maxima (LoM)
- Mean of maxima (MoM)
- Weighted average method

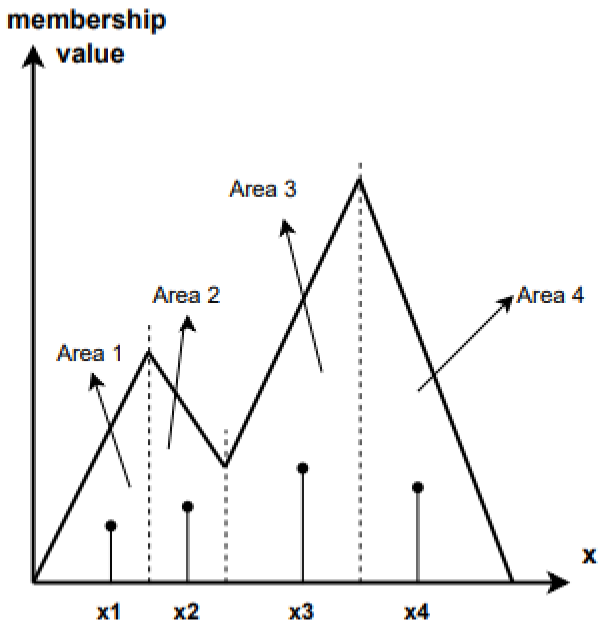
6.4.1 **CoG (center of gravity defuzzification)**

The CoG, also known as "centroid method", is the one actually used inside the current project. Basically Centroid defuzzification returns the center of gravity of the fuzzy set along the x-axis, and provides a crisp value based on that center.



33 - Centroid defuzzification method (COG-1)

It practically works by dividing the total area under the membership function in a certain number of sub-areas and then the centroid of each of them is calculated. At the end, by making the summation of each sub-area the defuzzified value for a discrete fuzzy set is taken.



34 - Centroid defuzzification method (COG-2)

The operation that leads to obtaining the "center of gravity" can be formalised as follows:

$$x = \frac{\int \mu\bar{c}(x) \cdot x\,dx}{\int \mu\bar{c}(x)\,dx}$$

Equation XII - Center of gravity (1)

The following mathematical formalism:

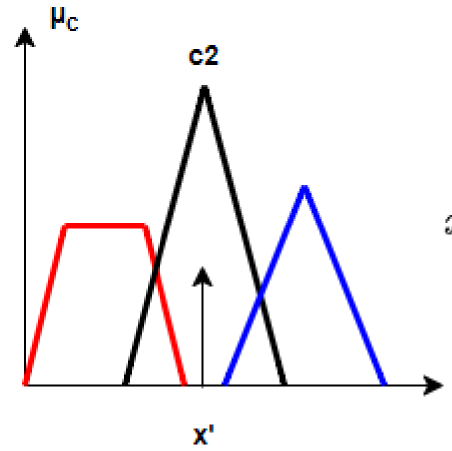
$$\int \mu\bar{c}(x)\,dx$$

Equation XIII - Center of gravity (2)

is equivalent to the area under the curve.

CoA (center of largest area defuzzification)

The "center of largest area" is a very efficient defuzzification method. In the application case of a fuzzy set composed of two sub-regions, this translates into the fact that the center of gravity of the region which, between the two, has the greater area is taken as the center of gravity.

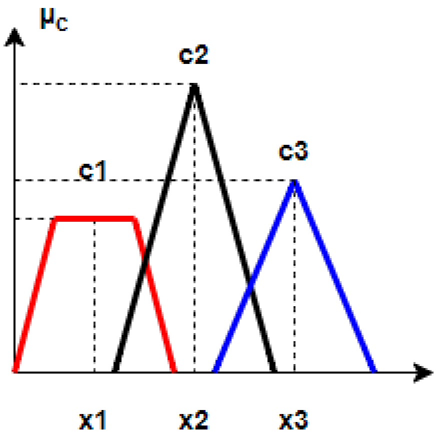


$$x = \left(\int \mu_{cm}(x) * x' dx\right) / \left(\int \mu_{cm}(x)\right)$$

35 - CoA calculation

Weigthed average method

This method, also known as "Sugeno defuzzification", requires very little computational effort and is therefore known for its great efficiency. However, its application possibilities are not unlimited, as it can only be used in contexts where the fuzzy sets have symmetric output membership functions. The calculation using this approach expects each outgoing membership function to have an impact weighted based on its maximum membership value.

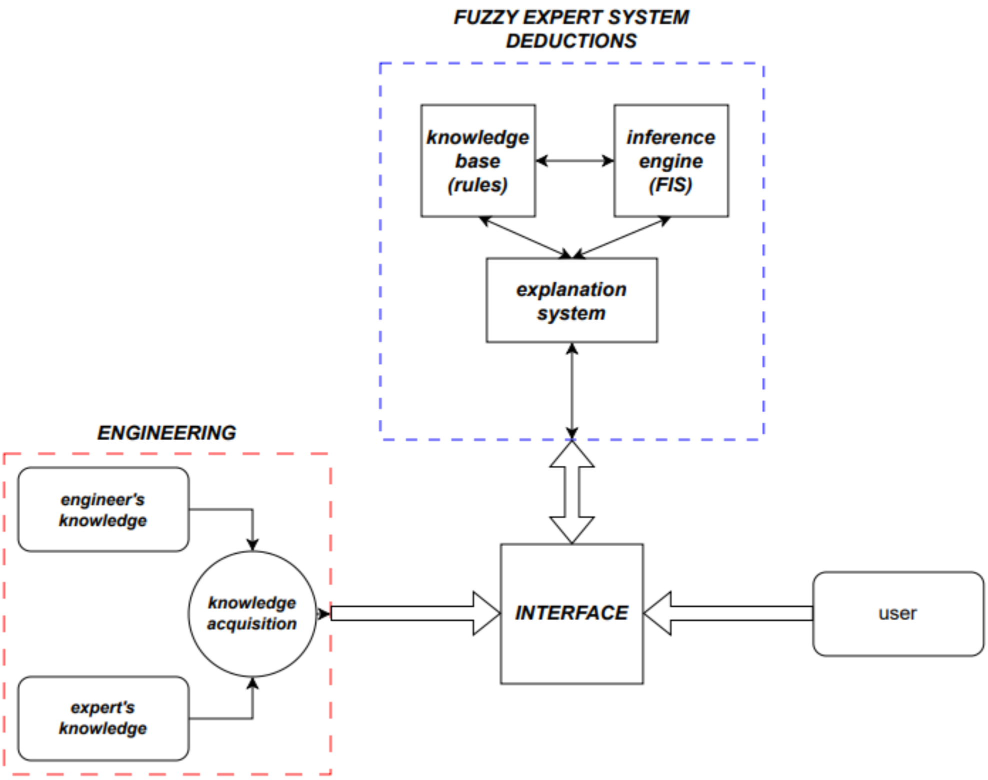


$$x = \sum_{i=1}^{n} \mu_{c_i}(x_i) * (x_i) / \sum_{i=1}^{n} \mu_{c_i}(x_i)$$

36 - Weighted average method

6.5 **Expert Systems and Fuzzy Logic** (FES: fuzzy expert systems)

That of expert systems is another of the artifacts that are welcomed in the sphere of artificial intelligence. According to (71) can be defined as systems that in the decision-making phase can provide solutions based on predefined and programmed "rules". We could model the elementary structure of an expert system as in the next figure [37]:



37 - Fuzzy Expert System logic

From the image:

- "knowledge base" contains all those rules (discussed in more detail below) that enable the system to reason and provide output according to the characteristics of the input (90)

- the "knowledge engineer" constitutes "the memory" of the system where those knowledge sets necessary to solve problems related to a specific domain are stored, allowing the experience of a domain expert to be captured. This set contains information that must be understandable and easily updated.

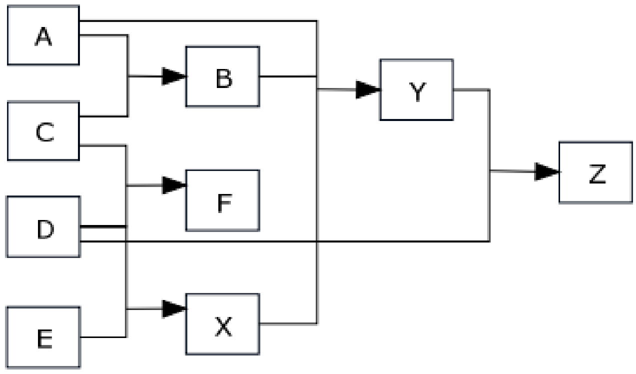
- "Inference engine," on the other hand, is the container of those externally programmed rules (membership functions) that determine the decision-making ability and quality of the final output. These rules return an output by intervening on the input data precisely according to the "knowledge base".

- the "user interface" allows the user to approach the expert system that is also required to be provided with a system that can explain to the same user the reasoning that guided it to provide a certain output.

Normally, however, it is the inference engine, knowledge base, and interface that are considered the brains of an expert system (91). In (92), a rather comprehensive definition is given to frame this type of tool: "Instead of representing knowledge in a declarative, static way as a set of things which are true, rule-based systems represent knowledge in terms of a set of rules that tells what to do or what to conclude in different situations." Kandel in (90) also explains very simply how an expert system can be thought of as a kind of program that uses its own domino knowledge (equivalent to that of a domain expert) and a set of reasoning techniques to solve problems.

At this point, what may seem to be very complex is precisely the step leading to the conversion of human knowledge into the form of rules, which are predominantly constructed according to a repeated IF-THEN pattern. Also, in (91), it is explained how it is more efficient to create an expert system for a specific application domain rather than one whose area of use is too extensive, since at that point the knowledge encoding phase would require an overly extensive set of rules that would end up making the whole system inefficient. Conferring domain knowledge to expert systems is precisely the "bottleneck" activity that most often ends up penalizing its use (90).

Extensive studies that seek to make clear how expert systems are provided with domain knowledge (93) and how they perform their functions are quite numerous (94) (95) and show no dissenting opinions, simply put: based on the "facts" contained in the system's database certain rules are selected or not selected, i.e., the expert system begins by evaluating all the "IF" conditions and selects those that may have a match with the problem under consideration; if the matching exists then it moves on to the "THEN" execution that will result in the generation of a certain result and so on and so forth until the system arrives at an answer to the problem that was initially presented to it. What is generated is a structure called the "Inference Chain" (91), like the one shown in the next image and taken from the same study:



38 - inference chain

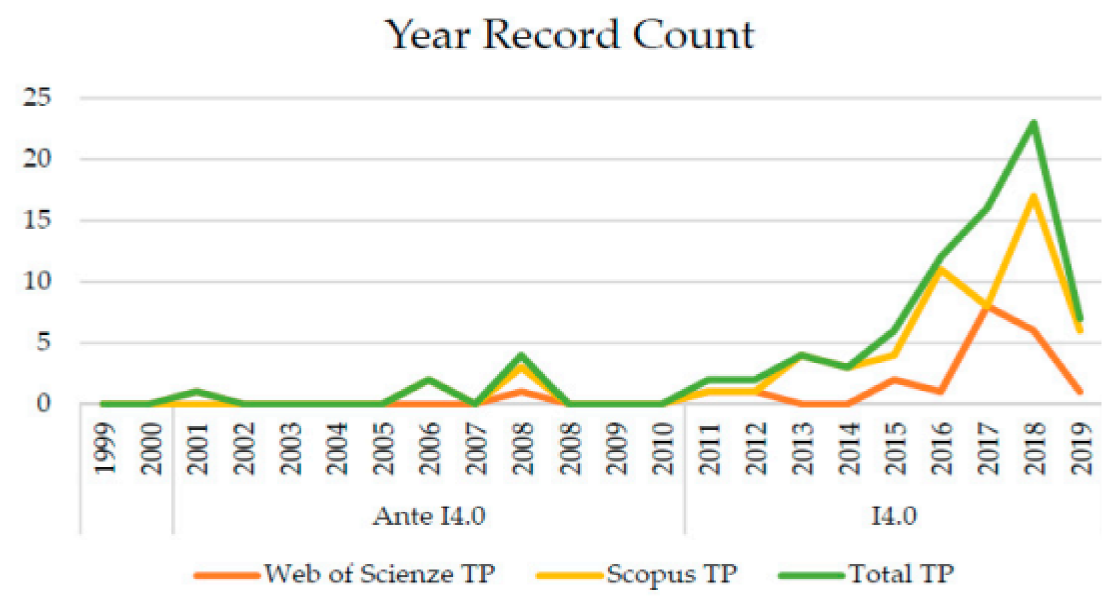
The potential of expert systems has become increasingly known to the scientific community since they have begun to be used in conjunction with other artifacts of artificial intelligence, such as neural networks for example (96). But the use of fuzzy logic and expert systems based on it is far from a recent introduction. Already in (97) their applicability in the context of manufacturing-related problems as a method for solving optimisation problems is demonstrated. These systems that are referred to as "rule-based" have proven effective precisely for very complex decision-making processes (81) and are also probably the most popular and dominant combination of those that may involve the use of "Fuzzy logic" (71).

This combined use demonstrated positive effects in contexts where the approach used for decision making was constrained to a heuristic basis and was a function of the decision maker's cumulative experience alone; a context where, moreover, the uncertainty associated with the use of natural language expressions introduced several problems related precisely to imprecision and misinterpretation of the meanings of certain terms with respect to the context of use. This is why some of them are called precisely "fuzzy terms". A very recurrent example in the literature (91) is that of the expression *"John is tall"* in which the term "tall" is difficult to transform into machine language, since binary logic alone would lead to a loss of information, in light of the fact that "true/false" or "yes/no" alone would leave no space for a third region where a proposition could be "possible" and have some distance from the values of true and false, as formalised by Łukasiewicz in the 1950s (98). For this reason, in application areas where the decision maker has to deal with information that comes in the form of fuzzy expressions, it has been necessary to equip expert systems with the capability of fuzzy reasoning (82).

7. The use of AI in production and logistics management: how fuzzy logic could play a key role

By conducting research on the use of artificial intelligence in supply chains, especially for logistical phases such as warehouse management, procurement of raw materials and components and decision making processes in general, it has emerged that this one has become a 'best topic' within both leading and emerging industrial realities that, for some years now, have devoted a large part of their efforts to experimenting and validating increasingly cutting-edge solutions; all this stands in stark contrast to the world of academia and research, since it seems to have lost 'ground' with respect to these topics that didn't received the same effort spent by the former and this is robustly justified by the small number of articles that can be consulted in the literature about the theme. As reported in (99) the topic of AI applied to the different stages of the supply chain has been the subject of research since 2008/2009 but has experienced a boom between 2019/2020, which can be seen from the increased number of publications dating back to that period. The reason for this trend is easy to understand: firstly, the Industry4.0 incentives have motivated greater interest in the sector by spurring research to find more and more advanced solutions (as in the case of Italy for example, where the government has planned a national plan named (99) and secondly, the technological evolution has meant that together with the possibility of accessing tools and devices capable of handling ever greater volumes of data at acceptable speeds, experimentation has not been discouraged (IoT evolution). Other interesting information that can be extrapolated from the same study (100) is that concerning the geolocation of the articles and the lack of relevance: according to the researchers who conducted it, of the 50 or so articles published over 10 years and selected from those that mention the use of AI as a tool for improving supply chain management, one can make a skim that reduces their number to 35 (which are the articles that do not simply mention the topic but adopt it as the subject of the study), while a second halves it to about 15. The latter are the articles that have been cited at least once, certainly proving to be more valid from a research point of view. The geolocation trend, on the other hand, shows how most of them were the result of studies carried out in the USA or China, countries where governments have a strong interest in artificial intelligence applications (particularly for military purposes) and are therefore not reluctant to fund research activities in the field (101). And it is precisely since machine intelligence has managed to generate a strong impact in the business world and society at large that it has attracted their attention and that of the scientific community (102). In the European context, on the other hand, the topic still remains taboo, probably due to a different conception and philosophy towards the topic of privacy and, above all, due to the ingrained mentality that AI applied in the industrial context could lead to the reducing of people employments in the workplaces; several studies such as that of (103) have tried to curb this conception by explaining how certainly artificial intelligence can help to eliminate the need for employment for certain tasks but, at the same time, it can generate the need for a very high number of positions due to the new support professionals it would require.

A study carried out in Italy (104) reports an interesting assessment of the number of publications dealing with the use of AI in various sectors and it is clear how these have soared since 2012 onwards, after the foundations of Industry4.0 were laid (the data refer to two repositories of scientific articles in particular)

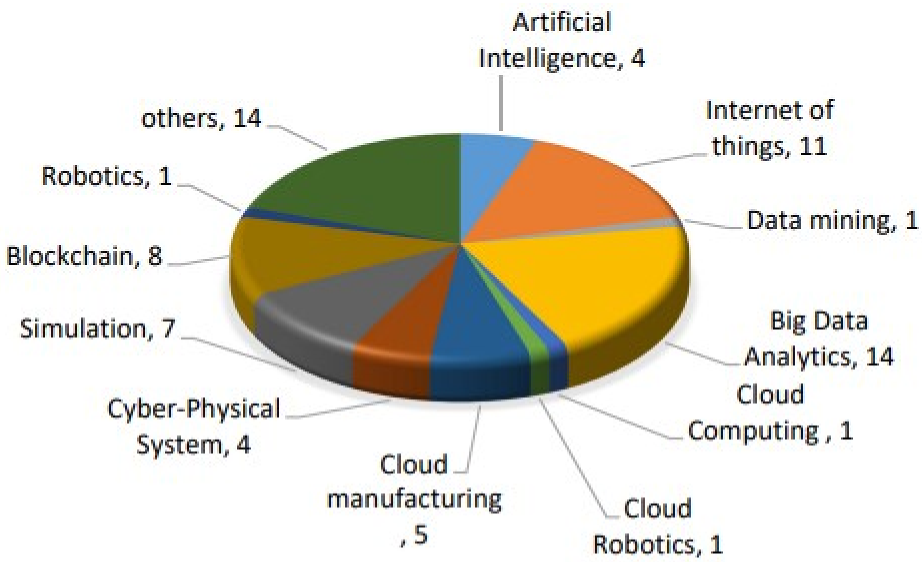


39 - Research for AI in Industrial Management (1)

Also clarifying is the figure regarding the top 20 research areas that have contributed the most to the popularisation of the subject, with those of computer science and engineering leading the way (so as to understand that despite the fact that the engineering research sector has not dedicated itself in depth to the subject of artificial intelligence, there are other areas where the contribution is even poorer). The graph shows on the X-axis the name of the research area and on the Y-axis the number of publications analyzed over a 20-year time horizon (1999-2019).

40 - Research for AI in Industrial Management (2)

In (105) rather representative statistics of the most applied technologies in procurement are given. The pie chart below emphasises how BDA and IoT are the ones that, out of the population considered, have received the most academic attention:



41 - Research for AI in Industrial Management (3)

Merely reviewing the technologies that can be used would not provide a completely meaningful figure if not accompanied by all the constraints and difficulties that lie behind the implementation of IoT-based solutions and the analysis of large masses of data. In the paper (106) is described the result of a study, based on both a literature analysis and consultation with industrial managers interested in the context, about the attempt to classify what are the main barriers to the use of these tools within the industrial reality that are putting a brake on the propagation of this evolution. Having extrapolated the data of interest from the documents studied and gathered the opinions of people who have worked, directly or otherwise, on projects of this type and encountered the related problems, the whole concludes with a ranking carried out using the "Nonlinear Fuzzy Prioritisation" method that shows how it is the lack of suitable infrastructures, insufficiently robust cybersecurity systems and environmental risks that most undermine the digitisation of supply chains. The three items can be traced back to three different categories of problems, the first one to a 'managerial' one, because it is obvious that in order to reach a level of performance that can manage a supply chain with a high degree of digitisation, it is necessary to invest large amounts of resources, which company managers would only be willing to give after very careful evaluations of what the advantages and associated risks could be; but since the first investments are always the largest and with the most random returns, there is always strong pessimism to govern this type of choice. The digitisation of a plant ranges from the purchase and installation of all those sensors and instrumentation that allow the collection and storage of data in real time to the far from trivial construction of a robust connection network that ensures the minimisation of interference or signal loss. The second belongs to a category linked to the management of data security, since the absence of protection systems can be a danger both for guaranteeing its integrity (the loss of which can ripple through several stages of the supply chain) and for the privacy associated with its dissemination (about this topic is also clarifying is the study carried out in (107). For the last one, on the other hand, we fall into the area of sustainability, since the Industry4.0 philosophy has shown itself to be particularly sensitive to the problem linked to the environmental sphere, promoting digitisation also in order to create systems that seek to reduce energy consumption and minimise the impact on the environment. For what is linked to the topic 'environment' is not so much the search for eco-friendly materials or the choice of non-impacting processes, but it is precisely the energy consumption linked to all the systems and devices connected to the same network that together create a rather energy-intensive entity (108).

While scientific articles describing direct applications of artificial intelligence in the context of decision making in supply chains (a context where material requirement planning management can be included) are relatively few, studies about methods for optimizing the planning process and the result of the same, in order to make it as superimposable as possible with the real production scenario, are on the other hand numerous and, some of these that will be described, can open up rather interesting avenues not only for the design of new tools but also for enhancing more rooted ones that nowadays enjoy greater prestige. Given the rather large number of researches we will pause in reporting those that have a greater correlation with the features one wants to implement in the project.

The management of the company's MRP is entrusted to a planner who, theoretically, should possess all the industry knowledge that can guide him or her in making targeted decisions to foster the organisation's economy (minimizing expenses) and to ensure its stability in the market (maintaining a high level of service for the customer) (111). Unfortunately, the randomness of some data classes that populate the planning descendants is caused by their sources as they intrinsically possess a certain degree of uncertainty, making the former unreliable: just think, for example, of demand forecasts over large time intervals, the goodness of which depends precisely on the way in which they are performed and on the extension of the period; then there is the talk about lead times that depend from supplier to supplier (assuming that the organisation has the ability to interact with multiple suppliers for the same product) and any anomalies that plague the supply chain and the market in general (manifestation of the Bullwhip effect).

"*If one wishes to create rational-suitable mathematical models of the real world, then they must be able to incorporate uncertainty and suggest how to process with imprecise information*" (45)

At this point, the justifications for developing an optimisation model that considers the uncertainty factor (fuzziness factor) have already been presented, since the deterministic one is no longer considered compatible with the characteristics of the modern market. As already anticipated, the literature has offered number of insights to interface with this problem:

- analytical approaches (109)

- simulation approaches (110)

- hybrid modelling approaches (111)

These listed are approaches based on the availability of statistical data, mainly extracted from the organisation's historical database, which is not always assumed to be available. Therefore, to make the character of the modeling even more versatile to different application contexts and scenarios, it was chosen to opt for "Fuzzy Mathematical Programming," seen as the most effective way to introduce the uncertainty factor within the model (112). Several researchers have experimented with its use in the context of resource and commodity planning, enriching the studies with real-world application cases that could highlight its benefits, thus providing inspiration for the project characteristics under consideration (111), (113), (114), (115). Given that some of these models been tested in industrial settings such as automotive manufacturing, its validity can also be hypothesised within a manufacturing environment such as the one for which the algorithm is designed (industrial machine manufacturing), given that the two are accumulated by the processing of products with very complex BOMs, the involvement of many actors in the supply chain, a make-to-order manufacturing philosophy.

Most of these research agree to first implement algorithms that exploit linear programming and then introduce fuzzy character into programming. In (114) a linear programming model is developed for target cost minimisation called MRPDet ("Det" stands for deterministic), on which various constraints are to be imposed: the model is extremely effective if the value of the target cost were known precisely, but what if it were not? The material planner might know approximately the value of the target cost because, due to internal choices within the organisation, a solution to the scheduling would be deemed acceptable to stay within a target cost range; that is, one might have an idea of what roughly the target should be (perhaps based on past choices that the organisation has decided to adopt for a certain product category) and consequently obtain the value of decision variables that satisfy a "Target Range" rather than a unique value.

8. IIRA Architecture (wb – 6)

From the official summary of the AIDEAS project, it is precisely required that all documentation supporting the implemented AIDEAS solutions be created following a specific standard called IIRA (acronym for Industrial Internet Reference Architecture), first published in 2015 (116) .

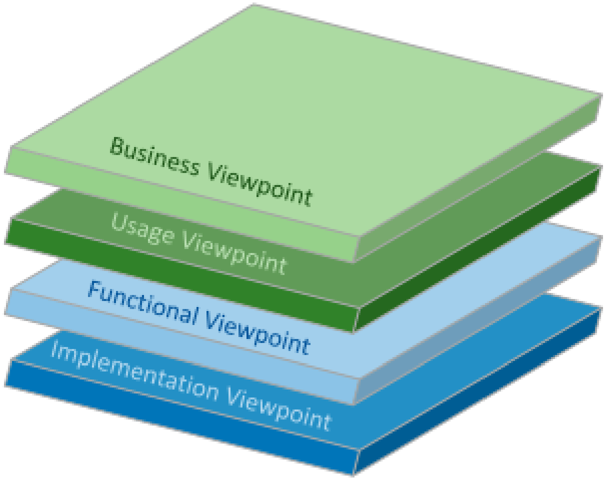
The need to develop a standard arose precisely as an effect of the continuous developments in the industrial sector and the rise of those digital tools, used in collaboration between stakeholders in SCs, known as the 'industrial digital platform for manufacturing', which gave rise to the need to find a common way of looking at things regarding the design of systems of this kind and the drafting of the documentation that accompanies them (117) ; These are therefore real guidelines that avoid personal interpretation and ensure that those working in the industry can easily understand, minimizing effort and maximizing output. The absence of standardised methods that bring together the scientific community and the stakeholders who will make use of these digital tools would make the situation even more difficult and confusing, undermining the possibility of collaboration between the various parties. Other reference models such as RAMI4.0 (Reference Architecture Model for Industry4.0) now provide a method for documenting IoT, smart manufacturing and big data analytics projects in their conceptualisation, design and use phases. That of IIRA is a true complementary model of RAMI4.0 (and this nature has been recognised by both consortia responsible for the two architectures); what is not yet possible today but will be ensured in the years to come is the total interchangeability and compatibility of the two (already RAMI4.0 is a standard that unifies all the previous ones used in the sector); that of complementarity is a fundamental aspect, especially for companies that have a globalised market and that currently, in order to meet the needs of the market and ensure their business, must participate in both standards. In (118) an analysis is made to find the complementary aspects between the two relevant architectures. As can be seen from the same article, the IIRA architecture is based on the 'ISO 42010 System and software engineering' standard (119) and is mainly based on 4 'viewpoints' (the description is based on the info gathered from the two previous articles mentioned above):

- *A) Business Viewpoint*

- *B) Usage Viewpoint*

- *C) Functional Viewpoint*

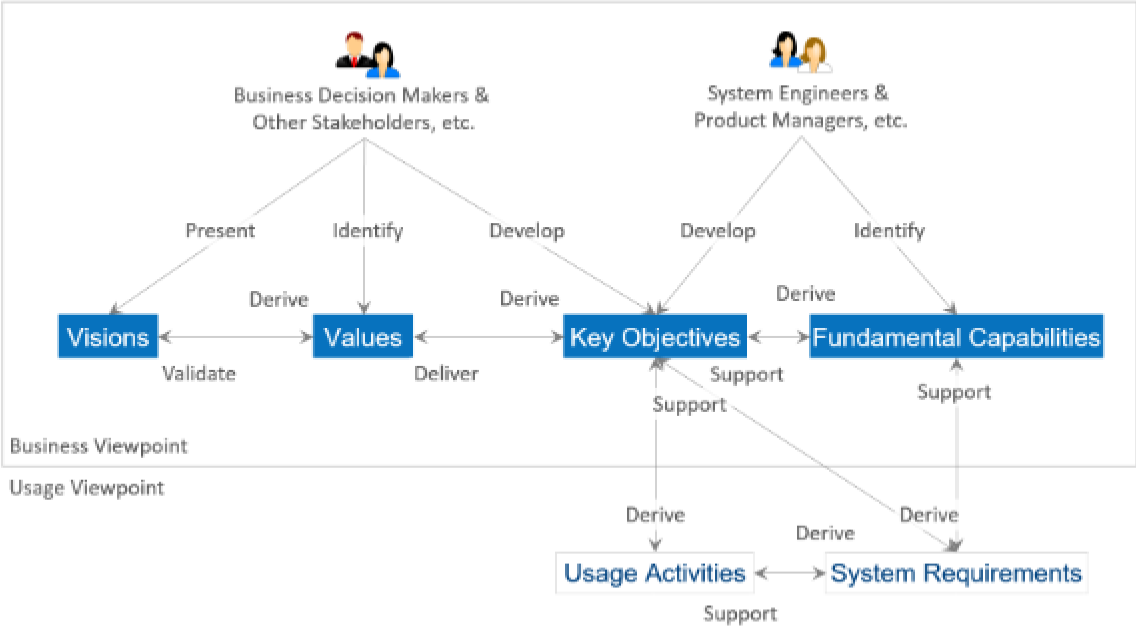
- *D) Implementation Viewpoint*



42 - IIRA Architecture (Industrial Internet Consortium Resource Hub)

A) The business viewpoint

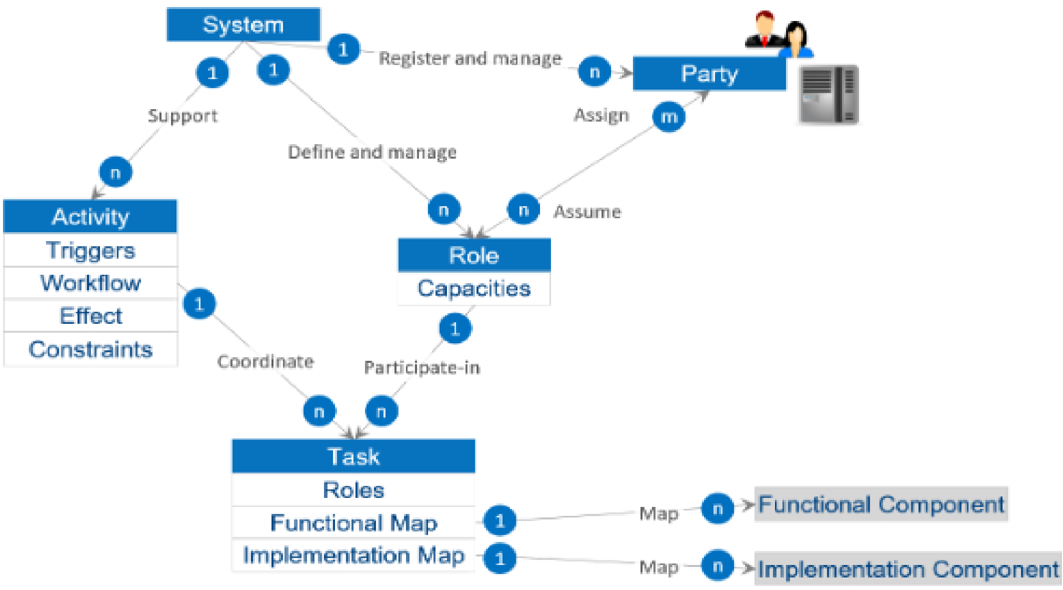
The business viewpoint identifies those stakeholders who are interested in the project in question for which documentation is being written and those who may be in the future (they drive the conception and development of IIoT systems in an organisation). Therefore, it is a fundamental step to be able to frame firstly what the operational context is and secondly what the potential of the project could be at a company level and outline its vision. Normally, the items to be defined for this viewpoint are stakeholders, vision, values, key objectives and fundamental capabilities. The "fundamental capabilities" are a review of the project's technical specifications and main functionalities, while the "key objectives" are a sort of KPI, measurable performance indicators that estimate the project's goodness according to whether they are achieved or how far they have come.



43 - Business ViewPoint 1 (Industrial Internet Consortium Resource Hub)

B - The usage viewpoint

It explains how to realise the "fundamental capabilities" showed in the previous viewpoint. Those capabilities are reached through a decomposition of activities in unit of works (single task) in order to provide a better control of the improvements. The activities focused on this viewpoint involve either human or logical users. This step has its main characteristic too:



44 - Business ViewPoint 2 (Industrial Internet Consortium Resource Hub)

The numbers and letters in the figure denote the quantitative relations between the elements, e.g. along the arrowed line from System to Activity, they denote a (1) System supports many (n) Activities; along the Party and Role arrowed line, they denote a Party (can) assume many (n) Roles; a Role (can) be assigned to many (m) parties.

- ***Task***: it used to name as task all these activities that allows to split the work into single actions. So they could simply be the invocation of an operation. It is important to define what or who is responsible of that action through the role it has.

- ***Role***: is a set of capacities assumed by human/logical entities. The role defines which tasks are expected to be execute (and which not) by the specific entities.

- ***Functional map***: map of functions or functional components of the task. Practically this step shows the input and the output of every single task previous defined.

- ***Implementation map***: describes the implementation component(s) the task relies on for its execution. If role(s) are associated with the task, the map also defines how these roles map their capacities to the component(s) and related operations.

- ***Activity***: it's a train of task required to realise a single process of an IIoT system. The activity is made by some defined elements:

- ***Trigger***: is the specific condition which initiates an activity. There is a specific "role" previously defined who is responsible of trigger activation.
- ***Workflow***: it consists of the specific organisation of tasks
- ***Constraints***: they consist of those characteristics of the system which need to be preserved as the execution proceed (such data integrity or data confidentiality)

The descriptions provided previously about Task, Role, Functional map, Implementation map and Activity are based on (119).

C - The functional viewpoint

Is an architecture splitting the IIoT system in its functional parts in order to provide an explanation about the interrelations between them and with other external systems. To realise a correct functional viewpoint is necessary to introduce a system decomposition into five characteristic functional domains which distinct functionalities belong to:

- ***control domain***: functions such ones realised by industrial control and automation systems belong to this one. Specifically, those functions could be reading data from sensor, applying rules and logic, exercising control over the physical system through actuators. The functional components implementing those functions are usually deployed very closed to the production environment.

- ***operations domain***: it is the domain which provide the management operations (monitoring, provisioning, optimisation. prognostics) for the control one.

- ***information domain***: is a functional domain for the managing and processing of datasets. Normally, it acquires data from the control domain and applies transforming, modelling and analyzing operations on them in order to generate a no-direct control on the physical system through the control domain. For example, "it provides to the changing of the output of an automated production plant based on condition of the facility, energy and material cost, demand patterns and logistic.

- ***application domain***: all the business functionalities related to the specific IIoT system are implemented by the functions collected by this domain.

- ***business domain***: it implements the business functional logic, so it supports the activities and processes related to all the business functions connected to the IIoT system such Enterprise Resource Planning, Human Resource Management, work planning and scheduling system, service lifecycle management and so on.

D - The implementation viewpoint

It contains all the technologies and technical issues needed for the IIoT system structure implementation. All the choices taken in this phase are strongly linked to the business viewpoint, which has previously defined the market strategies and constraints and making it meets the system requirements already planned is necessary. The implementation viewpoint defines three patterns for a coherent IIoT System implementation: three-tier architecture pattern, gateway-mediated edge connectivity and management architecture pattern, and the layered data bus pattern.

9. Project Business Viewpoint

Once the IIRA architecture has been studied from a theoretical point of view, it is necessary to try giving it a specific interpretation for the project in question.

9.1 **Business Viewpoint: Stakeholders**

The business viewpoint is a rather difficult set to frame in a concrete manner, as it is directly conditioned by the interaction network that the company has with the belonging market and according to its internal organisation. Remaining on an intermediate level of detail, one can first draw a picture of those who could be the toolkit's stakeholders by dividing them into two levels discretised according to whether they have a direct influence on toolkit's output or whether they are simply its users (levels 1 and 2) and then distinguish a final category made up of those who contribute strategically by providing the data necessary for programming the MRP (level 3 stakeholders).

Level 1 Stakeholders

The organisation's manager is the first stakeholder as he is the one whose economic returns may be compromised by inaccurate management of production planning and resource acquisition.

In the other hand, being a toolkit that concerns a very strategic phase of business planning, i.e. the issuing of purchase or production orders for raw materials/components that can then be used in subsequent phases of need, it is intuitable how it is necessary to have a figure who deals specifically with its use and manages this step, practically a Material Planner.

The material planner is certainly the most strategic figure in terms of MRP control. He does not deal directly with the implementation of management systems from an IT point of view but must have an in-depth knowledge of how these work, how they can be managed in order to optimise their use and the leadership to take some managerial decision in its competencies. Knowing the planning methods, he is the one who must guarantee the integrity of the connections within the supply chain by interacting with various figures both inside and outside the company. If we look at the case of the project under consideration, the material planner decides how to set the programming constraints, which require some changes and how the problem elasticity coefficients should be designed, based on his knowledge and experience and the company's historic. He must be able to perform the function of glue between the various departments of the company as the inputs of the MRP programming have different sources, from the sales / purchasing department to the production department up to the warehouses.

Another figure that belongs to this "class" is that of the software engineer, whose knowledge is necessary to transform the needs of the designer and the company into project specifications. It has the task of evaluating and choosing the technology/approach that can guarantee the best results for the organisation and, in terms of cybersecurity, its contribution is functional to ensuring that the use of management systems is not undermined by internal factors/ external factors that can damage the planning result. Based on their skills, the software engineer could manage and improve the architecture of the toolkit

Level 2 Stakeholders

The requirements planning carried out in the previous phase ends with the generation of production orders (in case the components/sub-assemblies are produced internally) or supply orders (if it is necessary to process the purchase of products from suppliers to have a restocking of stock levels).

Once the orders received and the sales forecasts have been evaluated, it is possible to determine (through an "explosion" of the bills of materials of the items present on the market) what are the dimensions of the purchase and production orders for each individual product code. Whoever manages plant and production resources has the task of evaluating the availability of these resources (data provided in part by the management systems that track warehouse levels and in part by the material planner via MRP) and of the production capacity available in a certain time horizon to be able to compare them with those necessary to fulfill the orders. The output of his activity will be an optimal production scheduling.

Beneficiaries of the needs planning are also the human resources who deal with the external/internal logistics aspects: the former organise how and when the materials and components will enter the company while the latter will take care of their best allocation within the warehouses that must be monitored frequently depending on the strategic value of the product.

The human resources that deal with the internal/external logistics aspects are also beneficiaries of the scheduling of requirements: the former organise how and when materials and components will enter the company while the latter will ensure that there is the best possible allocation of them within the warehouses that will have to be monitored with greater or lesser frequency depending on the strategic value of the product.

Level 3 Stakeholders

Those involved in the management of the master production schedule play a key role. The MPS is drawn up primarily based on orders received/planned and forecasted orders with various forecasting tools. The output of this schedule becomes the input of the MRP: by choosing to pursue a Lean philosophy, it will seek to minimise the number of items in stock and maintenance costs. Thus, the goodness of the MPS output is certainly a key factor for the MRP one.

The engineers and programmers involved in sales forecasting have an important responsibility because as the quality of the forecast for output increases. there is a reduction in the probability of mismatch between what is to be produced and the number of resources available to do so.

In conclusion, thus the clients could also be defined as indirect stakeholders of the toolkit. By clients is meant, depending on the type of product the company makes, those who will be the end users (people, other organisations). The client can influence, by contract, the level of service to be rendered to it (time, quantity, quality) and is therefore capable of placing very tight constraints on the planning of requirements.

The following table provides a schematic representation of the level of stakeholders:

STAKEHOLDERS	LEVEL 1	LEVEL 2	LEVEL 3
	-Comp. Direction - Material Planner -Software Engineer	-Production Manager -Logistic Managers -Procurement Department	- MPS planner - Technicians/Programmers - Clients - Suppliers

Table 2 - Stakeholders levels

9.2 **Business Viewpoint: vision, values, key objectives, fundamental capabilities**

Below there is a description of the four points above in reaction to the stakeholders with the most strategic role among those already described. Regarding the definition of some of the items described, it is worth noting that there has been an attempt to hypothesise them, since there is no in-depth knowledge of the operational and organisational context of the company. For example, the "values" are typically identified by senior business and technical leaders in an organisation.

1) **Company Direction**

Vision: intelligent MRP management and increased supply chain flexibility

Values: the toolkit must provide an "Intelligent" method to be able to reduce management difficulties regarding requirements planning, while increasing the organisation’s flexibility with respect to sources of uncertainty.

Key objectives: Material requirement planning is a type of planning that could be managed by imposing different objectives. From a generic corporate point of view these could be:

- Minimisation of the stock size of materials/components in the inventory
- Reduction of management errors with repercussions on profit
- Reduction/limitation of disruptions on the supply chain
- Ensuring a high level of customer service
- Reduction of non-value-added activities

2) **Material Planner**

Vision: to improve requirements planning while reducing errors.

Values: given that the planner's experience plays a key role in the functionality of this project, it is important that the tool can guarantee easy manipulation of the model structure according to the needs and situations that arise, allowing planning errors to be minimised.

Key objectives:

- Reduction of errors in production planning/purchasing of raw materials and components
- Improved interaction with the warehouse, production scheduling, sales office
- Improved control of deadlines and time management
- Better control of productive resources
- Minimizing of backlogs and wastes
- Better control under demand nervousness
- Ensure high service level.

Fundamental capabilities: using the toolkit, the material planner will be able to have better co-ordination with the various departments of the company and be able to have a major impact on the company's finances. The use of artificial intelligence combined with optimisation algorithms should lead to a reduction in both planning errors committed and supply chain disruptions.

3) **MPS planner / Production Manager**

Vision: to generate the most reliable production plans possible for fairly long-time intervals, so as to release the main input for material requirement planning and purchasing orders. Managing production in such a way that the market requirements (known from the MPS), the company's resource availability, and the warehouse management system (Production Manager), overlap perfectly.

Values: in this circumstance, the value added by the production management tool is not directly noticeable, although it is strategic. The detailed production planning (the one intended to cover the short-term demand) is completely dependent on the material requirement planning management, since it will be performed in such a way as to utilise the resources currently present and reserve production capacity for that part of the stock that is currently not "on hand" but has been planned, anyway.

Key Objectives:

- Ensure that there is the allocation of the required amount of resources within the required timeframe for the fulfilment of foreseen and planned orders.

4) **Procurement Department**

Vision: To ensure there are no errors in fulfilling outbound replenishment orders and arranging inbound customer orders. Management must be carried out as streamlined as possible, guaranteeing the integrity of the information flow and making sure that all information of a strategic nature can reach the material planner (for example, the presence of any problems with suppliers or with external logistics, interruptions of the supply chain etc).

Values: The toolkit should provide the sales department with the ability to process purchase orders without leaving room for data interpretation, providing the most intuitive data possible and limiting the need to make decisions.

Key objectives:

- Reduction of management errors
- Ensure orders are on time/on schedule.

10. Project Usage Viewpoint

In this part are defined the strategic roles of the stakeholders previously introduced in the "business viewpoint". They can be described as sets of actions that the stakeholders can perform. Since the project deals with the implementation of a management toolkit then the concept of "role" can also be a way to constrain access to the tool, creating a way to manage cybersecurity aspect: in this way, depending on the role, limits of accessibility to sensitive planning data can also be defined. For example, it can be assumed that the material planner would have total access to the parameters that alter the planning while the company's purchasing department might not have the same credentials but rather reduced access to only the planning outputs, in order to organise purchase transactions (119).

10.1 **The Roles**:

as far as roles are concerned, it could be made a subdivision of these into categories, each of that has a specific set of responsibilities to be assigned to who is part of it. Given that the unique subject of study is an algorithm itself, a broader scenario on which it could be used has been assumed within a support tool for the organisation management. Thus, it's automatically that the concept of role is linked with the possibility of having different access keys to sensitive programming data.

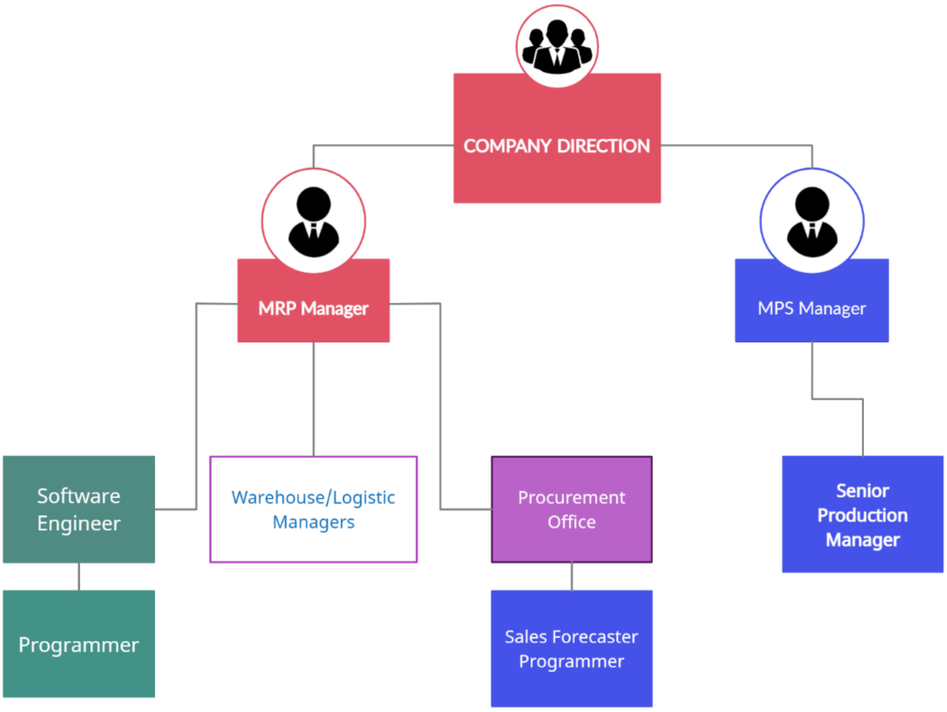
The categories of roles, ranked using colors, are the following (nb. roles are defined in relation to the project algorithm only and have no "real" validity in the hierarchies of the organisation):

-***DECISION-MAKING*** *role* -> “red level”: this is the role held by the person(s) responsible for planning and empowers them to make strategic planning decisions. In the case of the algorithm under consideration, they are the ones who have the final say on the structure of it and how its hyperparameters should be handled. Obviously, this role is defined as such in relation to the algorithm itself, since it is clear that goal setting is a power vested in the top management of the organisation. With these assumptions one could define two different levels for this role, depending on the decision-making power they have access to (lvl.1: organisation management; lvl.2: MRP manager)

-***TECHNICAL*** *role* -> green level: by the name of this role, we mean that the person who possesses it, can intervene in modifying the structure of the algorithm by making variations in order to meet the needs of the planner and the organisation. It is practically the role possessed by a software engineer or computer programmer, who cannot deliberately make his own decisions on the matter but follows instructions given from higher roles. The accessing to sensitive programming data is necessary in this case, in order to intervene on the algorithm structure.

-***COLLABORATO****R role* -> blue level: this role belongs to those entities contributing strategically in defining and providing programming inputs. Just think, for example, about the office responsible for establishing demand forecasts and trends in the market or those who manage production and maintenance scheduling, which would provide data about the health and availability of production resources. Following the same reasoning made for the decision-making role, an architecture organised in levels could be adopted here too (lvl.1: MPS manager; lvl.2 : production manager, maintenance manager).

-***USER*** *role* -> purple level: whoever possesses this role is a figure who may operate within the organisation or be part of its supply chain. The “user” has completely different access keys than those possessed by previous roles. One "user" might be the company's purchasing department, which exploits the output of programming (assisted by the algorithm) in order to organise the acquisition of raw materials and components. Those responsible for logistics and warehouse management themselves can use the output, for example, to make layout changes in such a way as to accommodate incoming items in the most optimised way possible for who manage products internal movements. The creation of a common data cloud among supply chain actors would make the suppliers of the organisation other possible users of the output of this algorithm so that, given the customer-consumer relationship, they could have a solid data for their own internal planning.



45 -The roles

10.2 **Activity Definition**

For the completion of the next step,find strong the key activities definition is required. Every company's figure introduced has a set of activities to take into account:

Decision making:

LVL. 1: *company director*

- definition of short/long-term market objectives
- conversion of objectives into tasks to be assigned to roles in the organisation
- definition of strategies

LVL. 2: *MRP Manager*

- ensure the availability of raw materials and components to meet market needs
- determine, based on objectives, the best planning strategy
- ensure correct data flow, in the required time and quantity, from other roles

Technical: *Software Engineer, Programmer*

- ensuring that MRP programming requirements are converted, in a functional manner, so that the software tools used can assist in their management
- choice of technology that, according to the organisation's investment possibilities, allows the maximum return
- algorithm creation and parameter management
- data flow management

Collaborator:

LVL. 1: *MPS manager*

- Ensure that production and resource availability requirements are transferred, on time and in the required manner, to purchase order and production scheduling
- knowledge of scheduled production interruption activities (e.g. maintenance) as their duration imposes different constraints in MRP management

LVL. 2: *Production manager*

- control resource availability (downtime, breakdowns, production time)

User:

- ensuring material resources on time to meet the needs of the MPS plan and market demand.
- communication with customers/suppliers.
- organising business layout to optimise space and reduce handling time.

11. Project Functional Viewpoint

11.1 **Generic Features**

Given the high frequency of use of the algorithm, the fast pace with which various tasks must be completed within an industrial production context, it is essential that the structure of the toolkit ensures certain characteristics (summarised in the image [44]):

- **interpretability**: the algorithm itself, the constraints and the data that populate it, must be easily interpretable in order to optimise processing times in the event of anomalies. The quicker the intervention, the lower the economic losses for the company.

nb. with interpretability we refer to that attribute needed for the figure who deals with the management and manipulation of the algorithm, not for the users which deals with the output only.

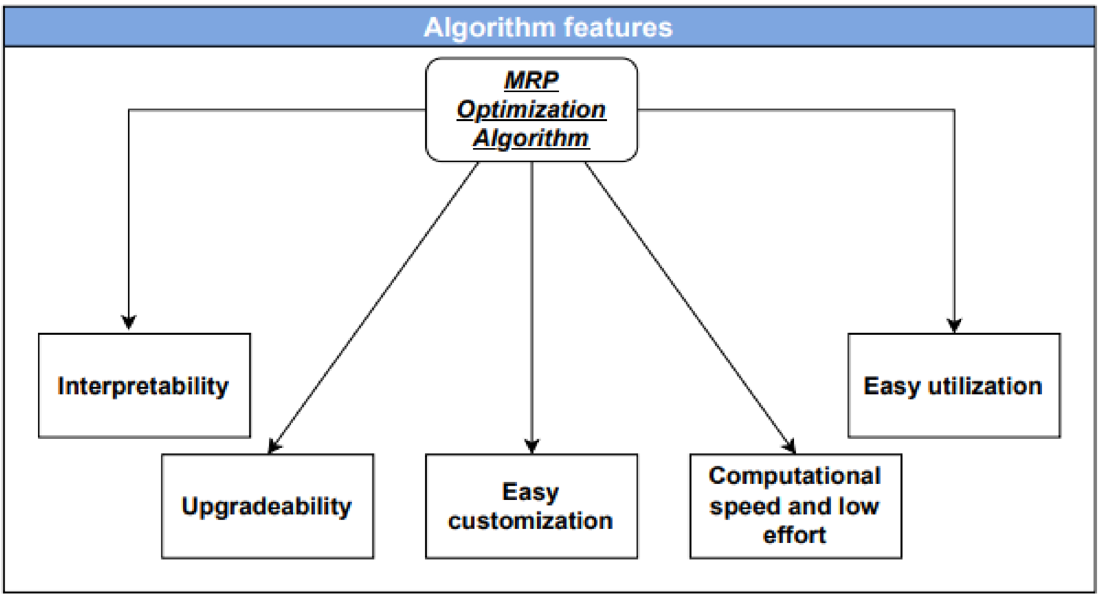
- **upgradability**: the algorithm must be updatable from the point of view of the optimisation method it uses; it must therefore guarantee the possibility of integrating new functions that can "enhance" it

(Use of different solvers for planning, control of membership functions and elasticisation coefficients which will be presented later)

- **customizable**: the algorithm must have a structure whose inputs can be easily organised and customised by the company based on the type of products it produces, internal choices for the type of production or management strategy; therefore, it must not have a rigid architecture applicable to a single context as the MRP is already a highly versatile model itself.

- **ease of use**: the complex mathematical formalisms that make up the structure of the algorithm must not compromise its ease of use. The programming inputs must be organised in data structures that allow the automation of the flow of information, minimizing the possibility of making mistakes if managed "manually". Choosing a user-friendly computer language would help this aspect as planning requirements could be translated easily into technical specifications for the algorithm.

- **speed/ low effort**: contextually with the domain and the dimension of the problem, the algorithm must not be "energivorous" in terms of computational expense that requires and must process an output in a reasonably short time. This aspect is important as the algorithm is used and updated with very high frequencies.



46- Algorithm functional features

11.1.1 **How to ensure the features**

Interpretable and Upgradable algorithm

In order to ensure that the algorithm would be easy to interpret and improved, it would be advisable to opt for an intuitive programming language closer as possible to the "natural" one and that makes available libraries that can gradually be inserted within its structure so as to improve its characteristics. The factors previously listed find a strong correspondence with the Python programming language (127), as it is one of the most user friendly. The use of fuzzy logic within the algorithm will be introduced rather "implicitly" but nevertheless, even for Python, libraries have recently been developed that allow the creation of real inference engines that exploit this logic, such as “Scikit fuzzy” and "Simpful” (128).

Easy to customise and Easy to use

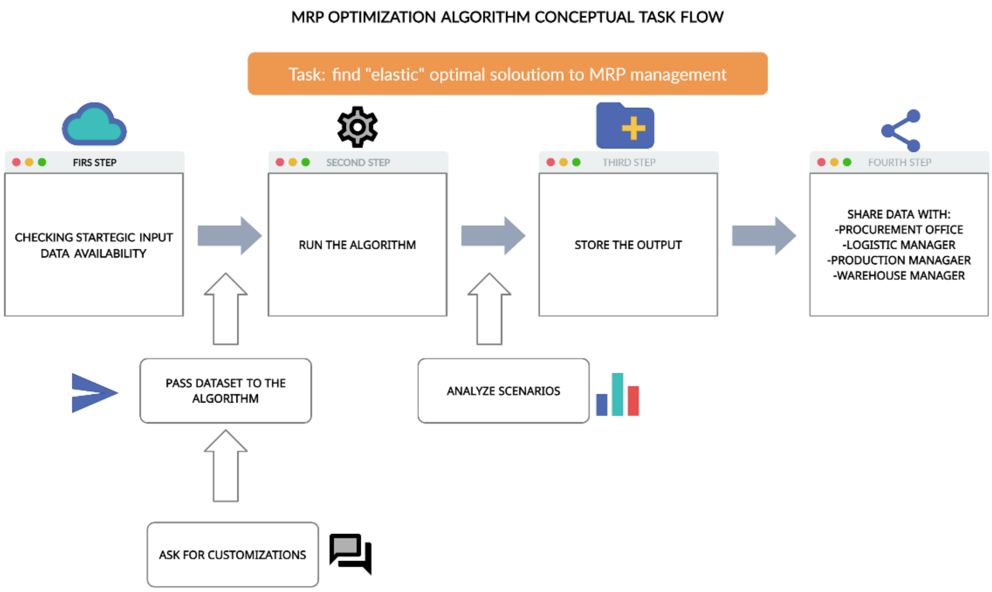
Making the algorithm easily customizable is a specification that should be imposed early in the design phase by going to create an architecture that is as versatile as possible depending on the variability of the contexts for which it is to be made applicable. Since this is a method for optimizing material needs planning activities certainly goes in favour of this aspect, as it is itself a management model that does not have a very "sectoral" character.

As for utilisation, this needs to be improved in terms of ease: collecting input data, processing them, and managing parameters are activities that should not require excessive cognitive effort. The algorithm must anticipate what might be the errors made by the user and guide him or her with suggestions that ease the process. As far as the possibility of intervention by an external user is concerned, it must be made clear that this does not have to be able to work on the more mathematical aspect of the algorithm (i.e., go changing the type of membership functions used or the aggregation and defuzzification methods) but understand the effects related to the manipulation of the hyperparameters of fuzzy programming.

Speed/Small Computational Effort

From this point of view, the goal is to make the algorithm as performing as possible in terms of execution speed and executable on hardware systems that are not necessarily very performing. These two characteristics would also help to make it applicable in different contexts and not to transform its use into a bottleneck activity. How quickly the algorithm generates output depends heavily on the size of the problem (for example, how many weeks are scheduled through a single run of the algorithm); with the same size, different effects could be had by managing the problem-solving system (therefore the solver employed).

11.2 **Algorithm conceptual task flow & Architecture for Runtime**



47- MRP optimisation algorithm conceptual task flow

What is depicted in the figure [47] is a (very simplified) conceptual model of the succession of tasks from input management to output management of the algorithm.

Step 1)

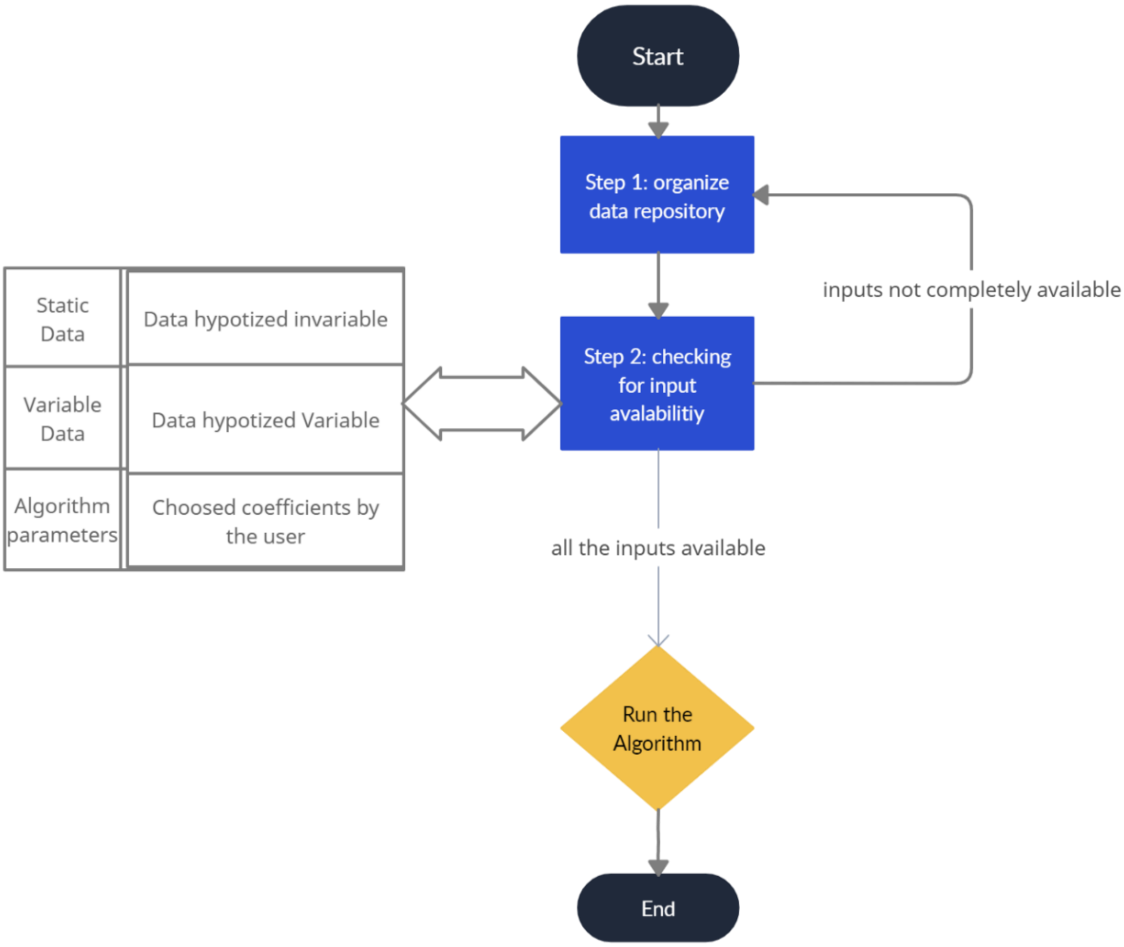
The first step that the system must perform is to verify the availability (and existence) of all the input data needed in order to feed the algorithm and, if necessary, to tell the user if one or more are actually not available. There's not a specific format required for the data passed as input (but it would probably be the csv or xlsx).

As for the input data, one can imagine dividing them into 3 categories:

- static data: belong to this category those data that can be assumed to be constant over the planning horizon (such as production/purchase cost per unit of product, maintenance cost, and even the cost related to the use of production resources can be considered as a non-variable data along the considered time horizon).

- Variable data: these are data that are updated each time the algorithm is launched. Since this occurs weekly, compared to the previous (and next) one will almost certainly see a change in demand, orders and penalty costs for product demand backlog.

- elasticisation coefficients: when the algorithm is first launched, the user must choose these coefficients to determine the degree of elasticity of the planning constraints (such as the target cost). The coefficients remain constant throughout the planned week, but due to their dinamic nature, they will change at the beginning of the next week to be planned.



48- MRP input organisation

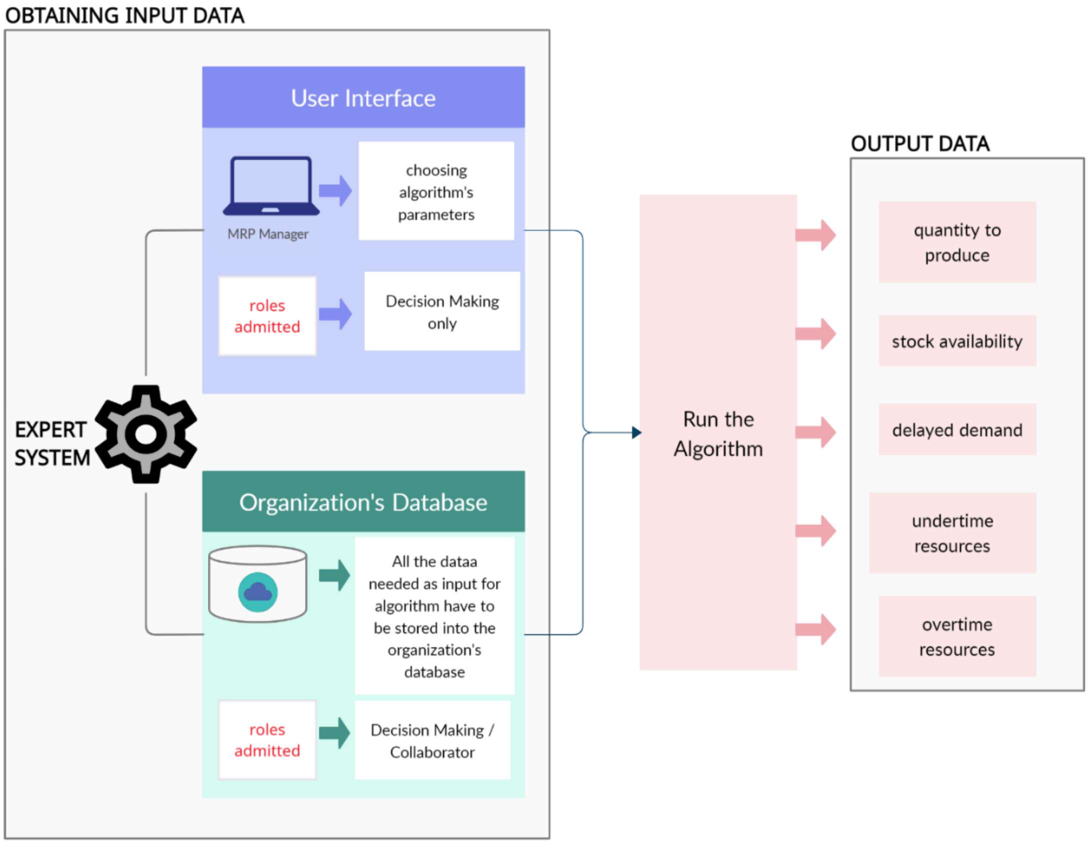
Step 2)

The program must verify that all schedule inputs are present. If this is not the case then, via user interface, it must be able to communicate the source of the anomaly and postpone the first step

Step 3)

This step, which corresponds to the launch of the algorithm, can be triggered when the condition of "all inputs available" has occurred.

The one displayed so far, in image [48], represents the logical path followed by the algorithm, and according to that, it can process an output for programming raw material, component requirements, stock levels and used resources. At this point, however, it is necessary to provide further explanation of how the input data is made available. From now on, reference will be made to the functional diagram in the next image [49].



49 - algorithm functional diagram

There are two classes of data that feed the planning process: those chosen by the user and those available to the organisation. On the second class it is not necessary to dig deep as it consists of all those data that the company monitors continuously and that are updated with high rotation: we are talking about expected demand values, resource availability, the level of stock in the warehouse, confirmed orders, production and stock maintenance costs, etc., so depending on how these are stored and maintained it will be necessary to equip the algorithm with "tools" to access and read them (most of the time the formats are standardised). Normally it is the expert system itself that performs this function of through data, making it always available for programming. For the purposes of the project under consideration, however, it is the other category of data that is of greatest interest: we are talking about the coefficients of the algorithm that condition its degree of relaxation and whose value strongly conditions the final output. To say that these coefficients are deliberately chosen by those in charge of performing MRP planning on the basis of their own experience would be a mistake, yet one is not actually far from this condition. Choosing these key elements is something that can be done by a "fuzzy inference system," i.e., a system consisting of an inference engine that performs a series of inferences on a data set in order to provide an output answer, and it is the logic that this uses that can be built based on the planner experience. As already explained an expert system that uses fuzzy logic exploits its deductive capabilities that are the result of externally planned rule sets that condition it in the way it operates.

In the article (114) it is clarified that these parameters are the result of choices within the organisation and that, once identified, they are kept constant in the planning window. An inference engine, on the other hand, could change their magnitude by virtue of changes that may occur over an expanded window of time, making the algorithm more dynamic.

For example, in the hypothesis that the organisation plans its needs according to a cost target, or even a "target range" in the case in question, then there could be the need to narrow/broaden its boundaries as a consequence of the occurrence of certain conditions: assuming the case where the company receives an order from a customer that imposes a very high penalty cost on the single unit of product that is not delivered on time, at that point the organisation (through the inference engine) might decide to act with an increase in the size of the target cost range, thereby accepting that the decision variables take on values that result in a cost that is higher than the company would normally incur, since the analyses performed would show a more cost-effective consequence when compared with the costs that would be incurred in the other scenario.

12. Project Implementation Viewpoint

To have an overview of how the algorithm can make its contribution and interact with other "virtual" entities, we should first understand how it could be integrated within a "sophisticated" structure that allows a concrete answer to the problems posed by the AIDEAS project. For this reason, there is a specific architecture called “Architecture for AIDEAS AI Runtime”, it provides, in a basic way, a concept of this structure.

A number of strategic levels are identified:

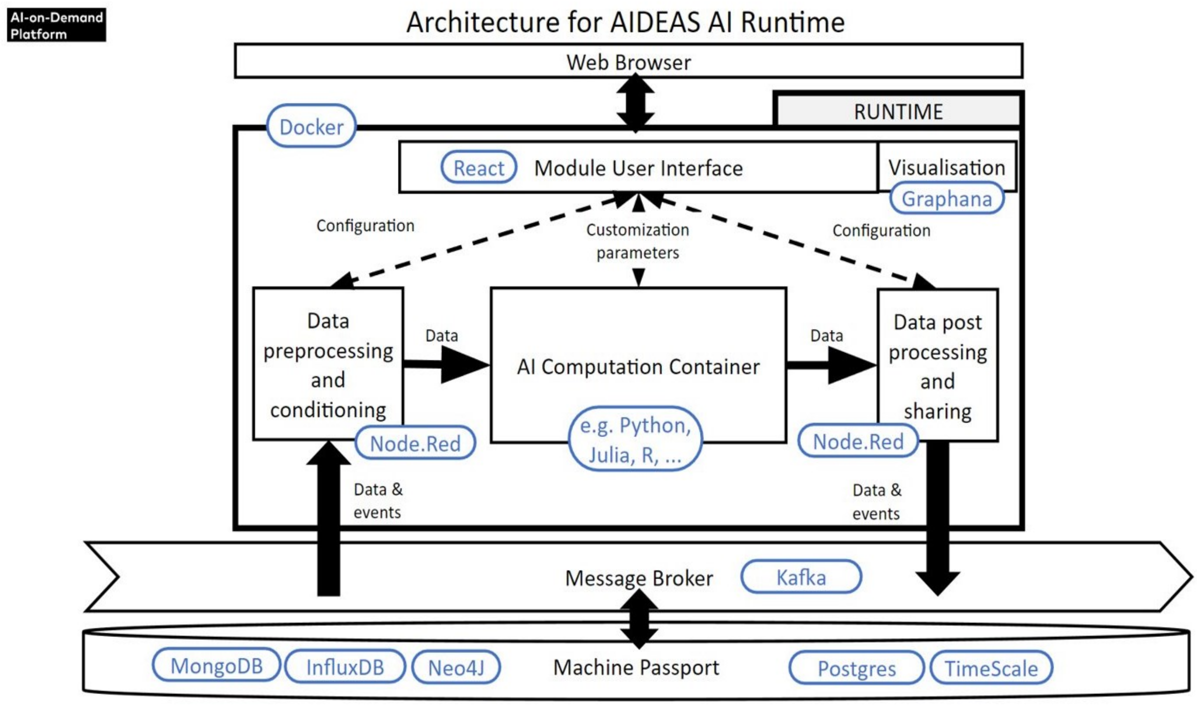
-1) the RUNTIME SYSTEM: understood as a kind of software that makes available to a user everything needed to run an application. In this case, the backbone for this function is that constituted by Docker. It is precisely within Docker that we can imagine the algorithm that is the subject of this study, more precisely in the "AI Computation Container." Embedded in this sphere of the runtime system we find mutually connected sub-modules:

- 1.1) the "Module User Interface" (REACT): constitutes the interface through which the user can exercise a certain set of actions
- 1.2) "Visualisation" module (GRAFANA):
- 1.3) "Data preprocessing and conditioning" module (NODE, RED):
- 1.4) module of "AI Computation Container" (PYTHON, JULIA, R...):
- 1.5) module of "Data post processing and sharing" (NODE, RED).

- 2) the MESSAGE BROKER, in this case Apache Kafka; it serves as the data distributor

- 3) the MACHINE PASSPORT.

  - -3.1) MongoDB
  - -3.2) influxDB
  - -3.3) Neo4J
  - -3.4) Postgres
  - -3.5) TimeScale



50 - Architecture for AIDEAS AI Runtime

Below will be a brief description of the technologies reported within the architecture.

Runtime:

- Docker (wb-7)

51 - DOCKER logo

What one reads directly from this software's website is quite clarifying to understand what its usefulness is and to get an idea of how the project under consideration may relate to it. Docker is an extremely functional tool for those building applications, simplifying the steps from building them to sharing and using them, thus enabling optimised and agile management of their entire "lifecycle." All this in order to make up for the shortcomings and defects of the tools used before its introduction.

In it is described as a tool created with the primary goal of facilitating the sharing phase (of an application or code) between Developer and End User. The idea behind the open source "Docker" project is to be able to automate the distribution of a software or application in the form of "self-sufficient" containers (containers). A Docker container will have inside it everything that the application needs in order to be used, so it will contain very diverse items ranging from libraries, scripts and the various appropriate executables, which will in no way interfere with the host operating system. One of its strengths lies in its flexibility, since it was designed as a tool aimed at use by developers who could work on any workstation, care was taken to ensure that it could be compatible with any operating system.

Previously to Docker, it was usual to work with a hardware-level virtualisation approach (virtual machines) but the advent of this tool allowed a departure from the need to use a virtual machine by making containers "executable" all on the system hosting the programming. Another key aspect of Docker is whereby the various containers are deployed as "images" so that effectively a single object can contain what is needed to run a virtual environment that can be replicated by hosts on different environments; these images are organised on "Layers" and allow easy customisations since is possible to decide to modify them starting with a specific layer and keep all previous ones as such (wb – 8)

For realistic examples to capture the true potential of Docker, see the official documentation (wb –9).

-React (wb –10)

52 - REACT logo

This is an open-source framework and library developed in JavaScript by Facebook. Its main purpose is to help with user interface developing for an application; its main strength is the dynamic and flexible programming it offers, in that the application can be thought of as composed of many small units (they can be programmed separately and independently, and reusable as individual blocks) that joined together create a single "entity" (wb – 11). The specific definition given by the official documentation is “React is a library for building modular user interfaces” (wb – 11).

- Grafana (wb-12)

The definition of Grafana that is given by most is that it is a web application whose main purpose is to make possible interactive data visualisation and analysis. Grafana is an open-source and compatible with major operating systems and works very well with InfluxDB. With Grafana it is possible to make real-time graphs based on time series of data through its customizable dashboards (wb-13). More information about the benefits offered by Grafana and many examples can be found in the official documentation (wb – 14)

53 - GRAFANA logo

- Python (wb – 15)

Python has two main characteristics: is an object-oriented and high-level programming language. The former means this language is based around objects (such as data) rather than functions (so the opposite of the C language, for example), and the latter means it's easy for people to understand. Python's is currently one of the most widely used programming languages, given its great versatility in different application contexts and the high intuitiveness in its use it is definitely seen as one of the most appreciated. The possibility of automating tasks and doing data analysis in a very simplified way are the principal features that justify its success but there is no lack of cases where it is used to build software and websites. For those reasons it is considered as a general purposes language. What makes it really user-friendly is the simplicity of its "grammar," very close to the "spoken" language, and above all the presence of a very large number of libraries that nowadays meet the needs of any user, even the most disparate. "Data analysis" is the main context of use for Python given it's ease at work with complex statistical data, multidimensional matrices, representation and visualisation of data (even in a three-dimensional way), algorithms for machine learning (for these reasons, it is often employed for AI contexts). As anticipated, the scripting (i.e., code writing) phase is facilitated by the availability of innumerable

54 - Python logo

documented libraries whose use lends itself even to less experienced programmers; other merits to note are that of extensibility (i.e., the fact that Python has the exact same interface on a wide range of hardware).

- Node JS (wb – 16) / **redis** (wb – 17)

55 - NODE js & Redis logo

Node.Js, or simply "Node," is an execution environment that allows Javascript to be executed like a normal programming language by making it possible to execute any type of program (wb – 18). It’s a platform built on Chrome’s JavaScript runtime and is perfect for realise network applications used to work with a lot of data in real time (129)

Message Broker:

- APACHE kafka (wb – 19)

56 - KAFKA logo

Kafka offers itself as the answer to a problem directly related to the challenges of Industry4.0 and modern industrial organisations, namely that of big data analysis and analytics. Kafka can be seen, in a very simplified way, as a platform that enables the streaming flow of large masses of data coming from different sources, to be distributed at one or multiple destinations; it is perfect in contexts where scalability and efficiency are required in these data transmissions (130).

More technical information and real application cases can be accessed on the official documentation (wb – 20).

Machine Passport:

- mongoDB (wb – 21)

57 - mongo DB logo

in a nutshell, MongoDB can be seen as a database management system designed to make application development more dynamic and rapid. It uses JSON as the way for inspire its data structures, named BSON (131).

As described in (132) , MongoDB facilitates data access operations.

- influxdb (wb – 22)

58- influxdb logo

This one is another open-source database, programmed in "Go" programming language, created for heavy loads time series data management (133). One of the best benefits given by InfluxDB is that it can aggregate values with no manual interference in moment buckets on - the-fly. A visualisation tool such Graphna (already cited) can access InfluxDB (134). Its schematic architecture is available in (135).

- NEO4J (wb – 23)

59 - neo4J logo

Neo4J is practically a graph database. The first difference between other ways to represent data is that, this type of database isn't made by columns and rows but by nodes. As reported on the Neo4j website (wb – 24)

There are a few advantages gained by its using:

- High Performance Thanks to Native Graph Storage & Processing

- Easy to learn and use
- -Highly Performant Read and Write Scalability

Every time working with connected data is needed (as in such many industrial's scenarios) , using a graph database is a good choice since its offers best performance (in agility and flexibility), thus allowing to know data value, provenience and their relationship type (127).

- POSTGRESQL (wb – 25):

60 - POSTGRESQL logo

As suggested in (128), PostgreSQL is a "objects relational database". It is open-source and it could work correctly with all the operating system. Postgres allows to create a database in the backend of an application (data storage), and with the amount of data that modern ones work with it is certainly an advantage together with its scalability in the number of users and amount of manageable data and flexibility (in the support from Unicode to complex multi-byte character encodings, different programming languages and offer the chance of create functions and trigger) (wb – 26)

- ***Timescale*** (wb – 27):

61 - Timescale logo

It is an open source "Time Series Database" whose main purpose is to allow data to be stored in the form of time series; this feature is certainly useful in the industrial context, where data collection (mainly from sensors) in production processes can be segmented with respect to specific time intervals, serving as an effective assistant for monitoring (wb – 28).

12.1 **Proposed AIDEAS Solution Technical Features**

AIDEAS SOLUTION TO BE DEVELOPED	MRP optimisation algorithm with uncertainties management (fuzzy approach)
How the algorithm is supposed to work	The solution consists of an optimisation algorithm resolved by linear programming which includes an "uncertainty" controller realised with the aid of a fuzzy logic inference engine
TECHNOLOGIES	Python programming language
TECHNICAL FEATURES	**Dependencies**
	- **development language**: Python - **editor**: Visual studio code - **libraries**: puLP , scikit-Fuzzy, Openpyxl, Pandas - **container**: Docker - **operating system**: adaptable - **problem solver** : "CBC" (Coin-or branch and cut) is an open-source mixed integer linear programming solver written in C++ developed by IBM Research
	Interface
	- **user Interfaces:** React - **data Repository:** Excel - **programming :** Visual Studio Code editor
	Requires
	Other external solutions: **-** MRP : Deterministic formalisation

Table 3 - Features for the proposed AIDEAS solution

13. Project proposal

The idea sought to be developed through this project concerns the creation of an algorithm that can be an aid tool for those performing requirements planning of raw materials and components for their organisation. Its implementation is carried out in a "hybrid" manner by exploiting together both fuzzy and linear programming logic (previously introduced and explained); in this way, one can aim at a raw material and component requirements planning approach that proves to be "more flexible" for a productive “multi-product, multi-level, multi-period” environment, thus characterised by possessing the following requirements:

- *"multi-product" environment*: this property refers to the fact that, the application of the algorithm, can take place under different production conditions; thus, a manufacturing environment with differentiated production.

- *"multi-level" production*: referring to the fact that the finished product can be seen as the composition of several items, divided into levels. This characteristic is typical of products that are obtained by assembly (e.g., automobiles, industrial machines, household appliances) and that have rather complex BOMs. Production can be seen, many times, as the union of several sub-assemblies.

- *"multi-period" scheduling*: defined in this way precisely to indicate that scheduling does not just cover a single production period.

Obviously, with its actual form, the algorithm, is not ready to replace more rooted MRP planning methods, but it could be a viable alternative that would offer the possibility of evaluating different scenarios by providing greater elasticity to planning, especially in contexts where the application of a deterministic model would not reflect actual working conditions. One could envision the use of the same in a combined manner with an expert system, in which the knowledge of experts in the field is brought together, allowing for context-specific and context-customised optimisation of the solution to this type of problem.

In a traditional decision-making problem, such as the one under analysis, there are some recurring ingredients:

- a set of possible alternatives (different scenarios for solving planning requests)

- a set of constraints in order to limit the variability of the former

- a function that allows to evaluate the goodness or not of the alternative (function that will be minimised or maximised).

In the case of material requirement planning, there are several factors that have a "fuzzy" behavior:

1) the target: the planning objective always remains that of minimizing management costs, but the approach used in order to achieve it could change; it is difficult for the planner to know precisely the value of the weekly target.

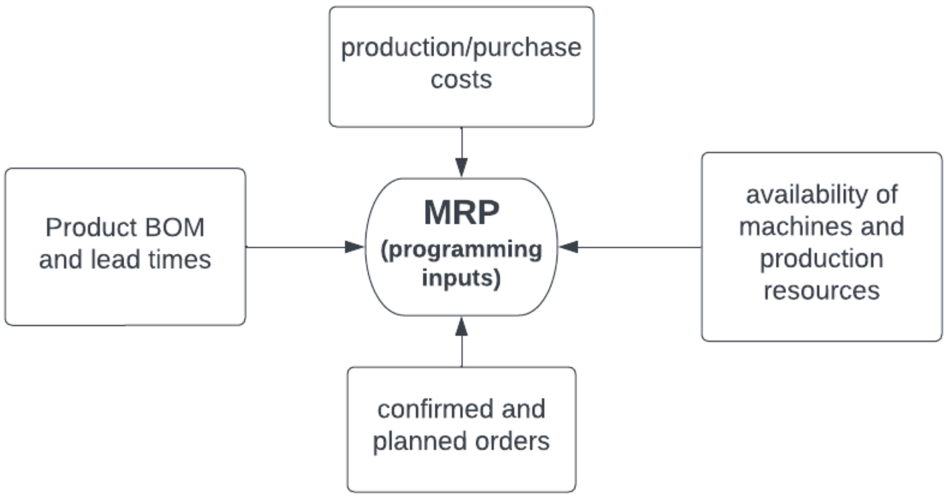
2) the availability of productive resources: there is a certain maximum daily capacity that the company can support, and its extent depends precisely on the working strategy adopted (for example, it can vary according to the number of daily work shifts). On the other hand, this certainty is contrasted by the occurrence of unpredictable events (accidents/breakdowns/interruptions) that could cause a decreasing, by a more or less large amount depending on the relevance of the event, of the expected production capacity value.

3) the demand value: since market is uncertain by default, the demand could have uncontrolled fluctuations that are too dependent on many external factors; for these reasons, forecasting tools cannot be completely relied upon, especially if they are used for long-term forecasts.

At this point, if those involved in planning use a classic linear programming approach, they would have to know precisely the value of the target that would guide the solution process, thus we could expect that a deterministic algorithm would be able to answer a question such as the next one: what is the value of the decision variables (quantity to be produced/purchased, number of hours of internal and sub-contracted work, etc.) that would guarantee the achievement of the minimum projected cost? But since within organisations there is a large sharing of verbal information (despite the possibility of using communication systems supported by data analysis that would make the circulation of timely information easier) these will always be characterised by a certain degree of randomness, fueled by the nuances that the "linguistic variables" are capable of conferring on a datum, especially when the information to be transmitted is quantitative (constraints and goals are linguistically formulated). This scenario therefore makes it difficult to know how to quantify certain quantities such as, for example, the target cost for a week of planning and consequently an algorithm that combines linear programming with the use of fuzzy logic could prove to be a valid solution since it would automate the process decision-making of some parameters but, whoever uses them, should be able to understand which are the factors that most condition it.

A company could choose, for example, to use a raw material and component requirements planning method with a cost reduction philosophy and could, depending on the planning horizon to be covered, choose the maximum cost not to be exceeded in function of the size of the market demand for that same time window and of the availability of production resources. So far it would seem like a decision-making process that does not involve a particular effort if it weren't for the fact that demand and production capacity have a high unpredictability therefore the programming approach becomes of the "Fuzzy Goal - Fuzzy Constraints - Fuzzy Decisions" type.

The design of the algorithm starts by examining the flow of information and data revolving around an MRP plan. In a very basic way, the most strategic inputs are those shown in the next image [62]:



62- MRP Programming Input

Wanting to use a linear programming approach by integrating fuzzy logic within it, start by defining which the planning "objectives" are. It is well known that linear programming is used to maximise or minimise a function (known as the "objective function") under conditions where a set of constraints is present. In the case of the algorithm under consideration, the objective is to minimise the costs incurred by the company, including those of purchasing and production (for the latter, also evaluating the costs due to suboptimal use of production resources, and those spent on subcontracting externally some of the work should internal availability be insufficient), those of maintaining inventories in the warehouse, penalty costs due to any delay in fulfilling a customer's order. As already anticipated, the use of fuzzy logic allows the planner to avoid that the value of the target (in this case the cost, but also the availability of production resources and the value of market demand are considered) is known precisely a priori, and this creates the condition whereby it is possible to find the value of the optimal decision variables according to the degree of relaxation of the targets (the less relaxation the more "optimal" the solution is in absolute terms).

The idea is that the algorithm is utilised weekly to cover a time window of 5 to 6 months (depending on the organisation's choices), a period that the MRP routinely covers. Assuming that market forecasts are updated weekly, it is necessary to reiterate the approach over the entire planning period and over all items, so as to account for changes in forecasts (that are also reproduced with some periodicity) and the arrival of new confirmed orders. It would be unthinkable to be able to schedule requirements over such a long period with only one schedule, since it is well known that the goodness of forecasts is greater the closer the period over which they are made.

Regarding the structure of the algorithm and the assumptions made, we model the one developed in (114) and a specific section will be devoted to highlight its peculiarities. The upgrade that can lead to the improvement of this, is the possibility for the user/MRP manager, to be guided in the choice of parameters that allow to widen/shrink the range of variability of the objectives, thus going to affect the value of the decision variables and/or to be able to manage the type of "membership function" that one wants to use (see section dedicated to fuzzy logic). Normally, these factors are chosen based on the business organisation, but one could find a way, through the development of an inferential model using fuzzy logic, to recommend their value to the user based on historical data or the value of current crispy inputs.

If one could provide the algorithm with a way to change what are its coefficients based on a certain cumulative experience, then one would have an "intelligent" algorithm: these coefficients could be seen as "weights" whose value would have a certain effect on the programming. At this point, fuzzy logic and fuzzy expert systems make it possible to create a set of rules (as already pointed out) that allow different decisions to be made as a consequence of the occurrence of specific conditions; the same rule-set approach, at this point, could be used to manage the algorithm's coefficients, in a manner similar to how it is done for a project activity scheduling problem in (120), where an Inferential System is created to guide CCPM (critical chain project management) activities.

13.1 **Problem Model**

In (114) a linear programming model for managing MRP is proposed and taken up in the present project.

The model makes it possible to determine the value of the decision variables of interest under the condition that they minimise the management total cost incurred by the firm. By setting a specific target cost, however, it is equivalent to doing deterministic programming: but since its value is not always known a priori then the use of fuzzy logic can be introduced within the algorithm.

Going to consider cost as a fuzzy variable, along with also the value of demand for i-th product in the t-th week and the availability of r-th productive resources in the t-th week one could propose a new formulation of the previous model, but before exposing it one must give explanations about the choice of considering these as fuzzy variables.

Starting with the target cost, it is unlikely that the resource planner, or even more so the firm, will know precisely its magnitude: it is possible to have ideas about target values thanks to the application of target-costing methods (121), but the number of factors affecting their actual feedback is very large therefore, it is difficult for them to then come to fruition in practice. Despite a wide variety of approaches to cost evaluation (122), some of which also take into account uncertainty factors (123), considering them as a fuzzy variable is an alternative to create a kind of problem relaxation. Obviously, it is necessary to have an idea of the cost range that should not be exceeded, and then to accept the value of the decision variables that allow it to fall within that range. A similar argument applies to the availability of production resources (that may vary over a certain period due to scheduled maintenance activities, breakdowns, or a sudden reduction in the workforce) and market demand since they are equally unpredictable.

Other factors that could be modeled with a fuzzy approach are product lead times (equally difficult to keep constant), as well as production costs, but in this discussion, we will assume constant for simplicity.

On how to choose the most meaningful mode of fuzzy representation, some scholarly articles, such as (124) and (125) give many suggestions. To switch from the deterministic MRP model to the fuzzy approach, one can initially consider a membership function of a linear type such as the one showed in eq. [XV].

$$\mu_i(x) = \begin{cases} 1 & \text{if} \quad B_i x \leqslant d_i, \\ \dfrac{kd_i - B_i x}{kd_i - d_i}, & \text{if} \quad d_i < B_i x \leqslant kd_i, \; i = 1, \ldots, m+1, \\ 0 & \text{if} \quad B_i x > kd_i, \end{cases}$$

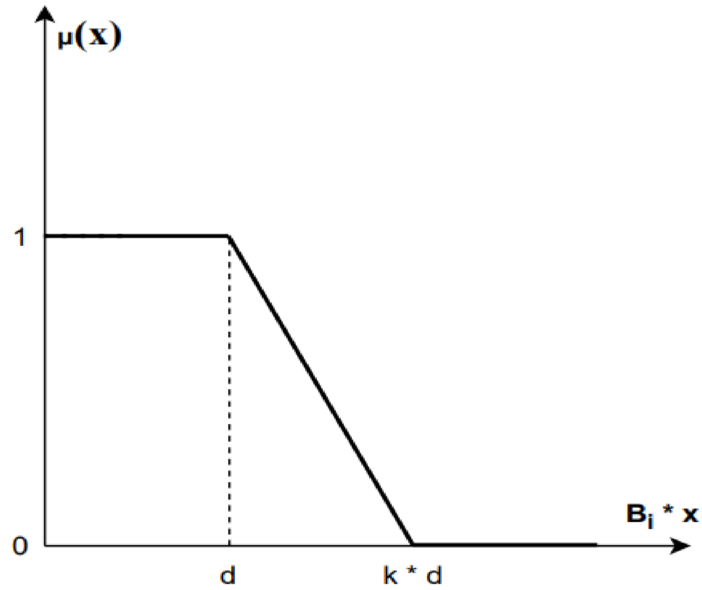
Equation XIV: Linear MF (constraints elasticisation)

Continuing to consider cost minimisation as an objective, one can use the following formalism and state:

- di = the minimum value that the cost can take (the reasoning is analogous in the case of demand modeling and resource utilisation); obviously to all the smaller values corresponds a maximum membership value (=1)

- k = is the multiplicative coefficient (>=1) that allows the elasticity of the problem to be handled and that the system must suggest to the user in planning. Increasing its value enlarges the range within the decision variables can change in value, making the solution still acceptable. It was explained earlier how the use of this coefficient ("k" in the case of cost; "l" and "j" for demand and resource availability, respectively) is the stratagem that the planner has to be able to interface with the uncertainty associated with defining a deterministic value for the target cost.

- Bix = is the part that represents the solution. So, if the solution falls within the defined range, then it is acceptable. The fact that the solution falls within the interval does not guarantee that it is the best in absolute terms, but it is optimal within the meaning of planning.



63- Open right function for constraints elasticisation

The change from a deterministic to a fuzzy model does not result in a change in the solving method, since it still remains a linear programming algorithm, but it does require that what was previously the objective function be transformed into a new programming constraint. By going to replace the equation [XV] in the model (even within the previous constraints limiting stock size and resource use) and introducing a new variable "λ" (on this one there is a constraint limiting its value to a maximum of 1) one goes to have as a new objective precisely the maximisation of the latter.

The use of this approach, seen in both (114) that in (126) is equivalent to what is called "min operator" in the theory of fuzzy logic and corresponds to:

$$\mu_{\tilde{D}} = \min_i \{\mu_i(x)\}$$

Equation XV - meaning of "min operator".

By then going to maximise the previous expression (that is equivalent to seeking the maximisation of the variable λ) we make sure to "maximise the satisfaction of the least satisfied constraint."

13.2 **Model Equations**

13.2.1 **Equations for the Deterministic model**

The formulation of the problem begins with the introduction of a linear programming model and the presentation of what is called an "objective function," in this case with a minimisation operator indicating precisely the purpose of the objective itself. The case of minimisation of the operating costs incurred by the enterprise is considered, and the model chooses to consider those that have greater strategic value and are easier to account for.

By reading the formulation from left to right, the decision variables of the problem can be identified (they will be the output provided by the algorithm as a programming result):

$$minimize(z) = \sum_{i=1}^{I} \sum_{t=1}^{T} (c_{prod_i}) * Q_{prod_{i,t}} + c_{stock_i} * Q_{stock_{i,t}} + c_{delay_i} * Q_{delayed_{i,t}}) + \\ + \sum_{r=1}^{R} \sum_{t=1}^{T} (c_{u-time_{r,t}} * Q_{u-time} + c_{e-time_{r,t}} * Q_{e-time})$$

Equation XVI - MRP Deterministic Objective

- Q_prod(i,t) = corresponds to the quantity of i-th product to be produced/ordered in the t-th week
- Q_stock(i,t) = corresponds to the quantity in stock of i-th product in week t-th
- Q_delayed(i,t) = corresponds (if any) to the portion of demand for the i-th product in week t-th that has not been met
- Qu_time = corresponds to the internal production capacity of the r-th resource in the t-th week not employed
- Qe_time = analogous to the previous one, but in this case the use of "extra-time" resources is considered, i.e., the amount of externally subcontracted labor hours

Each quantity is multiplied by a cost value, that is considered known for each unit of output:

- c_prod_i = corresponds to the production cost of the i-th product
- c_stock_i = corresponds to the cost of keeping the i-th product in stock (usually equal to a percentage of the selling cost)

- c_delay = penalty cost incurred by the company on the individual unit of product not delivered on time
- c_u-time = unit hourly cost of non-using the r-th internal production resource
- c_e-time = unit hourly cost for the use of the r-th external production resource

The horary costs referred to the using of internal/external production resources are considered constant values in order to add a simplification.

Having determined the objective function, we move on to define what are the constraints of the problem that condition the outcome of the schedule. There will be:

1) an applied constraint on reducing the quantities of i-th product in circulation (taking into account those scheduled to arrive, those to be produced for the current week, and those currently in stock) by ensuring that these are sufficient to meet the market demand of the current period (demand_i,t).

2) an applied constraint on the use of production resources that seeks to reward the use of internal resources to ensure the production of those quantities of product needed to satisfy demand. Since subcontracted or overtime production would have an increased unit cost.

3) a constraint on the unmet demand for i-th product at the end of the scheduling time window, making it zero.

4) a constraint on the value of the decision variables, so that the algorithm does not assign them a negative value. In the formulation of the algorithm on Python it will be ensured, moreover, that the variables Qprod, Qstock and Qdelayed are constrained to take an integer value.

$$Q_{stock_{i,t-1}} - Q_{stock_{i,t}} + Q_{prod_{i,t-LT_i}} + RP_{i,t} - Q_{delayed_{i,t-1}} + Q_{delayed_{i,t}} - \\ - \sum_{j=1}^{J} \alpha_{i,j} * (P_{j,t} + RP_{j,t}) = demand_{i,t}$$

Equation XVII : Det. Constraint 1

The term “$\alpha_{i,j}$” refers to the case in which the i-th product in programming is a component of a finished product, therefore the quantity relating to the latter can be deduced from a bill of materials (BOM); therefore it corresponds to the quantity of the i-th component necessary for the manufacture of a j-th finished product unit.

$$\sum_{i=1}^{I}(Labourtime_{i,r} * Q_{prod_{i,t}} + Q_{u-time} + Q_{e-time}) = AvailableCap_{r,t}$$

Equation XVIII: Det. Constraint 2

In the case of the general formulation, the existence of several "r" production resources was assumed, but in the application case, the further simplification of considering only one was introduced, making use of the assumption that the various finished product commissioned to the enterprise are realizable by assembly of components/sub-assemblies on the assembly line. Thus, it could be assumed that the case under consideration refers to an enterprise that manufactures sub-assemblies characterised by a not excessively large BOM and items that are not excessively "massive"; this would allow for the work to be carried out in a department that makes use of an assembly line.

$$Q_{delayed_{i,T}} = 0$$

Equation XIX: Det. Constraint 3

$$Q_{prod_{i,t}}, Q_{stock_{i,t}}, Q_{delayed_{i,t}}, Q_{u-time_{r,t}}, Q_{e-time_{r,t}} \geq 0$$

Equation XX: Det. Constraint 4

It is necessary, in some way, to "conserve" the value of the decision variables found week by week, as some of them become inputs for the following week's scheduling. The algorithm is runned weekly so that some variables can be updated, such as product demand for example, because they could receive an increasing or decreasing (due to confirmed orders, cancelled orders, change in forecasts) but also the availability of production resources (which may change in light of the fact that there may be problems undermining its continuity, such as anomalies or lack of personnel).

13.2.2 **Equations for the Fuzzy model**

Once the linear programming model has been created, we will proceed with its "fuzzification" by modifying its structure, according to the equation [XV] and the image [63].

What was considered the objective function in the deterministic model becomes, within the fuzzy one, one of the programming constraints. The new objective is determined by the introduction of the new variable λ, as already explained above, and consists in the search for its maximisation.

$$\lambda \leq k * z - [\sum_{i=1}^{I}\sum_{t=1}^{T}(c_{prod_i} * Q_{prod_{i,t}} + c_{stock_i} * Q_{stock_{i,t}} + c_{delay_i} * Q_{delay_{i,t}}) + \\ + \sum_{r=1}^{R}\sum_{t=1}^{T}(c_{u-time_{r,t}} * Q_{u-time} + c_{e-time} * Q_{e-time})]/(k * z - z)$$

Equation XXI: Fuzzy Constraint 1

$$\lambda \leq l * d_{i,t} - [(Q_{stock_{i,t-1}} - Q_{stock_{i,t}} + Q_{prod_{i,t-LT_i}} + RP_{i,t}) - Q_{delayed_{i,t-1}} + \\ + Q_{delayed_{i,t}} + \sum_{j=1}^{J}\alpha_{i,j} * (P_{j,t} + RP_{j,t})]/(l * d_{i,t} - d_{i,t})$$

Equation XXII: Fuzzy Constraint 2

$$\lambda \leq (l * d_{i,t} - 2 * d_{i,t} - [-Q_{stock_{i,t-1}} + Q_{stock_{i,t}} - Q_{prod_{i,t-LT_i}} - RP_{i,t} + \\ + Q_{delayed_{i,t-1}} - Q_{delayed_{i,t}} + \sum_{j=1}^{J}\alpha_{i,j} * (P_{j,t} + RP_{j,t}))/(l * d_{i,t} - d_{i,t})$$

Equation XXIII: Fuzzy Constraint 3

In the case of the two previous constraints, a dedicated coefficient called "l" has been introduced and manages the range of variability (therefore the elasticity) of the demand, considered as a fuzzy variable together with the target and the availability of productive resources. In the next two constraints, however, the coefficient "j" will be introduced: it’s “task” is to perform a function analogous to that of the previous two but intervening according to the availability of productive resources. The need to use three different coefficients and not a unique one is determined by the fact that these three fuzzy variables are dependent on factors of a different nature.

$$\lambda \leq j * AvailableCap_{r,t} - [\sum_{i=1} Labourtime_{i,r} * Q_{prod_{i,t}} +$$
$$+Q_{u-time} - Q_{e-time}]/j * AvailableCap_{r,t} - AvailableCap_{r,t}$$

Equation XXIV : Fuzzy Constraint 4

$$\lambda \leq j * AvailableCap_{r,t} - 2 * AvailableCap_{r,t} - [-\sum_{i=1}^{I} Labourtime_{i,r} * Q_{prod_{i,t}}$$
$$-Q_{u-time} + Q_{e-time}]/j * AvailableCap_{r,t} - AvailableCap_{r,t}$$

Equation XXV: Fuzzy Constraint 5

N.b. Those constraints that have not changed, in the transition from deterministic to fuzzy model, remain the same.

13.2.3 **How to manage coefficients calculation**

First of all, it will be necessary to define the "rules" that allow to manage the value of the programming coefficients decided by the planner. Hypotheses will be made in order to find, through inference, their value and then use it in programming. Secondly, the choice of the parameters developed by the rules must take place; they will have an effect on coefficients magnitude accordingly (the parameters are normally dependent on the internal choices of the company, which becomes able to strongly customise its programming according to the method most functional to the type of production).

An example could be the realisation of a series of rules for the factor "k" evaluation, which manages the elasticity of the acceptability range for target “z” variation: it can be assumed that the planner chooses the value according with the planning period (it inevitably triggers a certain average demand of the various product categories), the maximum demand (which can be evaluated from the historical trend) and with the penalty cost requested by the customer who the company is required to compensate for the single product unit not delivered on time. A possible approach for the planner, in the event that he does not have precise data available and therefore without uncertainty, could be hypothesised as follows:

A) Starting from an assessment of the current demand and the entity of the penalty cost that the customer imposes on the late delivery of the i-th type of finished product (cost that can vary periodically) and compare them with the limit values, which could be respectively the maximum historical demand recorded for that type of product (or the maximum demand that could occur from forecasts) and the maximum cost that the customer imposes as a penalty (this type of assessment would be much simpler in the event that the products in question were subject to seasonality, therefore characterised by demand and cost that has a rather predictable and repetitive pattern trend)

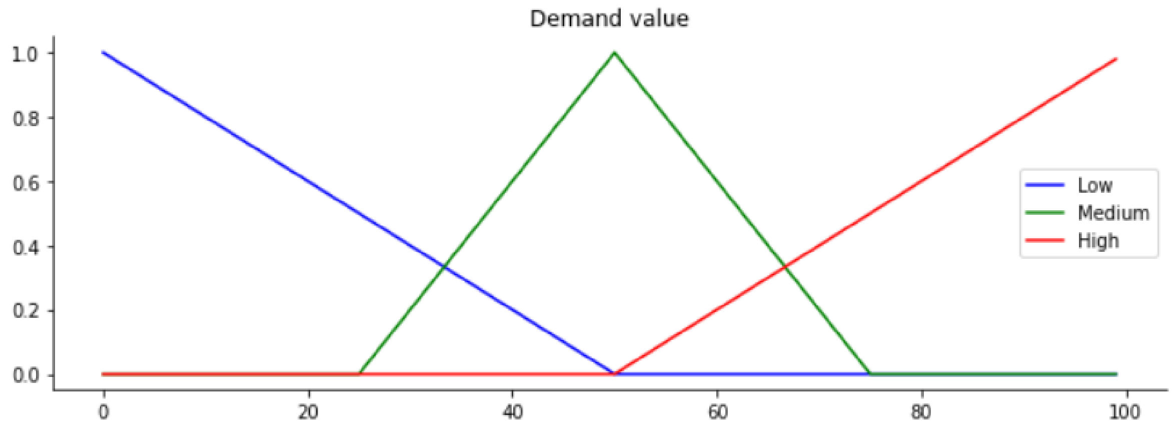
B) The limit values can be used to build the fuzzy sets that will be used to map the punctual values of demand and penalty cost at the beginning of each single week (since 1 week coincides with the launch interval of the algorithm and rescheduling for subsequent periods). In the case under examination, it was assumed to use 3 fuzzy sets (each characterised by a triangular membership function) for the representation of 3 possible linguistic variables representing the level of demand and cost. Quite simply, it was decided to use the following formalism:

- lvl 1: "Low" Demand / "Low" Penalty Cost

- lvl 2: "Average" Demand / "Average" Penalty Cost

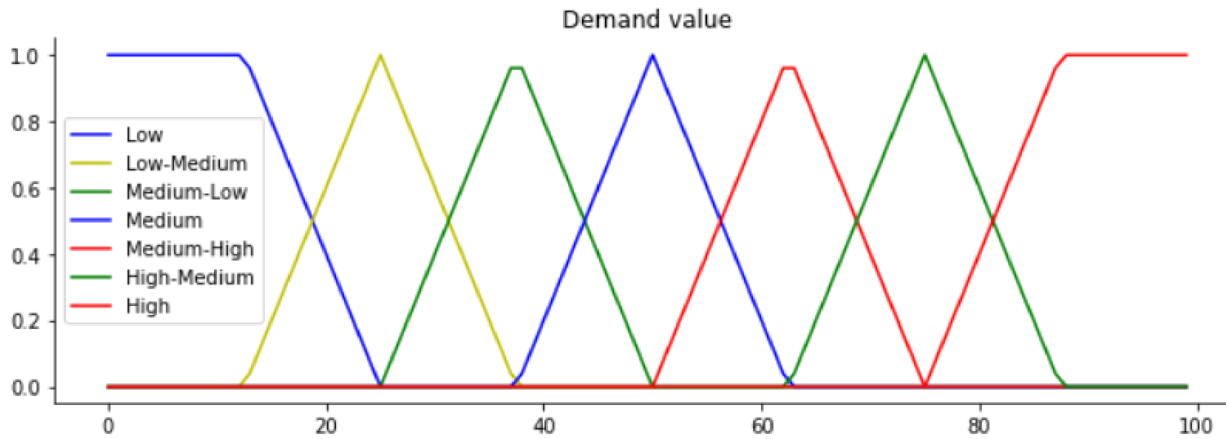
- lvl 3: "High" Demand / "High" Penalty Cost

In the case of the demand (assuming a maximum value set at 100 units for example purposes) the representation of the membership functions would become:



64- Demand Value 3 MF

The decision to work with 3 fuzzy subsets is linked only to functional reasons, in order to simplify the management of the algorithm and the evaluation of the coefficients; however, nothing would forbid multiplying the number of sub-sets to refine the quality and detail of the inference (image [65])

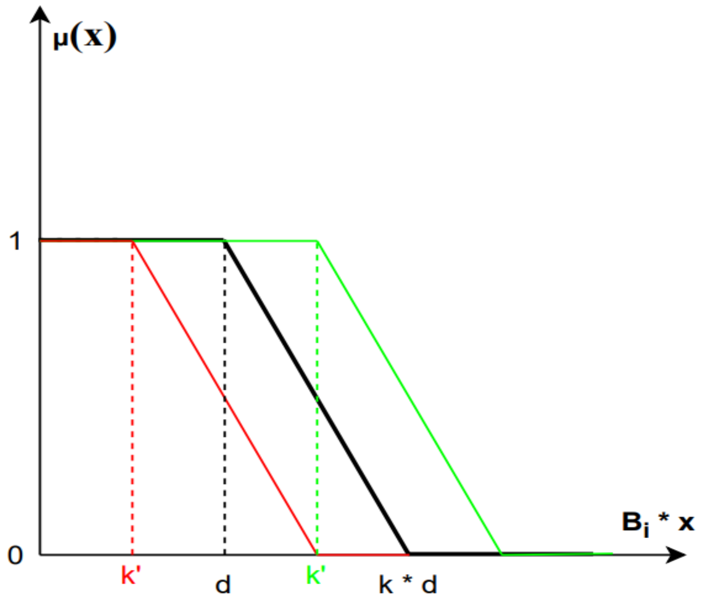


65 - Demand Value 7 MF

At this point, having generated the coefficient "k", the function [XV] could be applied and the effect shown in the image [61] could be replicated. However, this would reduce the dynamism of the problem because, in that specific case, a different coefficient would be generated for each programming week but always using the same "d" value (which could be equivalent to the target value). The idea that differentiates the current model from others available in the literature is that of not simply limiting itself to managing the size of the section "d"- "k*d" by acting on the value of the multiplicative coefficient but also changing from time to time that of “d”, creating a relationship between the schedule of the current week and that of the previous week. One could thus use as the objective target value of the week currently being planned (i.e. “z”) the one supplied in output from the previous week and use a coefficient k' which allows planning the current week with a new multiple target of the last one, with respect to which may be greater or less. To do this it was decided to design, with the same approach adopted for “k”, “j” and “l” already discussed, a new pair of coefficients which have been named "k_shrink" and "k_expand" precisely by virtue of the fact that their purpose is to restrict or increase the value of the target cost of the current week with respect to the one obtained in output in the previous one. The way in which one chooses to use one or the other is based on the comparison of the value of two very strategic parameters:

- the average weekly input demand compared with that of the previous week (taking into account the existence of possible fluctuations)
- the value of the total input backlog in the current week with that occurred in the previous one

More precisely: if the conditions were to occur in which the current average demand was lower than that of the already scheduled week or the current backlog was lower than the previous one, then one could opt for a reduction of the objective target (therefore an activation of the k_shrink coefficient) or, otherwise, for an increase in its entity if neither of the two conditions occurs. The possibility of intervening on the value of the target is far from trivial as it represents a part of the liquidity that the company "blocks" to allocate it to the management of needs.



66 -Open right function for constraints elasticisation: variable input target

The importance of using these coefficients of elasticity will become evident from the moment that the algorithm starts to simulate the presence of a certain planning "nervousness”. As already explained in (114), if one wants to create a solution capable of reproducing realistic scenarios, this factor must also be considered, and it could have various connotations. Firstly, the term refers to the condition by which a management plan such as MRP can be affected, to a greater or lesser extent, by variations in its results as demand varies. The demand value, as already said, could have a fuzzy nature and the ability to make the plan susceptible thanks to its variations in volume (i.e. the quantities referring to a certain order can change positively or negatively over time) also through the number of times that the same order "x" must undergo scheduling to be correctly programmed. The used forecasting system also plays a fundamental role in terms of the extent of the demand, as it will be less accurate the more the planning period lengthens over time.

14. The Algorithm

14.1 **Problem's assumptions**

Since it was not possible to access real data from an industrial setting, simulated data based on a set of assumptions were used to test the algorithm's functionality. This has been made possible since the Material Requirement Planning problem itself is rather generic, and its structure can be easily managed and customised according to the requirements of organisations, for which it is common practice to require customisation of their ERP systems (where algorithms and executives for scheduling, production planning and logistics control problems are stored). Thus, the problem is based on the following personalised assumptions:

1 - the first hypothesis concerns the characteristics of the production process: the algorithm is used by a fictitious company to plan the necessary needs of raw materials and components used for the production of four hypothetical finished products which don't require a transformation of the materials (e.g., machining or chip removal) but only assembly operations. For this reason, there is only one "productive resource" within the problem and coincides with the assembly line. Therefore, the production is consumed in a single type of assembly, emulating the characteristics of an ATO production philosophy.

2 - all the data needed for programming are on the same Excel file (simulating a kind of database) and divided into different "work_sheets" depending on their information content:

2.1 - data_sheet_cost: contains all the costs necessary for a complete requirement scheduling and an evaluation of the target cost (relative to the set of equations of the fuzzy model previously explained). These costs include:

- o prod/order_cost: depending on whether the item in question is a purchased component or a finished product realised by assembly, this cost indicates in one case the unit cost of purchase and in the other the assembly one (production)
- o stocking_cost: indicates the weekly handling cost due to keeping a unit of i-th product (whether it is a component, a finished product or a raw material unit) in stock

o delayed_demand_cost: this is the cost that the fictitious customer imposes on the manufacturing company as a penalty to be compensated if the weekly demand is not met (cost of a unit of backlog). This cost should be multiplied by the number of units not delivered on time, should a backlog situation arise.

o ut_prod_cost: hourly unit cost of not using production resources

o et_prod_cost: hourly unit cost of subcontracted/overtime production

2.2 - data_sheet_stock: reports the value of current stock levels at the time the algorithm was first run (initial stock levels).

2.3 - data_sheet_programmed_receptions: this contains data about scheduled orders for components that the company will receive in the various scheduling weeks.

2.4 - data_sheet_bom: is in effect a simplified representation of the "bill of materials" of the various finished products. It shows the code of the components required to make the part and the number of units required for each.

2.5 - data_sheet_other_info: contains data of various kinds, from lead times to the starting value of the demand in back log and the minimum order quantity for any components/materials.

2.6 - data_sheet_demand: is the work_sheet containing data about the demand values (expected/planned/confirmed) broken down according to the various finished products being assembled.

2.7 data_sheet_productive_capacity_disruptions: its contents refer to scheduled maintenance activities for the planning period. The assumption was to consider three different classes of disruptions according to their duration.

2.8 data_sheet_param: contains data about the range of variability of elasticity coefficients. Unless changes are to be made to the problem-solving method (changing the variability of the coefficients or using different membership functions) then the contents of this work_sheet should not be changed at all.

3 - the algorithm was used (for simplicity) for the purpose of planning the requirements of raw materials and components for the production of four finished products, coded as "item_fin_A" , "item_fin_B," "item_fin_C," and "item_fin_D" (the term "fin" in the acronym is representative of the fact that the item is a "finished" product, thus intended for the market), they correspond to the four RPNs of the problem (RPN = representative part number). The algorithm does not require any manipulation/adaptation in case more finished products are to be included.

4 - production orders are variable weekly while “programmed_receptions” are considered fixed during the week.

5 - to make scheduling more realistic, the possibility (controlled by a random variable) has been included whereby weekly production system failures may occur. So, the presence of unpredictable reductions in production capacity is simulated.

6 - the cost for the customer-imposed demand backlog is considered variable, both by product and by scheduling period.

7 - the demand for finished product is never zero throughout the duration of the scheduling horizon.

8 - every week a random variation in the demand for the various finished products is simulated and its maximum value can be set depending on the product category, the trend and reliability of the forecasting system or the seasonality. To simulate the tendency of forecasting systems to provide assessments of demand that are more susceptible to variations the farther away in time is the period in which this value of demand should occur, the condition has been added whereby demand can have random fluctuations in value the greater the distance the week in planning is within the planning horizon. Since at each iteration a different variation of demand can occur, this allows to evaluate the performance of the algorithm in conditions of demand nervousness (as a good replica of realistic programming conditions). The same reasoning is applied to the cost of the backlog.

9 - to facilitate the planning process, two separate problems were created:

- problem_1 : is the planning problem for the first week of the time horizon and uses data that comes from the source repository and whose output will be used as input to the second problem

- problem_th : th denotes "time_horizon" and refers to the requirements planning problem of the entire remaining time horizon (24 weeks in the assumed case)

This division into two problems was necessary because the data that are administered to the algorithm portion for programming the first week's requirements come from the deterministic model

10 - only one fictitious customer was assumed to be present

11 - the value that weekly can be taken by some variables is limited superiorly:

- should the production capacity required to meet all weekly demand be greater than that actually available, it will be possible to draw on "external" production capacity (overtime/subcontracting) of an amount equal (in hours) to the difference between the two

- in the opposite case the acceptable amount of non-productive hours will be limited

- the total size of the acceptable weekly backlog is fixed (and user-customizable) and must not be exceeded to avoid sharp drops in service level

- the number of hours that can be occupied per week by scheduled maintenance activities is considered equal to 4

12 - the level of service provided to the customer and the amount of backlog demand in at the end of the schedule can be used as a measure of scheduling performance

13 - the scheduling is simulated with a rolling approach: 6 weeks are programmed at each launch of the algorithm (this is because the lead times of the products in question are very short) and the results obtained are saved; at the first launch these will refer to week 1/6 but only those of week 1 are definitive and the others are provisional. In the next one weeks 2/7 will be scheduled and the results of week 2 will be kept and so on. This is to simulate the approach the material planner would use when scheduling.

14 - only the weekly programming target, the demand trend and the availability of production resources (in terms of hours) are considered having fuzzy behaviour

Once the assumptions of the problem have been presented, the architecture of the algorithm can be explained and, since it is rather sophisticated and large, we have chosen to document in this text only its strategic parts in functional terms and which can condition how the final output is processed. To lighten its structure and increase its comprehensibility, the choice was to opt for an organisation that was as modular as possible (that is, developing some functions on Python modules, created separately from the main code, which will be used only when necessary. To do this, the modules must be imported at the start of programming). Even before building the algorithm in its "fuzzy" version, it was necessary to start from the creation of the equivalent deterministic model to understand and refine the dynamics through the MRP programming should take place and then transfer them to the other model, for which a change on structural parts would require more effort given its more complex nature. The architecture of the final algorithm is then structured on a main script and 4 functional modules described below:

- 1 of these is dedicated to the calculation of the scheduling coefficients (the k, j, l coefficients already presented in the section from the theoretical content and the k1 coefficient that can have a shrinking or dilating effect on the starting target value) used to provide the elasticisation of the scheduling constraints.
- 1 module is used for results solved by the deterministic algorithm managing. This module is not provided at the end of the document since it is a sperimental version used only for the obtaining of the first week inputs (since the planning for others will be feed with the output of the previous one); as consequence, in the absence of this module, the user could use data from its own organisation
- 1 for the organisation of results; the module will insert the data allocating them in specific containers (mainly Python dictionaries) which allow to retrieve them easily and to pass them as planning input to the problem for the following weeks
- 1 for calculating the total costs incurred weekly

The part of the algorithm that is least intuitive to design and interpret is the part where the conversion of fuzzy programming constraints [equations from XXII to XXVI] into a "computer" language was realised. Most importantly, compared to the constraints previously discussed in the theoretical model, more "customised" ones were added in order to simulate more realistic application cases and to demonstrate how the user has some flexibility in handling them by intervening in some specific areas of the program.

These constraints include, for example:

- a constraint on the minimum purchase lot size (MOQ = minimum order quantity) of the components needed to make the finished product: assuming that the user has a database available in which specific commercial information are listed, such as the minimum purchase lot size that a supplier can take charge of for a certain product code, so they could be used in order to output a correctly dimensioned bill of necessary purchases that can be directly sent to the purchasing department. The addition of a constraint of this type causes, as a consequence, that the purchase of components will no longer be limited to the precise quantity (the minimum one) that production needs in order to meet weekly demand but will in some cases be higher depending on commercial issues, so the effect will be a size increasing of the stock and consequently the management costs linked to it.

Example: if during a planning week the need arises to purchase 25 units of a certain component, which corresponds to the precise number required for the production of the finished product, it would be ideal for an equal number to be purchased so that the maintenance management costs can be reduced (neither one more unit because it would weigh on stock levels nor one less because demand must be satisfied in its entirety); if, on the other hand, the supplier requests a minimum purchase lot of 50 pieces, he would be "forced" to purchase only that volume and store the 25 unused units in the warehouse (50 - 25 = 25), thus generating storage costs.

- a restriction on the ceiling of external work hours (subcontracted) that can be exploited weekly: if the value that this variable can assume is not restricted but the possibility is left for it to assume a value greater than or equal to 0 (and therefore without limiting it above) then the result could, depending on the demand volumes, suggest using an excessively high number of hours of subcontracted work compared to the company's possibilities.

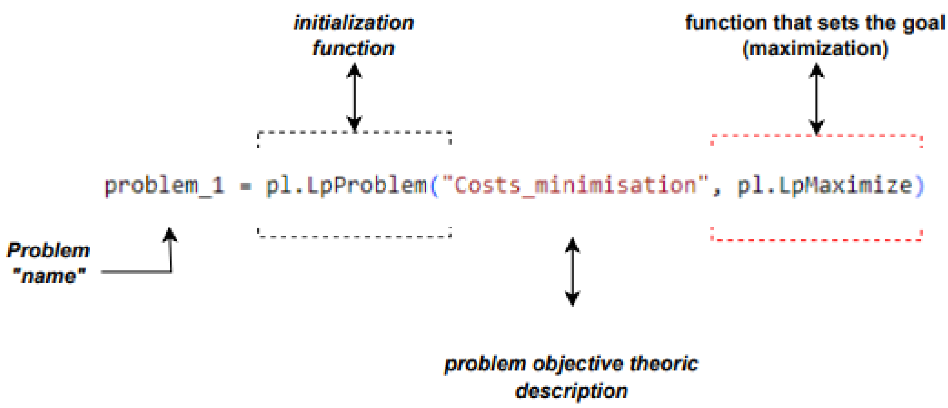
- a constraint on the maximum number of hours of underutilisation of production resources: this in particular limits the maximum acceptable number of hours not spent for production (a single assembly line in the case study hypothesised here). Limiting underutilisation triggers the consequence that, in order to amortise those management costs that the company would have pay due to machine downtime, programming ends up generating oversised production orders compared to what the weekly demand would require; therefore the difference between the quantity that is produced and that requested is equivalent to the volume of items that will be placed in the warehouse to then be able to be used as safety stock in the scenario of a possible excess demand in subsequent periods (obviously increasing the size of stocks will not be a cost zero action, but there are variable inventory costs depending on the type of product).

14.2 **Code building**

The linear programming algorithm was generated using the PuLP library (documentation available at (wb –29) and (wb – 30) , designed ad hoc for solving linear programming problems. Once the traditional installation procedure has been followed, this library offers very intuitive methods for problem design.

1- **Initialisation of the problem**

The first step requires creating the problem, whose initialisation takes place with specific formalisms:



67- pulp functionalities

The "pl.LpProblem()" method is used to initialise a new problem. This uses a minimisation function by default but, in the case under consideration, it was necessary to specify with the "pl.LpMaximize()" function that a maximisation problem was being created (maximisation of the variable λ entered later)

2 - **The variables**

The class "pulp.LpVariable()" is used for inserting decision variables:

```
class pulp.LpVariable(name, lowBound=None, upBound=None, cat='Continuous', e=None)
```

It's parametrs, as reported in the documentation, are:

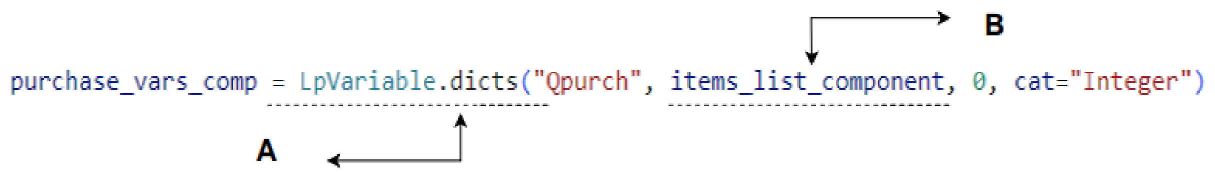
a) name – The name of the variable used in the output .lp file
b) lowbound – The lower bound on this variable's range. Default is negative infinity
c) upBound – The upper bound on this variable's range. Default is positive infinity
d) cat – The category this variable is in, Integer, Binary or Continuous(default)
e) e – Used for column based modelling: relates to the variable's existence in the objective function and constraints

In the case of the problem under consideration, the following decision variables were identified:

a. purchase_vars_comp: quantity of components needed to realise the weekly demand.
b. stock_vars_comp: size of component stocks in the week under planning
c. production_vars_fin: quantity of finished goods to be produced
d. stock_vars_fin: size of finished product stocks in the week under planning
e. delay_vars_fin: backlog demand size (in units of product)
f. ot_resource_fin: number of hours of underutilisation
g. et_resource_fin: number of hours under subcontracting/overtime

In the next few code lines is shown how some of those variables are initialised, according to PuLP functions:

```
# problem variables
production_vars_fin=LpVariable.dicts("Qprod",items_list_finished,0,cat="Integer")
lamb = LpVariable("lambda", 0, 1)
```



A slightly different method than that involving the use of the "LpVariable" class alone was used in defining the variables. The class "LpVariable.dicts()" (A) is used to create a dictionary of variables according to the acronyms contained in (B), which represents the list of component recognition codes. This was a necessary step since we are not working with a unique "purchase_vars_comp" variable, but there is one for each of the components being programmed. The value 0 refers to "low_bound" (i.e., the minimum value that the variable can take) and was set this way since programming should not generate negative values for decision variables. The "upper_bound" argument has been specified only for the variable "lamb" (= λ) since this can take (see theoretical explanation) only values between 0 and 1; the other variables, however, do not have a constraint limiting the upper extreme. For the variables "ot_resource" and "et_resource" there is the chance they can both non-integer or integer values; the same condition is not permissible for the others, given their nature (e.g., a result such that 10.2 finished goods are required is not permissible, let alone realistic).

14.3 **Elasticisation Coefficients**

As already stated in the theoretical explanation section of the algorithm, it was necessary to introduce in the script an area dedicated to the 5 programming coefficients. In the main code there are no lines showing how these are obtained, but only a callback of the respective calculation function that has been designed on an independent module ("coefficient_module"); To lighten the discussion then only the method used for obtaining one of the coefficients (the coefficient "k" in this case) will be shown and explained since a completely similar procedure was used for the others.
For the realisation of the functions that will be shown below we made use of one of the few resources available in the Python language for the realisation of systems using fuzzy logic, this is the library "Scikit-Fuzzy".

```
def get_coefficient(param_list, range):
    x_param_1 = np.arange(0, param_list[0] + 1, 1)
    x_param_2 = np.arange(0, param_list[1] + 1, 1)
    x_coeff  = np.arange(range[0], range[4], 0.01)
```

The first step to be performed is the choosing of the parameters that control the value of the scheduling coefficient. In the case under consideration, the scenario was assumed whereby the “k” coefficient (k_coeff), which is in charge of managing the range of variability of the weekly target cost, has a value conditioned by that of the current week's average demand (demand_value) and the current penalty cost that the customer imposes on the individual unit of finished product in the backlog (delay_cost_value). For both terms there is a strong contribution made by the existence of random variations that can add to or subtract from each other.

The following terms:

- "x_param_1"

- "x_param_2"

have different meanings depending on the coefficient that is to be computed, but in this case they correspond to the maximum values assumed by the variables "demand_value" and "delay_cost_value," respectively, and used as parameters to process the triangular membership functions constructed by the "fuzz.trimf()" function ("trimf" stands for "triangular membership function"). The three triangles constructed all have a base of the same size and a height equal to the maximum membership value, thus unitary. The scikit-fuzzy library allows the user to choose the type of membership function to use from a rather wide range that also includes the possibility of creating custom ones. For example, if there was a need to use a trapezoidal mf then the "fuzz.trapmf()" function would be used and would require 4 parameters to be constructed instead of the 3 needed in the current case.

Thus, for demand and cost, 3 levels are created:

- lo = low
- md = medium
- hi = high

This subdivision will be strategic in working out the value of the coefficient "k". Using a Mamdani-type FIS 3 membership functions will be associated with "k". The actual value (the crisp one) will then be obtained, following the aggregation of the results from the 3 functions, by choosing the appropriate "defuzzification" method. In the next code lines, in a similar way for what was done for demand and cost, the range of variability of the entity of "k" is defined.

```
param_1_lo = fuzz.trimf(x_param_1, [0, 0, (param_list[0]/2)])
param_1_md  =  fuzz.trimf(x_param_1,  [(param_list[0]/4),  (param_list[0]/2),
(3/4)*param_list[0]])
```

```
param_1_hi = fuzz.trimf(x_param_1, [(param_list[0]/2), param_list[0], param_list[0]])

coeff_lo = fuzz.trimf(x_coeff, [range[0], range[0], range[1]])
coeff_md = fuzz.trimf(x_coeff, [range[2], range[1], range[3]])
coeff_hi = fuzz.trimf(x_coeff, [range[1], range[4], range[4]])
```

The value of the terms x_param_1 and x_param_2 (only param_1 dependencies are showed) depend on those contained in "param_list" and "range" that are passed as input from the main script; as for "range," its magnitude is defined by the values found on the work_sheet "data_sheet_param."

The next two steps require that the value of the variables that trigger the various levels of "delay_cost" and "demand_level" be defined first (this means that the algorithm must be able to figure out when the demand is considered "low," when the penalty cost is considered "medium," and so on), and this becomes possible only if, as fuzzy logic and expert systems require, there is a repository of "knowledge" built by a domain expert. For example: "param_1_lo" is the weekly demand level perceived as "low" and defined as such by those communicating with the planner. The main script will pass to the function the value "max(list_max_demand)", which corresponds to the maximum average demand recorded so far for that product category (its value will update during the scheduling depending on the data that will occur); the same goes for "max(list_max_delayed_cost)". Consequently, the membership function corresponding to the various demand levels that maps the membership value of the current input will use that value (max(list_max_demand)) as "x_param_1" to auto-construct.

At this point, again through the main script, the current week's average demand for finished goods will be passed as the value of "demand_value," and its entity will be associated with a membership-value with respect to each of the 3 mapping functions (the same happens with "max(list_max_delayed_cost)" and "x_param_2").

```
[max(list_max_demand), max(list_max_delayed_cost), mean_demand_dict["week:", t+1], mean_delayed_dict["week:", t+1]]
```

The function "fuzz.interp.membership()", will calculate the membership value with respect to the previously designed membership functions.

```
param_1_level_lo=fuzz.interp_membership(x_param_1, param_1_lo, param_list[2])
param_1_level_md=fuzz.interp_membership(x_param_1, param_1_md, param_list[2])
param_1_level_hi=fuzz.interp_membership(x_param_1, param_1_hi, param_list[2])
```

Once the "levels" and membership-values have been defined, the inference rules can be constructed:

```
active_rule_1 = np.fmax(param_2_level_lo, param_1_level_lo)
coeff_activation_lo = np.fmin(active_rule_1, coeff_lo)
active_rule_2 = np.fmax(param_2_level_md, param_1_level_md)
coeff_activation_md = np.fmin(active_rule_2, coeff_md)
active_rule_3 = np.fmax(param_2_level_hi, param_1_level_hi)
coeff_activation_hi = np.fmin(active_rule_3, coeff_hi)
k0 = np.zeros_like(x_coeff)
```

The code lines shown above are those utilised for the construction of a set of rules that uses "max" and "min" operations thanks to the functions developed in the Numpy Library. As already reported in the section explaining the most common operations performed on fuzzy-sets, the execution of these operators corresponds to OR/AND type evaluations, respectively. What happens corresponds to the schematic representation in the image [30], but an example can make the sequence of actions performed by this module more understandable:

1 - the function receives the four inputs from programming

2 - two of the inputs are used to construct the membership functions, for the others the fuzzification and membership-value calculation with respect to each of the 3 membership functions (mf) constructed takes place

3 - the "active_rule_1" rule performs an evaluation of the membership-values "demand_level" and "delay_cost_level" respectively on the membership functions "param_1_level_lo" and "param_2_level_lo" (which would correspond to "demand_level_lo" and "delay_cost_level_lo" respectively. At this point, the “np.fmax()” operator performs the OR operation by going to select the larger value of mf between the two

4 - the operator "np.fmin(") can perform the function of comparing that value with "coeff_lo" (whose activation range is known - it would correspond to "k_lo") , i.e. it will allow to understand (on a level from 0 to 1) what is the membership value level of the combination of the current value of the inputs with respect to "coeff_lo"

5 - the same steps, with the same inputs, are repeated to evaluate the membership values with respect to "coeff_md" and "coeff_hi"

6 - the next step is the aggregation of the results from the 3 inferences; it is done again by applying the "max" operator

7 - the "fuzz.defuzz()" method is used to perform the defuzzification and obtain a crisp value in the output. This step requires that the defuzzification method to be used be passed as an argument. In this case the CoG (defined as 'centroid" by the documentation) was used:

```
# Aggregate all three output membership functions together
aggregated = np.fmax(coeff_activation_lo,
np.fmax(coeff_activation_md, coeff_activation_hi))
# Calculate defuzzified result by specifing the method to use
# (in this case the centroid (CoG = center of gravity)
coeff_value = fuzz.defuzz(x_coeff, aggregated, 'centroid')
```

As already anticipated, except for the parameters that are passed to the functions, the procedure for calculating the 5 scheduling coefficients is the same, week by week. In the main script, the calculation function simply needs to be called to pass the correct arguments depending on the coefficient to be calculated. Below are the dependencies of the other 4 parameters:

- **"j" coefficient**:

```
[max(list_max_demand), 4 , mean_demand_dict["week:", t+1] +
random_demand_excursion, disrupt_hours_week[t+1]]
```

In this case, the membership function will be constructed based on the maximum demand for finished product during the scheduling period (max(listmac_demand)) and the maximum number of hours (assumed equal to 4) that can be occupied by activities that require the interruption of production. The average demand values (also influenced by the presence of random variations) and the actual number of hours required for scheduled interruptions ("disrupt_hours_week[t+1]") will be supplied as inputs on a weekly basis. The inference that the algorithm uses to make the choice of "j" is based on the interruption activities scheduled during the week: if the company has a limit on the maximum number of hours that can be devoted to weekly routine activity on the production, the algorithm could generate a coefficient that is larger the higher the number of hours foreseen for the interruption and will add them to those of overtime as the difference between the weekly capacity (40 hours) and the actual available capacity.

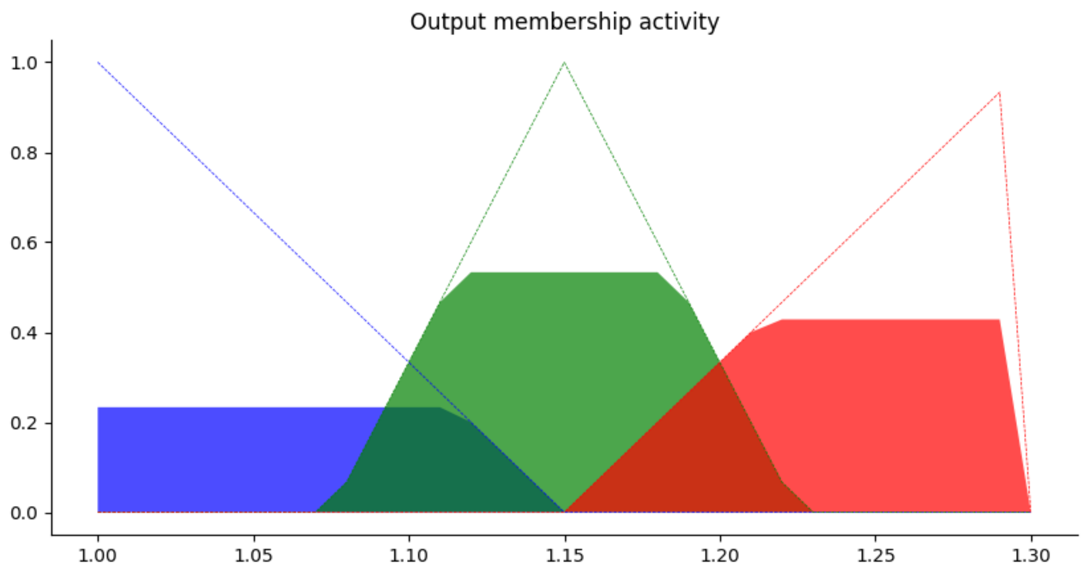
- **“I” coefficient:**

```
[max_storic_demand, max_actual_demand, storic_week_total_request["week", t+1]/product_types, (week_total_request["week", t+1]/product_types) + random_demand_excursion]
```

For the "i" coefficient, the functions will be constructed according to the maximum historical demand (obtained from an obsolete MRP plan) and the maximum demand in the current planning period ("max_historic_demand" and "max_actual_demand"). The arguments from dynamic behaviour are the average weekly demand and the maximum demand (this can vary as the presence of random fluctuations can add to its value).

- **“k1” coefficient**: depending on the case, this coefficient will have a dilatory or restrictive effect on the value of the starting target that will be used in the weekly programming, thus influencing the value of the decision variables. The inference takes place through a chain of verification of certain conditions that lead the algorithm to make a decision; in practice, the reasoning developed consists in "learning" from what was done during the previous week by verifying what was the average and backward demand in the week already programmed and what was the value of the coefficient that had been chosen which created the solvable problem; at that point, by comparing both the current average demand and the backlog input demand with those of the previous week, the algorithm will choose whether to reduce or increase the target by multiplying it by “k1” itself (“k1” will either have a value less than one or a value greater than one, depending on the cases).

The coefficient calculation functions moreover realise a graphical representation of the aggregation of results that looks as in the following images:

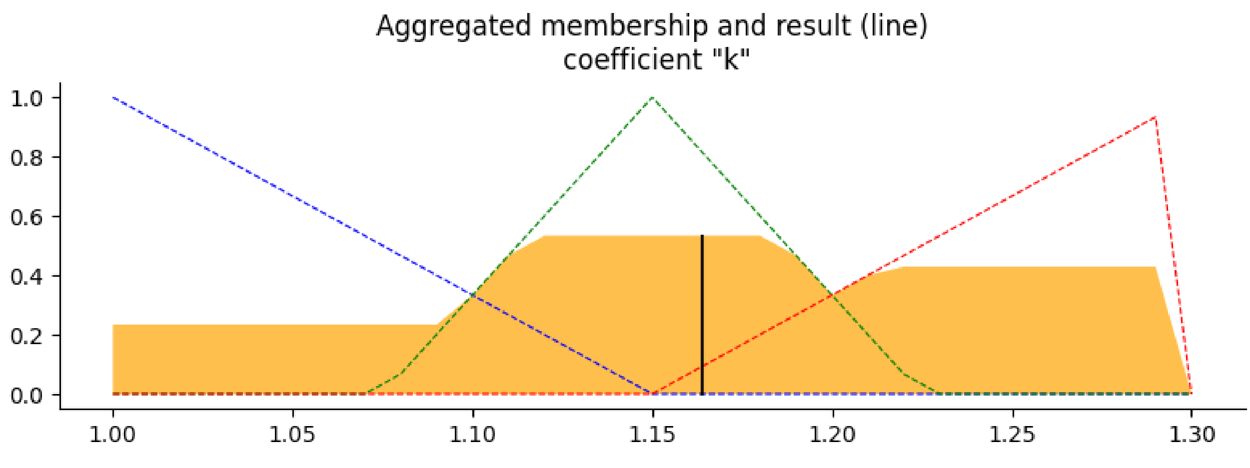


68 - output membership activity

In [68] is depicted the "Output membership activity," that is, the level of activation of the 3 membership functions with respect to "k_lo", "k_md", and "k_hi".

In the image [69], on the other hand, the result of defuzzification is shown, thus the value of the coefficient k calculated by "CoG" method.

It should be noted that the portion of code that determines the generation of the graphs is deactivated by default but can be put into operation by acting on the module for



69 - "k" coefficient calculation

calculating the coefficients. The choice to operate in this way finds justification in the fact that otherwise, generating many images at each iteration, the device would run out of memory forcing it to interrupt the resolution.

14.4 **Planning Constraints:**

at this point the coefficients obtained in the previous step can be used within the planning constraints to replicate fuzzy mathematical set equations [XXII-XVI]. In the next rows some constraints are translated:

```
problem_th += lamb_th <= (list_coeff[0]*z - lpSum([data_dict_cost[i][0] * production_vars_fin[i] + data_dict_cost[i][1] * stock_vars_fin[i]+ data_dict_cost[i][2] * delay_vars_fin[i]+ (data_dict_cost[i][3]) * ot_resource_fin[i] + (data_dict_cost[i][4]) * et_resource_fin[i] for i in items_list_finished]) )) / (list_coeff[0]*z  - z )
```

The lines of code shown above correspond to the conversion of the fuzzy equation [XXII] into the Python language. The content of the code makes possible to transform what was the objective function of the deterministic problem into a constraint of the fuzzy version. Whenever the formalism "problem +=" is carried over, a constraint is automatically added to the problem.

With the "lpSum()" function, on the other hand, an iteration is performed on all the variables: this means that technically a constraint has been added which is iterated on a list ("for i in items_list_finished") to be able to apply it on each finished product. Similarly, the constraint [XXIII] is transformed as follows:

```
for finished_prod in items_list_finished:
problem_th += lamb_th <= (list_coeff[2]*(data_dict_demand[finished_prod][t+1] + random_demand_excursion) - (finished_stock_dict[finished_prod] - stock_vars_fin[finished_prod] + production_vars_fin[finished_prod] - actual_delayed_demand[finished_prod] + delay_vars_fin[finished_prod])) / ((list_coeff[2]-1)*(data_dict_demand[finished_prod][t+1]+random_demand_excursion))
```

14.5 **The Solver**

```
print("Status Problem TH:", (t+1), LpStatus[problem_th.status])
status = LpStatus[problem_th.status]
for v_1 in problem_th.variables():
    print(v_1.name, "=", v_1.varValue)
    solution_problem_th.append(v_1.varValue)
```

```
if lamb_th.value() != 1 and lamb_th.value() > 0:
    lamb_count += 1
    non_opt_week.append(t+1)
if status == "Infeasible" or  status == "Undefined":
    infisibilities += 1
    non_opt_week.append(t+1)
```

Once all constraints have been defined, one can proceed with solving the problem using the "problem.solve()" function, that consists, by default, of applying the CBC solver (Coin-or branch and cut), an open-source mixed integer linear programming solver written in C++. It can be used as a callable library or using a stand-alone executable. The official Pulp documentation offers a deep explanation about all types of solvers which could be used. By using the formalism “LpStatus[problem.status]” it could be accessed the problem status after the resolution occurred; normally it could assume five different status:

- Optimal: if an optimal solution has been found

- Unbounded : if there aren’t sufficient constraints

- Infeasible: if the problem hasn’t a solution due to its nature or applied constraints’ nature

- Undefined: the optimal solution may exist but may not have been found

- Not solved: it’s the problem status before solver application

At the end of the resolution phase, if a value less than 1 for the variable λ (but in any case, greater than 0) has occurred, the algorithm will keep the number of the programmed week to allow, at the end of the planning, a verification of the factor (random demand, cost of delay or target value) that contributed to the failure of reaching the maximum value.

14.6 **The Output**

The algorithm is programmed both to provide an output on the screen that can be consulted quickly during the "running" of the same and to save the results in files dedicated to each programming week.

Normally, using a traditional and open-source Python editor such as Visual Studio Code (wb – 31), it will be possible to access the programming results from the terminal, which will be printed as follows:

```
Result - Optimal solution found

Objective value:                1.00000000
Enumerated nodes:               0
Total iterations:               0
Time (CPU seconds):             0.03
Time (Wallclock seconds):       0.02

Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.04   (Wallclock seconds):       0.04
```

70 - algorithm output

Firstly, a message is shown describing the status of the problem (in this case optimal) and the value of the objective function found (in this case equal to 1 and corresponding to the maximum). Below is information about how much performance the CPU had while the problem was running. The rest of the output is shown as follows (data taken from Week 2's schedule):

START PROBLEM WEEK n 2:

CAPACITY REPORT:

ACCIDENTAL CAPACITY REDUCTION OCCURRED: - 4 HOURS OF AVAILABILITY

IN WEEK 2 PRODUCTICE CAPACITY AVAILABLE IS: 32.5

THE TOTAL PRODUCTION TIME NEEDED IS: 39.04

First of all, any changes in production capacity following planned or unforeseen activities are communicated to the user by providing a residual of the usable production hours (capacity available). Secondly, it shows an estimate of the production hours needed to satisfy the demand for finished products in the current week without generating a backlog. After the information about the capacity, the fluctuations in the backlog cost per unit of finished product that have occurred compared to the starting values are shown, for example:

ACTUAL COST FOR item_fin_A BACKLOG OF DEMAND IS:48

ACTUAL COST FOR item_fin_A BACKLOG OF DEMAND VARIED TO: 49

ACTUAL COST FOR item_fin_B BACKLOG OF DEMAND IS: 52

ACTUAL COST FOR item_fin_B BACKLOG OF DEMAND VARIED TO: 50

Before the results for the decision variables, the ones of the programming coefficients and that of the weekly target, which will condition the extent of the decision variables, are shown as below:

- *THE ALGORITHM PROCESSED A K1 VALUE OF: 1.252*

- *PREVIOUS OBJECTIVE VALUE 'Z' WAS: 11837.09*

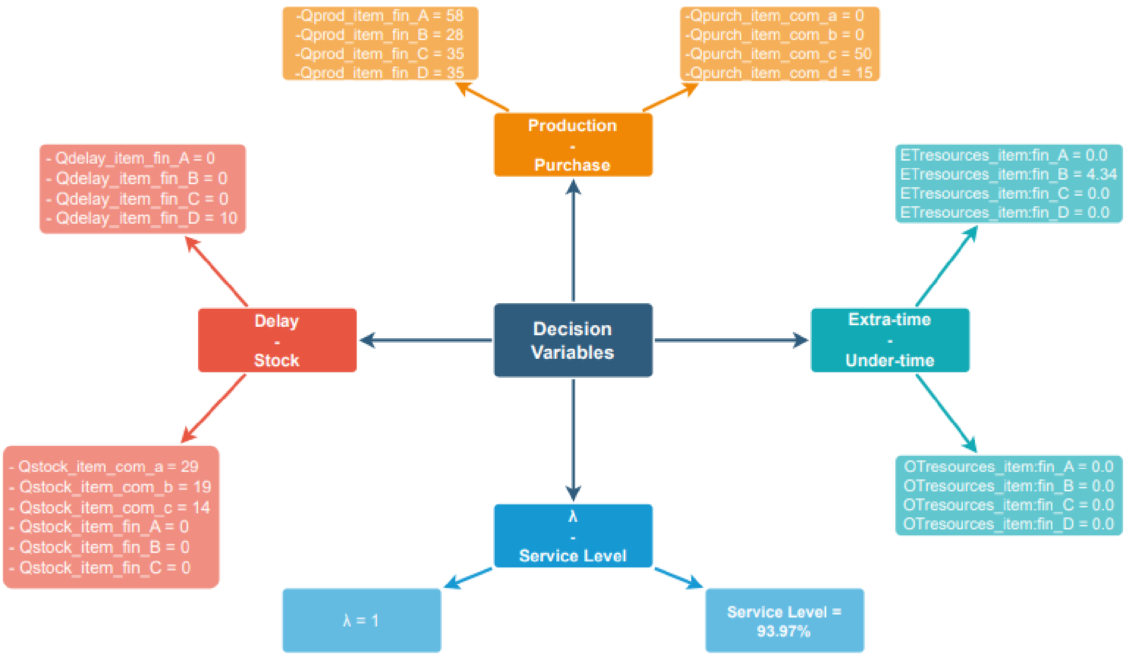
- *AFTER PROCESSING NEW OBJECTIVE VALUE 'Z' IS: 14816.27*

1) THE ALGORITHM PROCESSED A K VALUE OF: 1.245

2) THE ALGORITHM PROCESSED A J VALUE OF: 1.099

3) THE ALGORITHM PROCESSED A L VALUE OF: 1.122

Lastly, the values assumed by the decision variables will be shown together with that of the level of customer service. To facilitate representation, only some of them are shown in the following diagram, in a more schematic way than they appear on the editor terminal



71 - Decision variables programmed by the algorithm (week 2)

The same results will be saved in excel and divided into dedicated work_sheets.

14.7 **How does the algorithm actually work?**

Having given an overview of the libraries used in the algorithm, the various dependencies, variables and constraints of the programming problem, it is necessary to explain, albeit briefly, how this arrives from reading the input to returning an output. We can see this cyclic process as a sequence of operations:

- 1. The algorithm reads two files:
    a. the database that contains all the data that refer to the planning horizon to be planned
    b. a database with the company's history relating to the same product category (it could be a plan that the company programmed in the previous year, in the same "season").

This step also includes an error check that can protect the user. Problems that can occur are of various nature, such as the absence of all the dependencies necessary for the "run" of the algorithm (failed to install a dictionary or import a functional module) or an error in inserting the correct file from which to read the database (path or file format error)

-2. The second step is to organise the data contained within the database in iterable containers (python dictionaries) that can be used during programming. This leads to having a homogeneous subdivision of the data as regards the type of content (for example a dictionary that contains only the demand values of the various finished products and one that contains the breakdown of the bill of materials) and accessible in a targeted manner (using as the "key" to access the dictionary the name of the i-th finished product, is possible to have control over the values associated with it). This step is made possible by the "group_elements" function. This organisation is an emulation of how organisation occurs through an ERP software.

-3. The third step is to define the factors that control the value of the elasticisation coefficients:

a) Coefficient "k": for its calculation it will be necessary to enter the maximum values foreseen for the average demand of the various finished products and for their backlog cost. This could be done by assessing its extent from past programming.
b) Coefficient "j": two dictionaries are created, they have the same type of keys but different associated values. One contains key-value pairs corresponding respectively to the number of the week to be programmed (e.g. "week 1"/"week 2") and the average demand for the finished product (obtained as the sum of requests for the week divided by the number of different finished products), in the other the values will consist of the average cost of the backlog (obtained with the same methods). The work_sheet containing all the programmed interruption activities of the production process is accessed and the total number of hours of weekly downtime per intervention category is calculated. There are 3 categories: 4h intervention (cat.A), 1h intervention (cat.B), 0.5h intervention (cat.C).
c) Coefficient "l": for this coefficient, the result of the comparison between the historical and current values of demand, the historical maximum demand (already occurred) and the maximum foreseen for the planning horizon are calculated.

- 4. The linear programming problem can be initialised using the methods described above
   a) First, the problem is created by specifying the nature of the objective (minimisation or maximisation)
   b) Then all the decision variables to which the problem will attribute a certain value must be initialised
   c) All the constraints of the problem are inserted
   d) The solver is applied

- 5. Through the portion of code

```
while iterations_counter != effective_horizon-6:
    w = 0
    t = 1 + iterations_counter
    […]
    while t != (time_horizon-1) + iterations_counter:
```

the rhythms of planning are marked. The first while cycle allows to plan the entire time window in 6-week blocks. For each time the value of "t" equals that of "(time_horizon - 1) + iterations_counter" there will be, at the end of the code, a line that will increase the value of "iterations_counter" by one unit allowing the programming to always cover a period of the same size but moving forward in time:

```
t = t + 1
    […]
    iterations_counter +=1
```

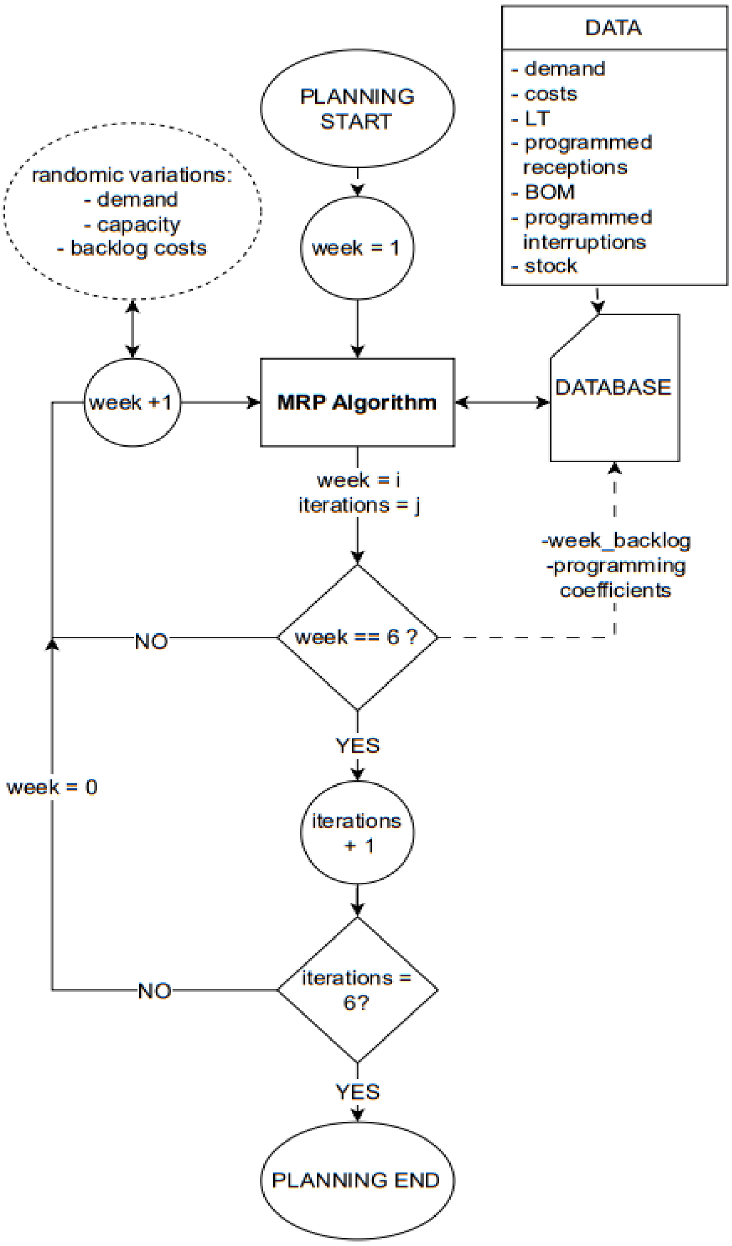
- 6. When the algorithm is launched, the week counter "w" is reset (it's maximum value is equal to 6). It will be used to keep track of the backlog demand of the current week to be able to pass it as input to the programming the following one. Since planning takes place in steps of 6 weeks at a time, when it starts with t = 1 (= week_1) weeks 1 – 6 will be programmed but if in 1 a backlog is generated then this must be saved to use it when it starts programming period 2-6 (t = 2 , since iterations_counter = 1 and w = 0) to pass it as input to week 2, and so on.

```
if w == 0:
      stored_delay = sum(actual_delayed_demand.values())
```

In the very first week (t=1) a random variation of demand will not be generated (given the proximity of the period there should be a rather high certainty about the value of the latter), but in any subsequent week there will be a random variation (positive or negative) with a peak value of 24% of the total average weekly demand:

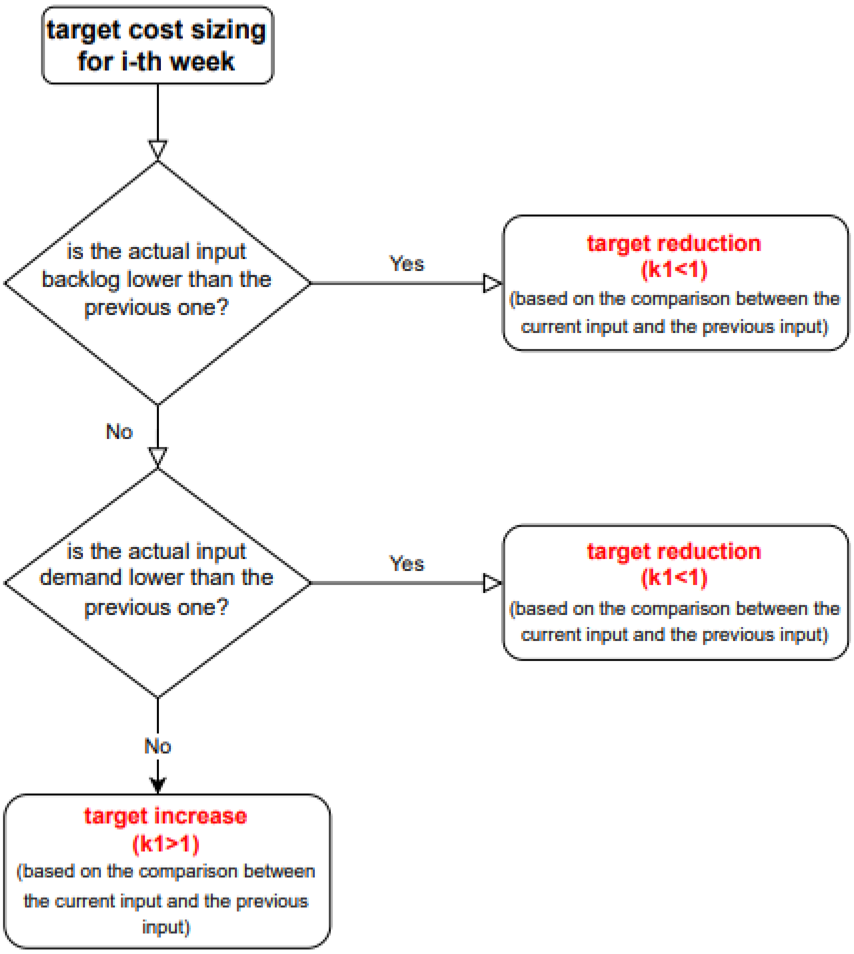
```
        if t == 1:
            random_demand_excursion = 0
            rnd_dem_tot = 0
        else:
            ex_d_perc = w * 4  # very week the excursion could grow by 4%
(ex_d_perc = excursion_percentage_demand : max = 6*4 = 24%)
            random_demand_excursion = rnd.randint(-
      round((mean_demand_dict["week:", t+1]/100)*ex_d_perc),
      round((mean_demand_dict["week:", t+1]/100)*ex_d_perc))
            rnd_dem_tot = random_demand_excursion * product_types
            prev_excursion.append(rnd_dem_tot)
```

With the same dynamics, a random variation of production capacity and backlog cost will be generated. Entire algorithm logic scheme is showed in [71].



72- Algorithm logical scheme

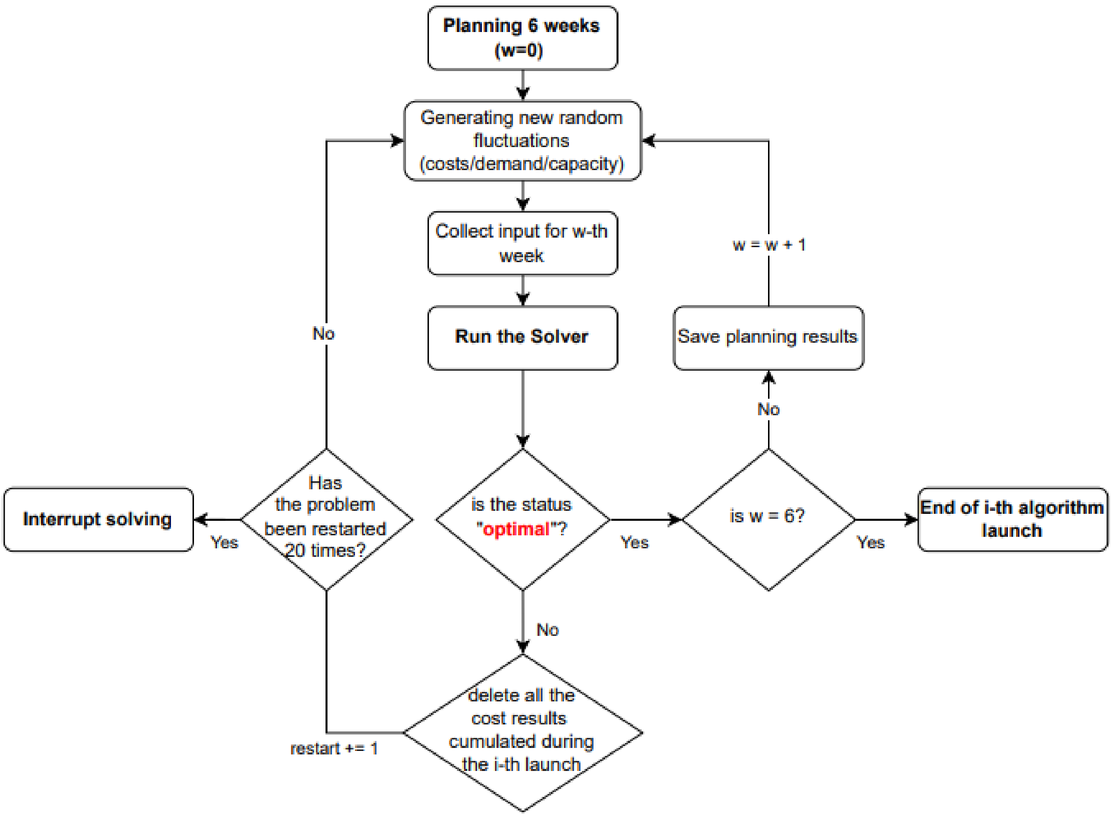
- 7. Before calculating the weekly target, the expansion or compression coefficient of the target cost achieved in the previous week is calculated. The calculation of k1 goes through a series of comparisons. The logic is that if in the current week there is an input backlog and a total average demand lower than those that occurred in the previous week, then the objective target can be reduced by processing a coefficient k1 lower than 1. However, this key step hides other critical issues: making a 6-week programming when, for example, period 1-6 has been covered and programming 2-7 needs to be started, the product demand in the backlog entered in the second step is not that outgoing from week 6 of the first but the one processed at the end of week 1 in the first iteration of the algorithm, which therefore must be saved and kept for the next iteration (as anticipated in step 6). During the other weeks within the same iteration the outgoing backlog value is saved in a list called "delayed_baseline".



73- logical scheme for “Fuzzy” reduction/increasing of target cost

8 – given that the algorithm carries out MRP planning in conditions of "nervousness" of demand (as previously discussed), as it can have random oscillations that do not reconcile with the constraints imposed on the planning, it could happen that the resolution of the problem becomes sometimes unfeasible. For this reason, a portion of lines is inserted within the main code, allowing the problem to start over in case the solver was not able to find a solution by using random demand values different from those used the last time; however, this step requires that all the costs calculated and added to the total ones that were obtained as output during the same iteration must be reset as they are fictitious and

not actually generated. However, it is evident that, if the design suffers from problems related to the "mathematics" of the problem (therefore to the type of constraints chosen or to the elements that condition the value of the coefficients), even a restart of the programming with the generation of different values would not be sufficient to get a solution: to prevent this process from repeating itself indefinitely without having an interruption, the user can choose the number of times that the algorithm is repeated (by default it is set to 20). The logical process is showed in [73].



74 - Logical scheme for algorithm re-start

The usefulness of this step lies in the fact that, at the end of the planning, the user can consult a recap of all the programming, check if there were any "unsolvable" problems and refer to which week, but also know why they were, therefore, be aware of what were the fluctuation values of demand or even of production capacity that were capable of throwing the algorithm into crisis and take measures ahead of time that can prevent the occurrence of these events in reality or amortise their effects.

```
Number of lambda < 1 is: 4
Number of infeasibilities  is: 2
The problem restarted: 2 times
Non-optimal lambda values have been found for weeks: [9, 9, 11, 16, 16, 16]
Average service level: 92.98 %
Week 16 showed some infeasibilities
```

75 - algorithm performance 1

There is a specific condition (currently unsolvable) that could negatively condition the success of the programming and has roots linked to the mathematics of the problem: given that in the definition of fuzzy programming constraints there are fractions where denominators are constituted by a term multiplied by one of the programming coefficient from which the unmultiplied term itself is subtracted (as highlighted in red in the next portion of the code) if the coefficient were to assume a value equal to 1 (because that is the output value from the inference given a certain set of inputs) then 0 will be the divisor. At that point, the problem would become impossible, and the solver would generate an error resulting in a programming crash.

```
problem_th += lamb_th <= (list_coeff[2]*(data_dict_demand[finished_prod][t+1]
+ random_demand_excursion)
[…]
                          + delay_vars_fin[finished_prod])) / ((list_coeff[2]-
1)*(data_dict_demand[finished_prod][t+1] + random_demand_excursion))
```

14.8 **Results evaluation**

Since it was not possible to test the algorithm on real but simulated data, the best way to evaluate the results is to make a series of comparisons. First of all, a fundamental characteristic for a management-type algorithm such as the one in question, which is intended for the MRP phase, lies in ensuring a high "Service Level", a factor considered strategic for the position on the market and the relationship with the customers and, for that matter, one of the main evaluation metrics for this category of algorithms. Being able to ensure it with high values is one of the main missions of companies that produce goods. Making a global assessment of the "level of service" rather than local means understanding what the average insurable value is over an extended period and, above all,

in the presence of more or less significant variations in the determining factors such as demand and available production capacity. For this reason, the algorithm closes the programming by providing the output value of the average service level ensured throughout the time interval with an extension of 6 months (24 weeks); its level is currently between 91-95%. Considering that the programming is done in 6-week steps (therefore 114 total programming) and with a very pronounced demand nervousness (especially in quantitative terms), then the level appears to be quite good, especially if compared with that generated by the equivalent deterministic model which presents a similar score (but using static demand, cost and production capacity values over time).
Another meter is that of the costs generated by the two programs (which certainly, together with the level of service, is one of the main criteria that a real company would evaluate). In most cases, the two are comparable except for the fact that the generation of random cost peaks is repeated for many iterations, leading the total costs to be higher than those evaluated without fluctuations.
In principle, the behaviour of the algorithm is to prefer the use of production hours in extra time in order to penalise the generation of backlog when the cost on the single unit of product is now higher than the production one, thus exceeding the availability of default production capacity. This effect can be explained precisely by introducing those flexibility coefficients of the constraints since if the same input data were processed by the deterministic model (in which the constraints are completely rigid) there would be a greater tendency to generate backlogs (but obviously this also depends on the availability of some resources which the user of the algorithm can rely on).
From the point of view of performance, on the other hand, the algorithm does not significantly involve the device on which it is run but obviously this is a data that should be evaluated with problems of a different size than the one in question (29 rows, 33 columns and 111 elements) which is certainly smaller than in real application cases, which would certainly require a greater commitment of CPU in the resolution. The times

shown in the image [75] are not to be attributed to the entire programming (114 steps) but to the elaboration of the results concerning only one of the weeks.

```
Total iterations:               0
Time (CPU seconds):             0.03
Time (Wallclock seconds):       0.03

Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.05   (Wallclock seconds):       0.05
```

76 - algorithm performance 2

```
Problem MODEL has 29 rows, 33 columns and 111 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
Continuous objective value is 1 - 0.00 seconds
Cgl0003I 0 fixed, 15 tightened bounds, 0 strengthened rows, 0 substitutions
Cgl0003I 0 fixed, 4 tightened bounds, 0 strengthened rows, 0 substitutions
Cgl0003I 0 fixed, 2 tightened bounds, 0 strengthened rows, 0 substitutions
Cgl0004I processed model has 13 rows, 16 columns (12 integer (0 of which binary)) and 71 elements
```

77 - algorithm performance 3

The changes made to the Fuzzy model, i.e., the addition of a greater number of variables, constraints, and nonzero elements (therefore a greater growth of the density of the arrays is manifested) in themselves lead to an increase in the resolution times of the problem if compared with those employed by the deterministic model for its resolution. This effort, which on a small problem is almost irrelevant, is sufficient to justify the beneficial effects due to the increase of the overall flexibility of the problem (this might seem a contradiction with what was written previously about the increase of the constraints but with flexibility we refer to the range of variability of some decision variables that are no longer forced to rigid values).

15. Conclusion and possible future developments

Once all the necessary coefficients and inputs are available, the algorithm can process them to find the optimal solution that reduces costs and guarantees satisfaction of the constraints. Working with elasticised constraints, the algorithm could find different solutions, using the same input, every time it is run. This effect is clearly caused by the constraints themselves as they do not impose a fixed limit on the excursion of the variables but allow their variation limited to a finite interval. This feature is sufficient to confirm that this type of resolution may not be as satisfactory as the deterministic one when the values of all constraints are clearly fixed but, on the other hand, finding an acceptable solution when there are so many uncertainties gives a strong contribution to the planning.

In the light of what was evaluated with the use of the algorithm and the nature of the data tested, it can be considered that it is a valid tool and that it can be integrated within ERP systems (not necessarily in the form in which it is presented but at a more conceptual level) as a support to the algorithms of more traditional use, as it allows to integrate in the planning of the requirements the possibility of evaluating scenarios different from the deterministic ones thanks to the "management of uncertainties". Many production environments (such as that of industrial machinery or the automobile) that must manage a very large number of different products, including raw materials and components, could benefit from this tool since the choices that must be made under conditions of uncertainty have consequences amplified precisely by the greater diversification of the product species. For these reasons, this type of production environment, previously defined as multi-product, multi-level, and multi-period, could not rely exclusively on a "rigid" MRP planning model since it would not be representative of the application reality. Therefore, the constraints elasticisation approach must not be seen as something "unnatural" as it is common practice, in reality, to make these decisions in the field which are often also conditioned by the vision and perception of reality by the designer and by the level and the quality of communication within the departments. The evaluation phase of the programming coefficients through fuzzy inference, in fact, must have a representative purpose of how the communication between physical individuals and the use of different linguistic variables can have a strategic contribution to the design and therefore of how it

is necessary to invest in research and development of systems capable of smoothing out the multiple facets that at a communicative level can negatively interfere with the company's managerial and organisational functions.
In conclusion, some suggestions are proposed on the possible developments that the study in question could have in the future, as it was not possible to further explore them here:

1) test the algorithm in a real production environment (or with data coming from it) to study its behaviour in the field
2) test programming with a constraint relaxation that does not use a linear function (open-right function), but trying to develop non-linear functions
3) increase the number of "perception" levels of the variables that condition the value of the elasticisation coefficients (at the moment set to 3)
4) experiment with using a FIS other than Mamdani to see if is possible to get more reliable results
5) consider the fuzzy behaviour of other variables: for example, the quality of production and purchased products, various lead times and internal/external logistic activities
6) testing the algorithm using different membership functions than the triangular one: the idea would be to deepen the research to understand which are the most representative and significant functions for all the variables involved in the programming
7) improvement of the user interface: development of methods to reduce as much as possible the "direct contact" between the user and the algorithm so that the former does not have to interface directly with mathematical and programming functions but is limited to checking only a few parameters (the idea could be to use an expert system that manages the inputs)
8) integration of other aspects of artificial intelligence (for example the using of reinforcement learning to improve the choice of coefficients)

16. INDEX

16.1 **References**

1. *"The Fourth Industrial Revolution (Industry 4.0): A Social Innovation Perspective".* Morrar, et al. 2017, Technology Innovation Management Review, p. 12-20.

2. *"Digital innovation and the fourth industrial revolution: epochal social changes?".* Caruso, Loris. s.l. : Springer, 2018, Ai & Society, p. 379-392.

3. *Behind the definition of Industry 4.0: Analysis and open questions.* Culot, Giovanna. International Journal of Production Economics.

4. *Industry 4.0: A Supply Chain Innovation Perspective.* Zhou, et al. 2015. 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD).

5. *"Industry 4.0: A Supply Chain Innovation Perspective".* Hahn, Gerd J e et.al. 2020, International Journal of Production Research, p. 1425-1441.

6. *"Cyber-physical systems".* Wolf, Wayne. 2009, p. 88-89.

7. *"Cyber-Physical Systems".* Bahetu, Radhakisan e Helen Gill. 2011, p. 161-166.

8. *"Cyber-physical systems and their security issues.".* Alguliyev, et al. p. 212-223.

9. *"A framework for collaborative robot (CoBot) integration in advanced manufacturing systems.".* Djuric, Ana M., R.J. Urbanic e J.L. Rickli. 2016, SAE Internationa Journal of Materials and Manufacturing 9.2, p. 457-464.

10. *"The expected contribution of Industry 4.0 technologies for industrial performance.* Dalenogare, Lucas Santos. 2018, International Journal of production economics 204, p. 383-394.

11. *"Industry 4.0 implications in logistics: an overview.".* Barreto e Pereira. 2017, Procedia manufacturing, p. 1245-1252.

12. *"Service innovation and smart analytics for industry 4.0 and big data environment".* Lee, Jay e Hung-An Kao. s.l. : Elsevier, 2014, p. 3-8.

13. *"Industrial Process Monitoring in the Big Data/industry 4.0 Era: From Detection, to Diagnosis, to Prognosis.".* Reis, Marco S. e Geert Gins. 2017, Processes, Vol. vol.5, p. 35-.

14. *Opportunities of Sustainable Manufacturing in Industry 4.0.".* Stock, T. e Seliger, G. 2016, Procedia CIRP 40, p. 536-541.

15. *Sustainability outcomes of green processes in relation to industry 4.0 in manufacturing: systematic review.* Vrchota, Jaroslav e et al. 5968, 2020.

16. *Towards Industry 4.0: an overview of European strategic roadmaps.* Sàntos, Claudio e et al. 2017, Procedia Manufacturing, p. 972-979.

17. *INDUSTRY 4.0: EUROPE'S (RE)INDUSTRIALISATION NEEDS A GLOBAL LEVEL.* Kotýnková. Prague, Czech Republic : s.n., 2016.

18. *Factories of the Future: Multi-annual roadmap for the contractual PPP under Horizon 2020.* EFFRA. Brussels : s.n., 2013.

19. *How is COVID-19 altering the manufacturing landscape? A literature review of imminent challenges and management interventions.* Kapoor, Bigdeli e Dwivedi. s.l. : SpringerLink, 2021.

20. Ciarapica, F. e Bevilacqua, M. teaching slides - "Industrial Plant Management". Ancona (AN) - Italy - Faculty of Mechanical and Industrial Engineering UNIVPM : s.n., 2020.

21. John H. Blackstone Jr. *APICS Dictionary.* Department of ManagementTerry College of BusinessUniversity of Georgia : s.n., 2016.

22. Jacobs, F. e Chase, R. *Operations and supply chain management.* 3. s.l. : McGraw-Hill, 2011.

23. *Supply planning under uncertainties in MRP environments: A state of the art.* Dolgui, Alexandre. 2007.

24. *Freezing the master production schedule for material requirements planning systems under demand uncertainty.* Zhao, Xiande e Lee, T.S. Hampton : s.n., 1993, Journal of Operations Management, Vol. 11.

25. Boulaksil, Youssef. Safety stock placement in supply chains with demand forecast updates. *Operations Research Perspectives.* s.l. : Elsevier, 2016, Vol. 3, p. 27-31.

26. *Safety Stocks in MRP-Systems with Emergency Setups for Components.* Carlson, Robert C. e Yano. 4, s.l. : JSTOR, 1986, Vol. 32, p. 403-412.

27. *Determining supply chain safety stock level and.* Amirijabbari e Bhuiyan. 1, 2013, Journal of Industrial Engineering and Management (JIEM), Vol. 7, p. 42-71.

28. *A comprehensive survey of guaranteed-service models for multi-echelon inventory optimization.* Eruguz, A.S., et al. 2016, International Journal of Production Economics. 155713393.

29. *Managing Demand Uncertainty in Supply Chain Planning.* Gupta e Maranas. 8, 2003, Computers & Chemical Engineering , Vol. 27.

30. *Perspectives in Supply Chain Risk Management.* Tang, C.S. 103, 2006, International Journal of Production Economics, p. 451-488.

31. Goncalves, Joao N.C., Sameiro Carvalho, M. e Cortez, P. Operations research models and methods for safety stock determination: A review. *Operations Research Perspectives.* s.l. : ELSEVIER, 2020.

32. Guide Jr, V.D.R. e Srivastava, R. A review of techniques for buffering against uncertainty with MRP systems. *The Management of Operations.* 2010, Vol. 11, p. 223-233.

33. *Information Distortion in a Supply Chain: The Bullwhip Effect.* Hau L. Lee, V. Padmanabhan, Seungjin Whang. 1 April 1997.

34. *Proposed Model for Inventory Review Policy through ABC Analysis in an Automotive Manufacturing Industry.* Dr.S.Nallusamy e et.al. Jadavpur University Kolkata : s.n., March 2017, International Journal of Engineering Research in Africa, p. 164-174.

35. *Inventory Management and Reorder Point (ROP) Strategy Using ABC Analysis Methods in Textile Manufacture.* Teuku Faisal Umry, Moses Laksono Singgih. 5, Surabaya - Indonesia : s.n., 2019, IPTEK - journal of Proceedings Series.

36. *Setting MRP Parameters and Optimizing the Production Planning Process.* Malindzakova, M., et al. 4, s.l. : MDPI, Vol. 10, p. April.

37. *Material Requirements Planning Under Demand Uncertainty Using Stochastic Optimization.* Thevenin, Adulyasak e Cordeau. 1 October 2020, Production and Operations Management, Vol. 30, p. 475-493.

38. *Strategic Safety-Stock Placement in Supply Chains with Capacity Constraints.* Graves e Schoenmeyr. 2016, Manufacturing & Service Operations, p. 445-460.

39. *Strategic Inventory Placement in Supply Chains: Nonstationary Demand.* Graves, S. e Willems, S. 4 January 2008, Manufacturing & Service Operations Management.

40. *Material Management without forecasting: from MRP to demand-driven MRP.* Kortabarria, A. e et al. 2018, Journal of Industrial Engineering and Management, p. 632-650.

41. *Effective production planning for purchased part under long lead time and uncertain demand: MRP vs demand-driven MRP.* Shofa, M., Moeis, A.O. e Restiana, N. s.l. : IOP Publishing, 2018, IOP Conference Series: Materials Science and Engineering.

42. Ptak e Smith. *Demand Driven Material Requirements Planning (DDMRP) (Volume 1).* s.l. : Industrial Press, 2016.

43. *Strategic Inventory Positioning in BOM with Multiple Parents Using ASR Lead Time.* Jingjing Jiang e Suk-Chul Rim. 2016, Mathematical Problem in Engineering.

44. *Demand Driven Institute.* [Online] https://www.demanddriveninstitute.com/ddmrp.

45. Ptak e Smith. *Orlicky's material requirement planning.* s.l. : Mc Graw Hill, 2011.

46. *Artificial Intelligence and Machine Learning in Smart Production: Progress, Trends, and Directions.* Cioffi, R., et al. 2020, Vol. 12.

47. *An artificial, intelligence-based production scheduler.* De Toni, A., Nassimbeni, G. e Tonchia, S. 1996.

48. *Application of Industry 4.0 in the procurement process of supply chians: a systematic òiterature review.* Jahani, N. e et al. 2021.

49. APA. Chollet F. *Deep learning with Python.* s.l. : Manning Publications, 2017.

50. *Supervised learning - Machine learning techniques for multimedia.* Cunningham, et al. Berlin - Heidelberg : Springer, 2008.

51. *An empirical comparison of supervised learning algorithms.* Caruana, Rich e Niculescu-Mizil. 2006. Proceedings of the 23rd international conference on Machine learning.

52. *A survey on semi-supervised learning.* Van Engelen e Holger. 2020, p. 373-440.

53. Plaat, Aske. *Deep Reinforcement Learning.* s.l. : Springer, 2022.

54. *Playing Atari with deep reinforcement learning.* Mnih, Volodymyr e et al. 2013.

55. *Reinforcement learning: A survey.* Kaebling, et al. 1996, Journal of artificial intelligence research, p. 237-285.

56. *Data Mining and Knowledge Discovery.* Van Leeuwen, Matthijs e Knobbe, Arno. 2012, p. 208-242.

57. *Machine learning: Best way to sustain the supply chain in the era of Industry 4.0.* Devashish Nagar, et al.

58. *System Reliability, Qulaity Control, Safety, Maintenance and Managament Applications to Civil, Mechanical and Chemical Engineering.* Gunjan, Vinit Kumar e et al. 2019. ICRRM conference proceedings.

59. *What are artificial neural networks?* Krogh e Anders. 2008. p. 195-197.

60. Vasilev, Ivan e et. al. *Python Deep Learning: Exploring deep learning techniques and neural network architectures with Pytorch, Keras, and TensorFlow.* s.l. : Packt Publishing, 2019.

61. *Analysis of the Application Efficiency of TensorFlow and PyTorch in Convolutional Neural Network.* Novac, et al. 22, Vol. 22.

62. *Backpropagation Learning iN Expert Networks.* Lacher, R., Hruska, S. e Kuncicky, D.C. 1, 1992, Vol. 3.

63. Medsker, L. e Turban, E. *Neural computing and artificial intelligence.* 1992. p. 621-663.

64. *Exploring strategies for training deep neural networks.* Larochelle, H., Bengio, Y. e Louradour, P. Jan 2009, Journal of Machine Learning Research, p. 1-40.

65. *Changeable manufacturing - classification, design and operation.* Wiendahl, H.P., et al.

66. *Forecasting software: Past, present and future.* Kusters, U., McCullough, B.D. e Bell, M. 2006, p. 599-615.

67. *Exploring the use of deep neural networks for sales forecasting in fashion retail.* Loureiro, A.L., Miguéis, V.L. e da Silva, L.F. 2018, p. 81-93.

68. *Demand forecasting for textile products using statistical analysis and machine learning algortihms.* Lorente-Leyva, Leandro L. e et al. s.l. : Springer, 2021. Asian Conference on Intelligent Information and Database Systems. p. 184-194.

69. *Artificial Neural Networks in the Demand Forecasting of a Metal-Mechanical Industry.* Lorente-Leyva, Leandro L., Patino-Alarcon, Delio R. e Montero, Yakcleem. 2020, Vol. 15, p. 81-87.

70. Coupling Damage-Sensing Particles to the Digital Twin Concept. [Online] January 2018. https://ntrs.nasa.gov/search.jsp?R=20140006408.

71. *Digital Twin: Manufacturing Excellence Throught Virtual Factory Replication.* Hankel, M. e Rexroth, B. 2015.

72. Medsker, Larry R. *Hybrid Intelligent Systems.* s.l. : Springer Science & Business Media, 2012.

73. Zadeh, L.A. *Fuzzy sets as a basis for a theory of possibility.* 1999. p. 3-28. Vol. 100.

74. *Fuzzy logic, neural networks, and soft computing.* Zadeh, L.A. s.l. : ACM Dgital Library, Vol. 37, p. 77-84.

75. *utline of a New Approach to the Analysis of Complex Systems and Decision Processes.* Zadeh, L.A. 1973, IEEE Transactions on Systems, Man, and Cybernetics, p. 28-44.

76. *The Concept of a Linguistic Variable and Its Application to Approximate Reasoning.* Zadeh, L.A. 1975, Vol. Information Sciences, p. 199-249.

77. *Fuzzy logic in natural language processing.* Novàk, V. Naples : s.n., 2017. IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).

78. *From knowledge-based to data-drivem fuzzy modelling.* Hullermeier, E. 2015, p. 500-509.

79. Dantzig, George. *Linear Programming and extensions.* Princeton University Press, Princeton : RAND, 2015.

80. Murota , Kazuo. *"Linear Programming".* s.l. : Springer, 2020.

81. Pandian, M., Nagrajan, R. e Sazali, Y. Fuzzy Linear Programming: a Modern Tool for Decision Making. *Computational Intelligence for Modelling and Prediction.* 2005, p. 383-401.

82. *Fuzzy logic in artificial intelligence.* Slany, Wolfgang e Klement, Erich Peter . Graz : s.n., 1996, Encyclopedia of Computer Science and Technology, Vol. 34.

83. Siler, W. e Buckley, J. *Fuzzy Expert Systems and Fuzzy Reasoning.* s.l. : WILEY, 2005.

84. *Fuzzy Sets, Fuzzy Logic, Fuzzy Methods with Applications.* Gottwald e Bandemer. s.l. : John Wiley & Sons, 1995.

85. Bellman, Richard e Zadeh. Decision-Making in a Fuzzy Environment. *Management and Science.* Vol. 17, 4.

86. *Temperature Control using Fuzzy Logic.* Singhala, P., Patel e Shah. 1, Department of instrumentation and control, Sarvajanik College of Engineering and Technology Surat, Gujarat, INDIA : s.n., 1 January 2014, International Journal of Instrumentation and Control Systems (IJICS) , Vol. 4.

87. Foundations of Fuzzy Logic. *MathWorks.* [Online] https://es.mathworks.com/help/fuzzy/foundations-of-fuzzy-logic.html.

88. *Comparative Analysis of Membership Function on Mamdani Fuzzy Inference System for Decision Making.* Putri , harliana e Robbi , Rahim. Medan, Sumatera Utara, Indonesia : IOPScience, 2017, Journal of Physics: Conference Series.

89. *Comparison of Mamdani-Type and Sugeno-Type Fuzzy Inference Systems for Fuzzy Real Time Scheduling.* Blej, Mohammed e Azizi, M. 2016. Corpus ID: 39679343.

90. Levesque e Lakemayer. *The logic of knowledge bases.* s.l. : MIT Press, 2001.

91. Kandel, Abraham. *Fuzzy expert systems.* s.l. : CRC Press, 1991.

92. Grosan, C. e Abraham, A. Swarm Intelligence. *Intelligent systems: a modern approach.* Berlin : Springer, 2011, p. 409-422.

93. *Personal Construct Theory and the Transfer of Human Expertise.* Boose, J. 1984. AAAI Conference on Artificial Intelligence.

94. Durkin, John. *Expert systems : design and development.* New York : Macmillan, 1994.

95. Kidd, A.L. *Knowledge acquisition for expert systems: a practical handbook.* s.l. : Plenum Press, 1987.

96. Turban, E. *Expert Systems and Applied Artificial Intelligence.* New York : Macmillan Publishing Company, 1992.

97. Slany, Wolfgang. Scheduling as a fuzzy multiple criteria optimization problem - Technical report, Christian Doppler Laboratory fro Expert Systems. Technical University of Vienna : s.n., July 1997.

98. Łukasiewicz, Jan. *Aristotle's syllogistic : from the standpoint of modern formal logic.* s.l. : Oxford : At the Clarendon Press, 1957.

99. Piano Nazionale Industria 4.0. [Online] https://www.mise.gov.it/it/.

100. *Artificial Intelligence Applications in Supply Chain Management.* Pournander, et al. 2021, International Journal of Production Economics.

101. *Artificial intelligence and social responsibility: the case of the artificial intelligence strategies in the United States, Russia, and China.* Saveliev e Zhurenkov. 2020, Vol. 50, p. 656-675.

102. *A Literature Review on Artificial Intelligence.* Oke, S.A. 2008, International journal of information and management sciences. Corpus ID: 59874180.

103. *Application potentials of artificial intelligence for the design of innovation processes.* Vocke, Costantinescu e Popescu. 2019, Vol. 84, p. 810-813.

104. *Artificial Intelligence and Machine Learning Applications in Smart Production: Progress, Trends, and Directions.* Cioffi, et al. Napoli / Cassino : s.n., January 2020.

105. *Application of Industry 4.0 in the Procurement Processes of Supply Chains: A Systematic Literature Review.* Niloofar Jahani, Arash Sepehri e Hadi Rezaei. s.l. : MDPI, 6 July 2021.

106. *Analysis of the Challenges of Artificial Intelligence of Things (AIoT) for the Smart Supply Chain (Case Study: FMCG Industries).* Nozari, Hamed, Szmelter-JaroszJ, Agnieszka e Ghahre, David . Iran / Poland : MDPI, 11 April 2022.

107. *Addressing Industry 4.0 Cybersecurity Challenges.* Culot, et al. 3, 1 thirdquarter September 2019, Vol. 47, p. 79-86.

108. *Relationships between Indsutr 4.0, sustainable manufacturing and circular economy: proposal of a research framework.* Bag, S. e Pretorius, J.H.C. 4, 2022, International Journal of Organisational Analysis, Vol. 30, p. 864-898.

109. *A Multi-objective Approach to Simultaneous Strategic and Operational Planning in Supply Chian Design.* Sabri, E.H. e Beamon, B.M. 2000, Vol. 28, p. 581-598.

110. *Simulation Analysis of a Multi-item MRP System Based on a Factorial Design.* Sun, L., et al. 2009.

111. *A Simulation Based Optimization Approach to Supply Chain Management under Demand Uncertainty.* Jung, J., et al. 10, 2004, Vol. 28.

112. *Fuzzy goal programming for material requirement planning under uncertainty and integrity conditions.* Diaz-Madronero, M., Mula , J. e Jimenez, M. 2014, International Journal of Production Research, p. 6971-6988.

113. Mula, J. e Madronero, M. Solution Approaches for Material Requirement Planning* with Fuzzy Costs. *Industrial Engineering: Innovative Network.* s.l. : Springer, 2012, p. 349-357.

114. *MRP with flexible constraints: A fuzzy mathematical programming approach.* Mula, J., Poler, R. e Sabater. p. 74-97.

115. Mula, Poler e Sabater. Material Requirement Planning with fuzzy constraints and fuzzy coefficients. *Fuzzy Sets and Systems.* s.l. : Elsevier, 2007, p. 783-793.

116. *Industry IoT Consortium.* [Online] https://hub.iiconsortium.org/homepage.

117. Architecture Alignment and Interoperability - An Industrial Internet Consortium and Plattform Industrie 4.0 Joint Whitepaper. *industrial internet CONSORTIUM.* [Online] https://www.iiconsortium.org/pdf/JTG2_Whitepaper_final_20171205.pdf.

118. *Reference Models for Digital Manufacturing Platforms.* Fraile, et al. Universitat Politècnica de València / Escuela Politécnica Superior de Alcoy : s.n., 2019.

119. The Industrial Internet of Things Volume G1: Reference Architecture. *Industry IoT Consortium.* [Online] 19 June 2019. https://www.iiconsortium.org/.

120. *A heuristic scheduling algorithm based on fuzzy logic and critical chain project management.* Mazzuto, G., Bevilacqua, M. e Ciarapica, F.E. 1 December 2017, International Journal of Project Organisation and Management, p. 303-327.

121. *Target costing for supply chain management: criteria and selection.* Lockamy III e Smith. s.l. : Emerald Group Publishing Limited, 2000, Industrial Management & Data Systems, Vol. 100, p. 210-218.

122. *Evaluation of Models for Forecasting the Final Cost of a Project.* Zwikael, O., Globerson, S. e Raz, T. March 2000, Project Management Journal, Vol. 31, p. 53-57.

123. *A prediction model to forecast the cost impact from a break in the production schedule.* Delionback, L.M. 1977, Journal Cost of Estimating, Vol. 8.

124. *Ranking Fuzzy Numbers through the Comparison of its Expected Intervals.* Jiménez, M. 1996, International Journal of Uncertainty, Vol. 4, p. 379-388.

125. *Linear programming with fuzzy parameters: An interactive method resolution.* Jiménez, et al. s.l. : Elsevier, 16 March 2007, European Journal of Operational Research, Vol. 177, p. 1599-1609.

126. Liu e Shi. A fuzzy programming approach for solving a multiple criteria and multiple constraint level linear programming problem. *Fuzzy Sets and Systems.* s.l. : Elsevier, 1994, Vol. 65, p. 117-124.

127. *Python - the fastest growing programming language.* Srinath, K.R. 2017, International Research Journal of Engineering and Technology, p. 354-357.

128. *Simpful: A User-Friendly Python Library for Fuzzy Logic.* Spoalor, et al. s.l. : Atlantis Press, 2020, International Journal of Computational Intelligence Systems, Vol. 13, p. 1687-1698.

129. *Performance evaluation of web development technologies in php, python, and node.js.* Lei, et al. 2014.

130. what is apache kafka. *redhat.* [Online] https://www.redhat.com/it/topics/integration/what-is-apache-kafka.

131. *MongoDB in action: covers MongoDB version 3.0.* Banker, Kyle e et al. s.l. : Simon and Schuster, 2016.

132. advantages-of-mongodb. *MongoDB.* [Online] https://www.mongodb.com/.

133. *Time series database and influxdb.* Naqvi, Syeda Noor Zehra e Yfantidou. Université Libre de Bruxelles : s.n., 2017.

134. time-series-database. *kx.com.* [Online] https://kx.com/.

135. *influxdata.* [Online] https://www.influxdata.com/time-series-platform/.

136. *"The fourth industrial revolution (Industry 4.0): A social innovation perspective.".* 2017.

137. *Safety stock placement problem in capacitated supply chains.* Sitompul, C. e Aghezzaf, El. 17, s.l. : Taylor&Francis, 18 July 2008, International Journal of Production Research, Vol. 46, p. 4709-4727.

138. Medsker, Larry R. *Hybrid Intelligent Systems.* s.l. : Springer Science and Business Media, 2012.

139. *Fuzzy goal programming for material requirements planning under uncertainty and integrity conditions.* Mula, Josefa, Diaz-Madronero, Manuel e Jimenez, Mariano. 24 April 2014, International Journal of Production Research, Vol. 52, p. 6971-6988.

140. *Changeable manufacturing - classification, design and operation.* Wiendahl , H.P., et al. 2007, p. 783-809.

141. Business Motivation Model (BMM), Object Management Group (OMG), http://www.omg.org/spec/BMM/1.3/

16.2 **Web pages index**

[wb-1]. https://ec.europa.eu/info/files/presentation-horizon-europe_en

[wb-2]. https://aideas-project.eu/project/

[wb-3]. www.demanddriveninstitute.com

[wb-4]. https://www.spiceworks.com/tech/artificial-intelligence/

[wb-6]. https://www.iiconsortium.org

[wb-7]. https://www.docker.com/

[wb-8]. https://www.html.it/guide/docker/

[wb-9]. https://docs.docker.com/

[wb-10]. https://reactjs.org/

[wb-11]. https://blog.hubspot.com/

[wb-12]. https://grafana.com/

[wb-13]. https://technoglitz.com/italy/

[wb-14]. https://grafana.com/docs/

[wb-15]. https://www.python.org/

[wb-16]. https://nodejs.org/en/

[wb-17]. https://redis.io/

[wb-18]. https://devacademy.it/node-js/

[wb-19]. https://kafka.apache.org/

[wb-20]. https://kafka.apache.org/documentation

[wb-21]. https://www.mongodb.com/home

[wb-22]. https://www.influxdata.com/

[wb-23]. https://neo4j.com

[wb-24]. https://neo4j.com/top-ten-reasons/

[wb-25]. https://www.postgresql.org/

[wb-26]. https://kinsta.com/it/

[wb-27]. https://www.timescale.com/

[wb-28]. https://www.linuxadictos.com/

[wb-29]. https://pypi.org/project/PuLP/

[wb-30]. https://coin-or.github.io/pulp/

[wb-31]. https://code.visualstudio.com

[wb-32]. https://codecrucks.com/what-is-fuzzy-membership-function-complete-guide/

16.3 **Figures Index**

1 - HORIZON Europe Logo 7

2 - AIDEAS Logo 7

3 - AIDEAS Machine Passport 10

4- AIDEAS procurement optimiser 10

5 - MPS Scheme by optiptoerp.com 14

6 - Assemble to Order / Make to Order strategies. 17

7 - MRP logic scheme 21

8 - safety stock effect 22

9 - EOQ strategy 25

10- EOI strategy 27

11- BOM and requirement matrix 29

12 - CRP before corrective actions 30

13 - CRP after corrective functions 31

14 - ABC Analysis 32

15: DDMRP 1 (44) 34

16- DDMRP 2 (44) 34

17 – DDMRP Logic (42) 35

18- DDMRP colored “zones” 35

19 - Machine Learning in Artificial Intelligence (49) 39

20- Reinforcement learning behaviour - schematic figure (wb-4) 41

21- AI's Sphere (AI4Diversity - LinkedIn) 48

22 - AND / OR applied on fuzzy sets. 54

23- Fuzzification - Inference - Defuzzification 55

24 - Membership value on a Triangular MF 56

25- Trapezoidal MF 57

26 - Open right and open left MF 57

27 - Temperature feeling as fuzzy sets. 58

28- inference logic process 58

29- Three triangular MF for quality representation 60

30- Fuzzy resolution for the "tipping problem" - 1 61

31- Fuzzy resolution for the "tipping problem" - 2 62

32 - Takagi Sugeno FIS 63
33 - Centroid defuzzification method (COG-1) 64
34 - Centroid defuzzification method (COG-2) 65
35 - CoA calculation 66
36 - Weighted average method 66
37 - Fuzzy Expert System logic 67
38 - inference chain 69
39 - Research for AI in Industrial Management (1) 71
40 - Research for AI in Industrial Management (2) 72
41 - Research for AI in Industrial Management (3) 72
42 - IIRA Architecture (Industrial Internet Consortium Resource Hub) 77
43 - Business ViewPoint 1 (Industrial Internet Consortium Resource Hub) 78
44 - Business ViewPoint 2 (Industrial Internet Consortium Resource Hub) 78
45 -The roles 90
46- Algorithm functional features 93
47- MRP optimisation algorithm conceptual task flow 95
48- MRP input organisation 96
49 - algorithm functional diagram 97
50 - Architecture for AIDEAS AI Runtime 100
51 - DOCKER logo 100
52 - REACT logo 101
53 - GRAFANA logo 102
54 - Python logo 102
55 - NODE js & Redis logo 103
56 - KAFKA logo 103
57 - mongo DB logo 104
58- influxdb logo 104
59 - neo4J logo 104
60 - POSTGRESQL logo 105
61 - Timescale logo 105
62- MRP Programming Input 109
63- Open right function for constraints elasticisation 113

64- Demand Value 3 MF 120
65 - Demand Value 7 MF 120
66 -Open right function for constraints elasticisation: variable input target 122
67- pulp functionalities 129
68 - output membership activity 137
69 - "k" coefficient calculation 137
70 - algorithm output 140
71 - Decision variables programmed by the algorithm (week 2) 141
72- Algorithm logical scheme 145
73- logical scheme for “Fuzzy” reduction/increasing of target cost 146
74 - Logical scheme for algorithm re-start 147
75 - algorithm performance 1 148
76 - algorithm performance 2 150
77 - algorithm performance 3 150

16.4 **Tables Index**

Table 1 - Mamdani & Sugeno FIS differences 63

Table 2 - Stakeholders levels 85

Table 3 - Features for the proposed AIDEAS solution 106

16.5 **Equations Index**

Equation I - withdrawal of product 23

Equation II - normalised variable Z 23

Equation III - Reorder point 23

Equation IV - safety stock 24

Equation V - Total Cost 26

Equation VI - EOQ 26

Equation VII - reorder level 26

Equation VIII - membership value 53

Equation IX: membership value - Triangular MF 1 56

Equation X - membership value - Triangular MF 2 56

Equation XI - membership value - Trapezoidal MF 57

Equation XII - Center of gravity (1) 65

Equation XIII - Center of gravity (2) 65

Equation XV: Linear MF (constraints elasticisation) 112

Equation XVI - meaning of "min operator" 113

Equation XVII - MRP Deterministic Objective 114

Equation XVIII : Det. Constraint 1 115

Equation XIX: Det. Constraint 2 116

Equation XX: Det. Constraint 3 116

Equation XXI: Det. Constraint 4 116

Equation XXII: Fuzzy Constraint 1 117

Equation XXIII: Fuzzy Constraint 2 117

Equation XXIV: Fuzzy Constraint 3 117

Equation XXV : Fuzzy Constraint 4 118
Equation XXVI: Fuzzy Constraint 5 118