Fuzzy Multi-Item Newsvendor Problem: An Industrial Application in a Cloud Environment

Rodrigo Manuel Madeira Cunha Luís

Thesis to obtain the Master of Science Degree in

Mechanical Engineering

Supervisors: Prof. João Miguel da Costa Sousa
Prof. Luís Manuel Fernandes Mendonça

Examination Committee

Chairperson: Prof. Carlos Baptista Cardeira
Supervisor: Prof. Luís Manuel Fernandes Mendonça
Member of the Committee: Prof. João Filipe Pereira Fernandes

December 2021

Acknowledgments

I would like to express my gratitude in the Portuguese language:

Aos meus pais e à minha irmã, que durante todo o meu percurso pessoal, académico e profissional sempre se mantiveram a meu lado com amizade, carinho e as palavras certas para que me fosse possível crescer e evoluir a todos os níveis, ao longo dos anos. Sem o seu apoio incondicional este projeto seria impossível. Quero agradecer com especial atenção à minha mãe que deu sempre o seu melhor para me ajudar e orientar em todas as etapas da minha vida Obrigado mãe!

Agradeço também ao resto da minha família, que sempre esteve presente para mim, reforçando o agradecimento aos meus avós, tio e tia. O apoio destas pessoas cruciais na minha vida, ao longo dos anos, contribuiu de forma crucial para o que sou hoje, nomeadamente no meu desenvolvimento tanto pessoal quanto técnico.

Em termos académicos, gostaria de expressar a minha gratidão para com os meus supervisores, o Professor João M.C. Sousa e Professor Luís Mendonça. O aconselhamento e orientação que me deram foi indispensável no desenvolvimeto e enquadramento deste trabalho. Seguindo nesta linha de pensamento, expresso também a minha gratidão a alguns docentes que tiveram um papel importante no meu percurso académico. O Professor Valter, Professor Romero, Professor Margarida, Professor Inês, Professor Manuela, Professor Diogo e Professor Ana Rita.

Como não poderia deixar de ser, e faz todo o sentido para mim, mostrar o meu reconhecimento para com a Siemens por ter confiado em mim a 200% e por me ter dado esta oportunidade de crescer, aprender e desenvolver variadas competências com pessoas absolutamente extraordinárias. Dentro destas pessoas destaco o Rui, Silva e Vieira, ou mais conhecidos como ”Malta da Pesada”, Sem os seus conselhos, amizade e conhecimento o sucesso deste projeto seria uma mera miragem.

Por fim, mas não menos importante, gostaria de agradecer a todas as pessoas que de alguma forma marcaram a minha vida. Em especial aos meu amigos mais próximos: Batista, João, Carlos, Sara, Tomás, Marta, Bia, Martim, Rita, Joana, Pedro, Rubén e Matos.

A todos vocês – Muito Obrigado.

Abstract

This work proves the Fuzzy Multi-Item Newsvendor problem formulation is an alternative to the probability methods with the advantage of easily integrating human-expertise knowledge and machine learning when data is insufficient to accurately predict demand. The formulation consists of three fuzzy decision-making policies optimized through a genetic algorithm. The proposed solution improved previous literature by introducing enhancements in the Fuzzy formulation, a genetic algorithm with new problem-specific mechanisms and parallel computing in a cloud environment. Additionally, the thesis introduces an uncertainty simulation procedure that compares different solutions based on the generated profit instead of objective functions. On the one hand, these results proved the indispensability of the new problem-specific features with cases where the profit generated increased by 55 %. On the other hand, the parallel computing in a cloud environment ensured the solution scalability with a running time reduction of 98,3 %.

Keywords

Newsvendor problem; Single-period inventory; Fuzzy Logic; Genetic algorithm; Cloud environment.

Resumo

É comum em produtos inovadores e disruptivos não haver informação suficiente para se prever de uma forma precisa as curvas de procura probabilísticas necessárias para as soluções tradicionais de planeamento de inventario. Este trabalho estuda o problema do vendedor de jornais e prova que uma formulação *"Fuzzy"* é uma alternativa válida aos métodos probabiliticos tendo a vantagem de facilmente integrar conhecimento humano ou "machine learning" quando é necessário compensar a escassez de dados. A solução proposta melhorou anteriores trabalhos introduzindo melhoramentos na formulação "Fuzzy", um algoritmo genético com novos métodos específicos para este problema e computação em paralelo num ambiente em nuvem. Adicionalmente, a tese introduz um procedimento de simulação de incerteza que usa possíveis cenários de procura para avaliar as diferentes soluções baseado no lucro gerado e não em funções objetivo. Por um lado, estes resultados provaram a imprescindibilidade dos novos métodos introduzidos, havendo casos em que o lucro gerado aumentou em 55 %. Por outro lado, o uso de computação em paralelo através da "cloud" assegurou a escalabilidade da solução com reduções de tempo de computação na ordem dos 98,3%.

Palavras Chave

Problema do vendedor de jornais; Lógica Fuzzy; Algoritmo genético; Ambiente Cloud

Nomenclature

Classical Formulation

N:	Total number of items
i:	Item index
v_i:	Cost of revenue loss per unit of item *i*
h_i:	Cost incurred per item *i* for leftover at the end of the specific period
c_i:	Cost per unit of item *i*
x_i:	Ordering quantity of item *i* (decision variable)
D_i:	Random demand of item *i*
$f_i(D_i)$:	Demand probability density function of item *i*
$F_i(D_i)$:	Demand cumulative distribution function of item *i*
E_i:	Expected cost function of item *i*
E:	Total expected cost function
B:	Budget function
B_G:	Budget available
μ_i:	Mean value of the probabilistic distribution of item *i*
σ_i:	Standard deviation of the probabilistic distribution of item *i*

Fuzzy Formulation

ξ:	Fuzzy variable
$Pos\{\xi \geq r\}$:	Possibility of the fuzzy variable ξ assuming values higher or equal than r
$Nec\{\xi \geq r\}$:	Necessity of the fuzzy variable ξ assuming values higher or equal than r
$Cr\{\xi \geq r\}$:	Credibility of the fuzzy variable ξ assuming values higher or equal than r
$E[\xi]$:	Expected value of fuzzy variable ξ
u_k:	Demand Vector
$m(u)$:	Membership grade of element u
$\widetilde{Pos}\{F(x,u_k) \geq F_0\}$:	Estimated Possibility of solution x generating a profit higher than F_0
$\widetilde{Nec}\{F(x,u_k) \geq F_0\}$:	Estimated Necessity of solution x generating a profit higher than F_0
$\widetilde{Cr}\{F(x,u_k) \geq F_0\}$:	Estimated Credibility of solution x generating a profit higher than F_0
N_{cr}:	Total number of Credibility samples.
P_{min}:	Minimum profit target for Credibility sampling
P_{max}:	Maximum profit target for Credibility sampling
P_{int}:	Profit targets interval for Credibility sampling
$p(n)$:	Probability of element n for probability mapping

Genetic Algorithm

T_size:	Tournament size
T_coef:	Tournament coefficient
Pop_size:	Population size
R_{ratio}:	Resizing ratio
x_{ni}:	Normalized ordering quantity of item i

Acronyms

ACO	Ant Colony Optimization
CC	Core Coefficient
CU	Currency Unit
EOQ	Economic Order Quantity
EPQ	Economic Production Quality
GIM	Generic Iterative Method
GA	Genetic Algorithm
IA	Immune Algorithms
PSO	Particle Swarm Optimization
MF	Membership Function(s)
MINP	Multi-Item Newsvendor Problem
NQM	Number of Random Demand Vectors per Grade Quantile and per Material
SA	Simulated Annealing
SC	Support Coefficient
TS	Tabu Search
vCPU	Virtual Central Processing Units

Contents

List of Figures

List of Tables

1

Introduction

Contents

1.1 Motivation

Every day, a newsvendor needs to buy journals based on uncertain demand. Assuming each journal has a fixed cost and selling price, if he asks for too many journals and the demand is not enough, there is a reduction in the profit. Contrary, if the demand is higher than the number of journals ordered, potential sales do not happen, resulting in ”lost profits” [5]. Based on this dilemma, a fundamental problem on inventory management takes shape: ”The Newsvendor Problem” [6].

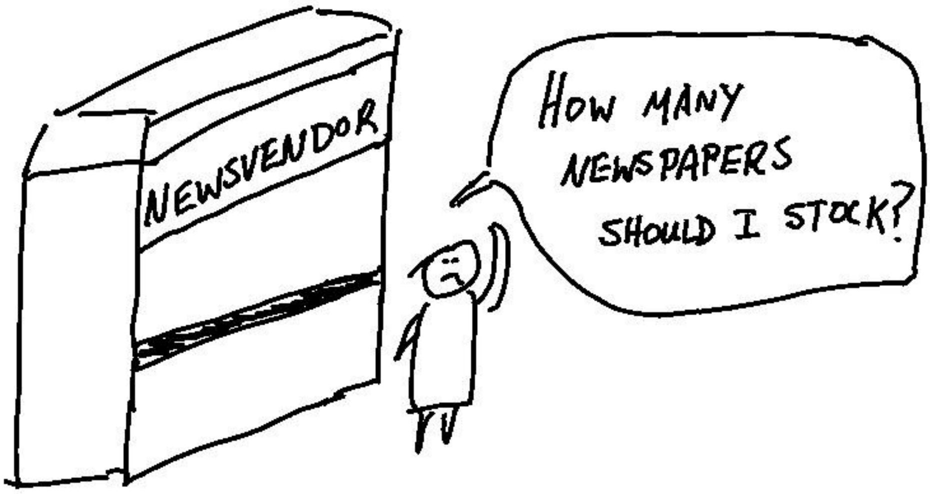

Figure 1.1: Newsvendor Problem Illustration (source [1])

This problem dates back to the nineteenth century when F. Y. Edgeworth first introduced it with his work: ”The Mathematical Theory of Banking” [7]. From that moment, the number of academic papers published on this topic has been increasing year by year [8], being inventory management also addressed by the set of solutions framed in the Industry 4.0 concept [9]. To keep up with the increasing relevance of this topic, Siemens proposed the challenge of reviewing the as-is works and build a framework capable of solving a real Multi-Item Newsvendor Problem (MINP).

The literature offers a wide range of solutions to solve the Newsvendor Problem (see [5]). Focusing on the MINP, solutions vary from the number of constraints and their type (costs, service level, etc.), decision-making policies (optimize expected profit, service level, etc.) or risk-averse techniques. However, the majority of these methods uses probability density functions to model the uncertain demand, what can be a limitation.

Probabilistic functions are difficult to derive or understand in real scenarios. This difficulty is specially true in innovative and disruptive products, where there is insufficient data to predict the demand probability distribution accurately. The integration of human expertise knowledge and machine learning can remove these limitations. Fuzzy logic is a suitable tool to perform this integration having literature works that prove its effectiveness in Newsvendor problems [4, 10, 11]. A Fuzzy environment can use few data points to describe uncertainty through meaningful Membership Function(s) (MF) (further see section 3.2). Furthermore, Fuzzy logic offers an ideal environment to describe the vagueness of human thinking through mathematical operations, precisely defining linguistic terms such as " around 2000" without the assumptions or discretization a probability approach would have. As an example, properties such as trust intervals, which are a common practice in the industry, can define the parameters of an trapezoidal MF (further see section 3.5.1).

The first Fuzzy solution for an inventory management problem dates back to 1995 [2]. A year later, Petrovic [10] applied this methodology to the Newsvendor Problem, which is a specific inventory management problem, forming the first **Fuzzy Newsvendor problem**. Since there, the interest in using this technique to solve inventory management problems has been increasing. Figure 1.2 shows this growth over the years. The figure is composed by the work of [2] that goes until 2015 and a rough estimation when searching for "Fuzzy inventory" on the *Google Scholar* platform.

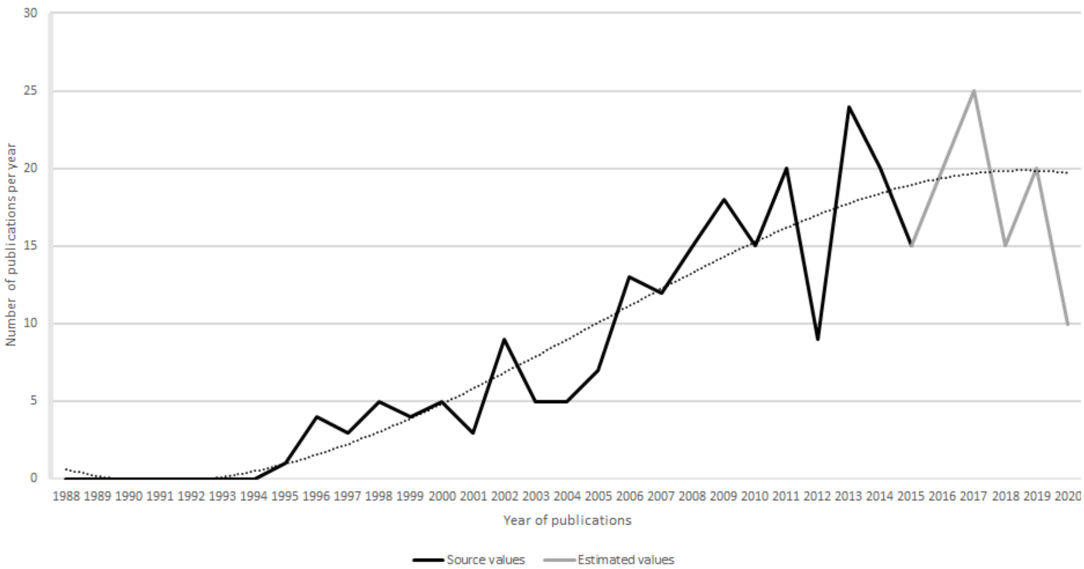

Figure 1.2: Fuzzy inventory management related papers per year estimated until 2020 (source [2] until 2015)

Given the growing academic relevance and capacity of solving this type of problems (section 3.1), Fuzzy logic was used to formulate this problem. The Fuzzy formulation proposed defines three fuzzy decision-making policies that must be optimized using optimization algorithm. The work implemented a Genetic Algorithm (GA) for this purposed, enhancing the work of [4] with new problem-specific mechanisms and computation techniques that ensure reproducibility in bigger problems. In the end, the intention is to develop an end-to-end scalable solution capable of deriving results as good as the academic ones but at the same time capable of being easily integrated into a real-world scenario.

1.2 Contributions

The following topics summarize the academic added-value proposal:

Enhancements in the Credibility estimation (Chapter 3)

This thesis simultaneously improves the performance and running time of the Credibility estimation framework proposed in [4]. Section 3.3.3 contains these enhancements being them an **early solution rejection** to discard solutions that violate constraints before the Credibility assessment, an **identification of inexplicable solutions** and an **adjustable** α_{cut} that tests the proposed solution against meaningful scenarios instead of doing it purely randomly.

Problem-Specific enhancements (Chapter 4)

Section 4.3 presents novel mechanisms in the genetic algorithm to suit the Newsvendor problem better. These mechanisms include an Initialization with Null values for low budget problems, a solution resizing to increase the number of feasible solutions and a chromosome normalization.

Scalability by implementing parallel computing in a cloud environment (Chapter 5)

This work explains why scalability is essential in the MINP, how to implement parallel computing in the designed genetic algorithm and how to leverage the use of a cloud environment. The results for both of these computational techniques are also included in this thesis.

Uncertainty simulation (Chapter 6)

Section 6.1 introduces a uncertainty simulation procedure that uses possible demand vectors to evaluate a solution based on a possible profit generated instead of using objective functions to perform this evaluation. This simulation offers an unbiased way of evaluating solutions.

The non-technical or industrial contributions are:

Easy-to-understand solution for real Multi-Item Newsvendor Problems

During the development this thesis, there was a constant concern regarding the performance, but also the framework explainability to non-expertise people. Given the feedback received by the close contact with the industry, the author believes the decision-making agents easily understand this solution increasing its value.

1.3 Outline

This thesis is organized as follows:

Chapter 2 - Classical Multi-Item Newsvendor Problem

This chapter starts by presenting the state of the art of classical approaches to the MINP. Next, section 2.2 introduces the classical formulation. Section 2.3 presents two case studies with different types of demand probability density functions.

Chapter 3 - Fuzzy Multi-Item Newsvendor Problem

From this chapter, the reader can familiarized itself with the state of the art of Fuzzy approaches to the Newsvendor problem (section 3.1) and the main concepts of the Fuzzy theory (section 3.2). Section 3.3 explains how to extend the Fuzzy logic to multi-item problems, and section 3.4 the different decision-making policies that derive from it. Finally, section 3.5 presents the different membership function types explored.

Chapter 4- Optimization Algorithm

Chapter 4 starts by justifying the use of a genetic algorithm. After it, section 4.2 presents the main mechanisms of a GA. Section 4.3 introduces the novel mechanisms introduced by the author to suit better the MINP.

Chapter 5- Computational Performance

This chapter starts by justifying the need for computational efficiency, especially for large optimization problems. Subsequently, section 5.2 explains how and in which steps to implement parallel computing in a GA. Section 5.3 explains how to run a solution in the cloud and shows the results obtained from it.

Chapter 6- Results

The purpose of this chapter is to compare the case studies (and respective solutions) presented in the chapter 2 against the framework proposed in this thesis. Section 6.1 explains the procedure to simulate the different demand scenarios. Section 6.2 describes the overall tuning process in detail. Section 6.3 presents the most relevant results for the different combinations between case studies and decision-making policies. Section 6.3.3 discusses the results obtained.

Chapter 7- Conclusions

This chapter contains the author conclusions as well as suggestions for future improvement.

2

Classical Multi-Item Newsvendor Problem

Contents

2.1 State of the Art

Following the categorization proposed in [2], five categories define the inventory management problems: Economic Order Quantity (EOQ), Economic Production Quality (EPQ), Joint/Supply chain, Inventory Control and Newsvendor Problem. The Newsvendor problem can also be named the traveling salesman problem, the newsboy problem, or the Christmas tree problem [11]. By definition, it is an inventory management problem where the commercialized items have a short cycle life and have a highly uncertain demand. Generally, the decision intends to maximize/minimize the expected profits/costs or its possibility of happening. Several analytical approaches tackle this problem in the literature, based on different constraints or decision policies [5]. These studies tend to use past pieces of evidence to derive probability density functions.

F. Y. Edgeworth first tackled this issue in the nineteenth century [7]. His publication in the *Journal of the Royal Statistical Society* used the central limit theorem to determine the optimal cash reserves to satisfy random withdrawals from depositors [12]. However, like many others, this work only considered a single-item problem, thus a single decision variable.

It was in 1964 that Hadley and Whitin [13] first extended the Single-item Newsvendor Problem. Their innovative work used Lagrangian operators, Leibniz rules and dynamic programming to consider a Newsvendor problem with multiple items, the so called: **Multi-Item Newsvendor Problem**.

In 2019 Mengting Mu, Junlin Chen, Yu Yang and Jian Guo [8] reviewed the MINP state-of-art. Their work identifies three variations for this problem: single constraint problems, multiple constraints problems, and problems with substitutes and complementary products. For this thesis, the author placed its focus on the single constraint problems. The remaining of this section describes the most relevant work done in this area.

Nahmias and Schmidt introduced heuristic methods in 1984 [14]. These methods made it possible to: reduce the computation effort inherited by the Lagrangian multipliers; and derive algebraic equations assuming the demand is a random variable with a known probability density function.

In 1995 and 1996, Lau and Lua [15, 16] extended the work of Handley and Whitin [13]. In 1995, they studied general demand distribution, and in 1996, they considered problems with a large number of materials.

At the beginning of the second millennium, two studies appeared in this field. Vairaktarakis [17] is the author of one of these works. He considered that the demand is entirely unknown and captured only in continuous (segmentation) or discrete windows. In the same year, Moon and Silver [18] presented both a dynamic programming and a heuristic solution for two cases: complete demand distributions knowledge; and incomplete demand distributions knowledge, where only the first two demand moments are known.

In 2004 Abdel-Malek et al. [19] presented an exact solution for uniform demand probability curves alongside a Generic Iterative Method (GIM) to handle generic distributions. First, this work was applied to a small set of 6 materials assuming exponential and beta distributions, but in 2005 Layek Abdel-Maleka and Roberto Montanari [20] decided to extend their work to a more extensive set of 17 materials, assuming normal distributions. Finally, in 2009, Zhang et al. [21] used this work to develop a binary solution method for generic demand distributions. The solutions proposed by Zhang [21] to minimize the expected cost function were very close to the ones offered by Abdel-Malek et al. [20]. That fact increases the credibility of these results, making them excellent candidates to serve as benchmark solutions.

Mentions should also be made to the valuable contributions of [5, 8, 22], reviewing the work done on the Newsvendor Problem (alongside other inventory management problems) until 2019.

2.2 Classical Formulation

The classical formulation suggested in [19] uses a modified form of the original model proposed in 1964 by [13]. This form minimizes the expected cost function, being this minimization equivalent to maximize an ”expected profit” function [19]. Also, the original model used ”the salvage value of the leftover items instead of the environmental disposal cost”. These changes have no mathematical impact. Equation 2.1 represents the model described.

$$Min. \quad E = \sum_{i=1}^{N} [c_i x_i + h_i \int_0^{x_i} (x_i - D_i) f_i(D_i) dD_i + v_i \int_{x_i}^{\infty} (D_i - x_i) f_i(D_i) dD_i], \tag{2.1}$$

Subject to

$$\sum_{i=1}^{N} c_i x_i \leq B_G \tag{2.2}$$

Where:

- *N*: Total number of items
- *i*: Item index
- v_i: Cost of revenue loss per unit of item *i*
- h_i: Cost incurred per item *i* for leftover at the end of the specific period
- c_i: Cost per unit of item *i*

- x_i: Ordering quantity of item *i* (decision variable)
- D_i: Random demand of item *i*
- $f_i(D_i)$: Demand probability density function of item *i*
- E_i: Expected cost function of item *i*
- E: Total expected cost function
- B_G: Budget available

2.3 Case Studies

The author selected two works out of the ones presented in section 2.1, to serve as benchmarks for different case studies. The selected works are [19, 20]. In the author opinion, these works are simple to understand, objective and have good comparisons to other works ([4, 21]).

Although considering different scenarios (demand curves, budget constraints), both studies use the same solution framework (with minor variations), a GIM. The remaining of this section contains two subsections dedicated to describing the case studies and proposed solutions of these works .

2.3.1 Exponential Demand Distribution

The first scenario is the one proposed in [19]. Here, the **item demand is exponentially distributed**. Equations 2.3 and 2.4 respectively define the probability density function and cumulative distribution function of an exponential distribution with a mean value μ.

$$f(x;\mu) = \begin{cases} 0, & x < 0 \\ \frac{1}{\mu}e^{-\frac{x}{\mu}}, & x \geq \mu \end{cases} \tag{2.3}$$

$$F(x;\mu) = \begin{cases} 0, & x < 0 \\ 1 - e^{-\frac{x}{\mu}}, & x \geq \mu \end{cases} \tag{2.4}$$

In figure 2.1, it is possible to see the exponential cumulative distribution function with a mean value of 50.

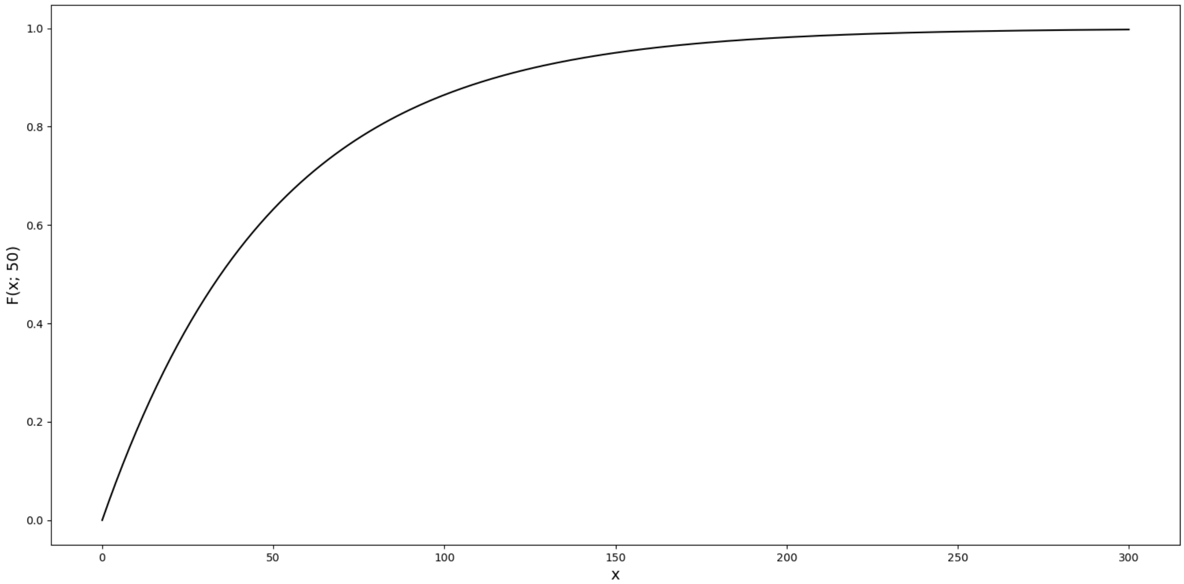

Figure 2.1: Exponential cumulative distribution function ($\mu = 50$)

The authors of [19] studied this demand-type considering a problem with six items and a budget of 3500 Currency Unit (CU). Table 2.1 presents the material data proposed:

Table 2.1: Exponential Distribution: Relevant Data

Item	v_i (CU/item)	h_i (CU/item)	c_i (CU/item)	μ_i (item)
1	7	1	4	200
2	12	2	8	225
3	30	4	20	112,5
4	30	4	10	100
5	40	2	13	75
6	45	5	15	30

Where:

- v_i: Cost of revenue loss per unit of item *i*
- h_i: Cost incurred per item *i*
- c_i: Cost per unit of item *i*
- μ_i: Mean value of the probabilistic distribution of item *i*

The GIM proposed in [19] obtained a solution for this problem relaxing the problem constrain, applying the Leibniz Rule and finally a Lagrangian optimization with a Lagrangian multiplier. For further detail consult [19]. Table 2.2 shows the proposed solution to optimize the expected profit.

Table 2.2: Exponential Distribution: Benchmark Solution

Item	1	2	3	4	5	6
x_i	78,41	58,16	30,06	81,74	70,91	25,29

2.3.2 Normal Demand Distribution

The second case study extracted from the literature is [20]. In this case, **normal distributions describe each item demand**. Equations 2.5 and 2.6 respectively define the probability density function and cumulative distribution function of a normal distribution with mean value μ and standard deviation σ.

$$f(x;\mu;\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{2.5}$$

$$F(x;\mu;\sigma) = \frac{1}{2}[1 + erf(\frac{x-\mu}{\sigma\sqrt{2}})] \tag{2.6}$$

Where:

$$erf(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^{x} e^{-t^2} dt \tag{2.7}$$

Figure 2.2 shows a normal cumulative distribution function with a mean value of 120 and a standard deviation of 30.

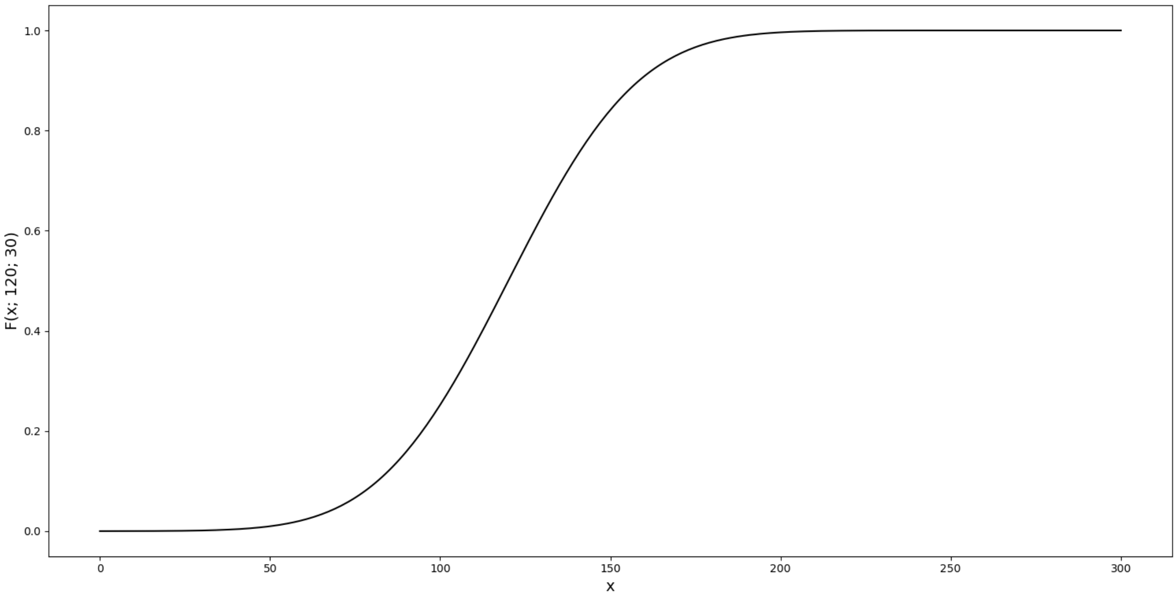

Figure 2.2: Normal cumulative distribution function ($\mu = 120$ and $\sigma = 30$)

The study of this type of distribution included 17 materials and a budget of 2500 CU. Table 2.3 contains each material specific data:

Table 2.3: Normal Distribution: Relevant Data

Item	v_i (CU/item)	h_i (CU/item)	c_i (CU/item)	μ_i (item)	σ_i (item)
1	7	1	4	102	51
2	12	2	8	73	18,3
3	30	4	19	123	30,8
4	30	4	17	95	23,8
5	40	2	23	62	15,5
6	45	5	15	129	43
7	16	1	10	69	34,5
8	21	2	10	83	41,5
9	42	3	40	120	30
10	34	5	20	89	22,3
11	20	3	10	115	38,3
12	15	5	7	91	30,3
13	10	3	4	52	17,3
14	20	3	12	76	38
15	47	2	33	66	16,5
16	35	4	21	147	36,8
17	22	1	11	104	34,7

Where:

- v_i: Cost of revenue loss per unit of item *i*
- h_i: Cost incurred per item *i*
- c_i: Cost per unit of item *i*
- μ_i: Mean value of the probabilistic distribution of item *i*
- σ_i: Standard deviation of the probabilistic distribution of item *i*

Additionally to the methodology proposed in [19], the solution framework used in [20] introduced a way of ”deleting products, in ascending order, that have low marginal utility”. The reader can consult the overall solution procedure in [20]. Table 2.4 presents the proposed solution for maximizing the expected profit considering this case study.

Table 2.4: Normal Distribution: Benchmark Solution

Item	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
x_i	0	0	0	0	0	106,86	0	14,02	0	0	15,58	42,2	34,56	0	0	0	15,23

3

Fuzzy Multi-Item Newsvendor Problem

Contents

3.1 State of the Art

Fuzzy sets theory, introduced by Zadeh in 1965 [23], is an appropriate framework to deal with uncertainties. The literature offers several approaches that use Fuzzy logic to solve Newsvendor Problems. These approaches started from being purely analytical, but studies further introduced heuristics methods.

Analytical analyses in a Fuzzy environment [10, 24–27] are very useful to specific cases where it is possible to study a limited number of items in a well isolated economic environment. Problems arise when the number of items and their co-relations increase leading to highly non-linear problems, making analytical approach inconvenient or hard to implement. In this cases, heuristics methods can be a suitable alternative.

Inspired by real-world phenomenons, the heuristic methods are algorithms that use computational power to find solutions when the classical methods cannot, due to time, complexity, or even possible nonexistence. Heuristics do not guarantee (and very often is not the case) the solution found is optimal. However, they can provide good results for highly complex optimization problems, so they are used in real-case scenarios [28].

In the literature, it is possible to identify (to the best of the author's knowledge) the paper that serves as foundation for the implementations of heuristics in a Newsvendor Problem with a Fuzzy environment, being it [4]. However, the idea of using objective functions appeared in 1996 with [10], the adoption of credibility theory concepts [29, 30] is only proposed by [4] in 2006. This work used the concepts of Possibility, Necessity and Credibility of a Fuzzy event, alongside the Excepted Value of a Fuzzy variable [31] to derive objective functions for different decision-making policies.

Finally, the work of [11] used a variety of meta-heuristic algorithms to solve a Fuzzy single-period newsvendor problem. In chapter 4 explains why choosing the genetic algorithm of [4] to the detriment of these other meta-heuristic methods.

3.2 Theoretical Concepts

3.2.1 Fuzzy Set Theory

In classic set theory, an object/element either belongs or not to a specific set. In contrast, a Fuzzy set associates membership grades to its elements. The universe of discourse consists of the universe formed by all elements of the Fuzzy set (being them continuous or discrete). By grouping the membership grades of the set's elements, it is possible to define a membership function. In other words, membership functions can assume different shapes depending on the problem context, but in its essence, they are a scale that represents ”how much element y belongs to the set X”. For example, observe figure 3.1 with the membership function of the profit of item ”i”:

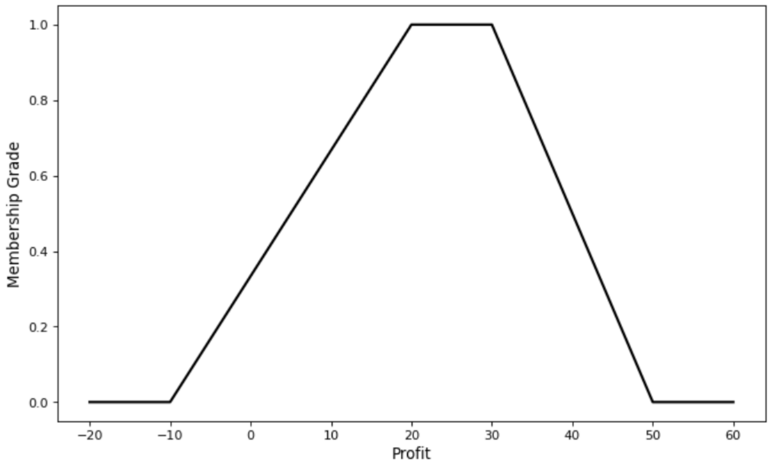

Figure 3.1: Membership Function: Profit of item ”i”

This membership function represents the several Possibility values of an item ”i” generating a given profit y. For instance, one says that ”the possibility of item ”i” generating a profit of five is equal to 0.5”. Looking at the function is possible to see that profits below -10 has a null possibility. Hence, the membership grade is zero. From -10 to 20 the Possibility starts to increase until 1, where it remains constant until 30. Finally, this membership grade decreases until it reaches zero at 50, which is the maximum profit that item ”i” can generate. Notice there are multi elements in the universe of discourse with a possibility or membership grade equal to 1, meaning the sum of all grades it is not necessarily 1 as in a probability framework.

The works [23, 32, 33] include further definitions on Fuzzy sets such as the union, interceptions and algebraic operation. However, this work does not aim to dissect these concepts, therefore, the remaining of this section will only present the most relevant definitionsto understand the proposed algorithm functionality.

3.2.2 Credibility Theory

Credibility theory was introduced by [29, 30]. [4] used some of the concepts proposed to define objective functions that describe multiple decision-making policies (section 3.4). This section helps the reader understanding these concepts by looking at their definition and a concrete example.

The Credibility theory starts by defining the terms: Possibility, Necessity and Credibility of a Fuzzy event. Formally, their definition is:

$$Pos\{\xi \geq r\} = sup\, m(u)\ ,\ u \geq r \tag{3.1}$$

$$Nec\{\xi \geq r\} = 1 - sup\, m(u)\ ,\ u \leq r \tag{3.2}$$

$$Cr\{\xi \geq r\} = \frac{1}{2}[Pos\{\xi \geq r\} + Nec\{\xi \geq r\}] \tag{3.3}$$

In plain English, the Possibility (equation 3.1) of a Fuzzy variable ξ being larger than a specified value *r* is equal to the largest membership grade m found for values larger or equal than *r*. The Necessity of a Fuzzy variable ξ being larger than a specified value *r* (equation 3.2) is the standard complement [23] (unit minus the membership grade) of the largest membership grade m found for values smaller or equal than *r*. Finally, the Credibility (equation 3.3) is the arithmetical mean between the Possibility and the Necessity.

With this in mind, it is also possible to define the **expected value of a Fuzzy variable** ξ [4]:

$$E[\xi] = \int_0^{\infty} Cr\{\xi \geq r\}\, dr - \int_{-\infty}^{0} Cr\{\xi \leq r\}\, dr \tag{3.4}$$

Example :

Consider the Fuzzy event $\{\xi \geq 10\}$ ("item "i" generates a profit higher than 10"), in the example presented in figure 3.1.

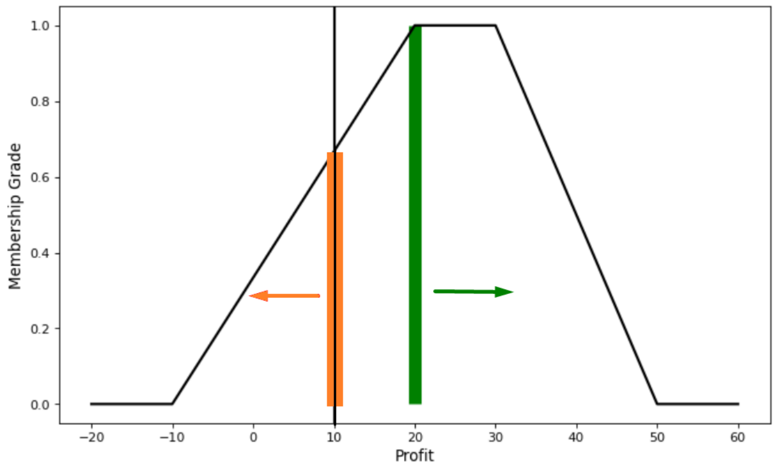

Figure 3.2: Credibility Assessment: Item "i" generating a profit bigger than 10

Using figure 3.2, it is possible to conclude that:

1. The highest membership grade found for profit values bigger than 10 (alongside green arrow) is 1, hence the **Possibility of "item "i" generating a profit bigger than 10" is 1.**

2. The biggest membership grade found for profit values smaller or equal than 10 (alongside orange

arrow) is $\frac{10-(-10)}{20-(-10)} = \frac{2}{3}$, therefore the **necessity of ”item ”i” generating a profit bigger than 10” is $\mathbf{1 - \frac{2}{3} = \frac{1}{3}}$.**

3. The average between the possibility and necessity of ”item X generating a profit bigger than 10” is $\frac{1+\frac{1}{3}}{2} = \frac{2}{3}$, so the **credibility of ”item ”i” generating a profit bigger than 10” is $\mathbf{\frac{2}{3}}$.**

For generic trapezoidal MF defined by (a,b,c,d) and the Fuzzy event $\{\xi \geq x\}$, the definitions are:

$$Pos\{\xi \geq x\} \begin{cases} 1, & x \leq c \\ \frac{d-x}{d-c}, & c < x \leq d \\ 0, & x > d \end{cases} \tag{3.5}$$

$$Nec\{\xi \geq x\} \begin{cases} 0, & x \geq b \\ \frac{b-x}{b-a}, & a \leq x < b \\ 1, & \text{otherwise} \end{cases} \tag{3.6}$$

$$Cr\{\xi \geq x\} = \begin{cases} 1, & x < a \\ \frac{2b-a-x}{2(b-a)}, & a \leq x < b \\ \frac{1}{2}, & b \leq x < c \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

Additionally, by calculating, alongside the complete universe of discourse, the Credibility of ”i” generating a **profit higher than ξ for positive values of ξ** and the Credibility of item ”i” generating a **profit smaller than ξ for negative values of ξ**, it is possible to access the **expected profit of item ”i”**. Figure 3.3 illustrates this analysis.

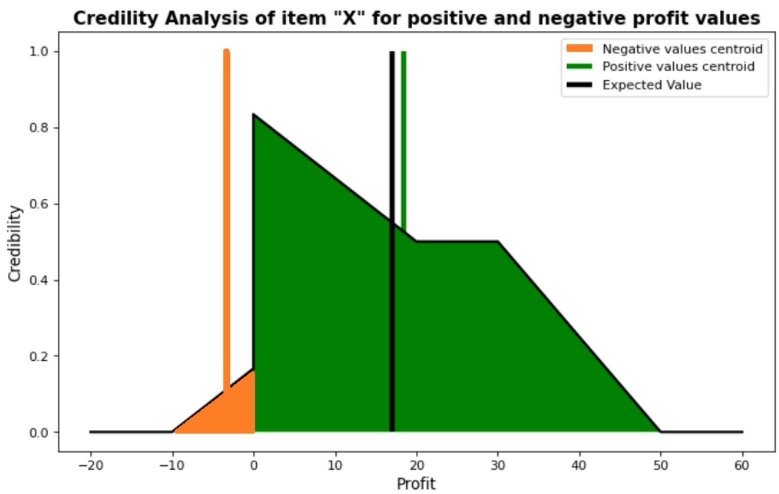

Figure 3.3: Expected Profit for item ”X”

From figure 3.3, the expected value can be perceived as a weighted average between the credibility of ξ assuming positive values minus the credibility of ξ assuming negatives values.

3.3 Multi-Item Fuzzy Extension

The examples of section 3.2.2 demonstrated that if the membership function of a continuous Fuzzy variable is well defined, it is possible to assess its Credibility of being larger or equal to a value *r*. However, problems arise when dealing with **multi-item problems** because each item demand has its membership function. That is problematic because only with all possible grades is it possible to access the membership functions. Besides, even if that was feasible (which already requires a tremendous computational effort for problems with a high number of items), all possible demand values must be combined to fully represent the universe of discourse. Finally there, a membership grade could be associated with each combination, based on an interception rule (further see section 3.3.1). As a matter of curiosity, following the reasoning explained, a simple multi-item problem with ten items, where each item has 50 possible demand values (and a membership grade associated with each value) would result in $50^{10} \ = 10^{16}$ (100 million of billions) combinations.

[4] proposed a solution for this problem by the name Fuzzy Simulation. This solution generates a high enough number of random demand combinations (which from now on will be called **demand vectors**) as a representation of the complete problem's universe of discourse. Subsequently, **Credibility** estimation of a solution satisfying a Fuzzy event and its **Expected Value** can follow procedures similar to those explained in section 3.2.2.

The remaining of this section contains four subsections to describe in detail the **multi-item Fuzzy extension**. The first subsection explains how to estimate the membership grade of a single demand vector. The second demonstrates how to use these membership grades to estimate the Possibility and Necessity of a multi-item solution satisfying a Fuzzy event. Section 3.3.3 presents the overall framework to estimate the Credibility of a multi-item solution and the enhancements introduced by the author comparing to previous works. And subsection 3.3.4 explains how to use different Credibility samples to estimate the expected profit of a multi-item solution.

3.3.1 Membership Grade Estimation of a Vector

In a MINP, demand vectors contain the proposed quantities for each item. Since each item has its unique demand membership function, it is fundamental to find a way of estimating the grade of a demand vector. This is the purpose of a **conjunctive operator**. This work studied two conjunctive operators, being the minimum (introduced in [4]) and the mean.

Let us assume $u_k = (u_{1k}, u_{2k}, ..., u_{nk})$ is a demand vector of n elements, $m(u_k)$ its estimated membership grade and $m(u_{nk})$ the membership grade associated with each item proposed ordering quantity.

The definition of the **minimum conjunctive operator** is:

$$m(u_k) = m(u_{1k}) \cap m(u_{2k}) \cap ... \cap m(u_{nk}) = min(m(u_{1k}), m(u_{2k}), ...m(u_{nk})) \tag{3.8}$$

And the definition of **mean conjunctive operator** is:

$$m(u_k) = m(u_{1k}) \cap m(u_{2k}) \cap ... \cap m(u_{nk}) = \frac{m(u_{1k}) + m(u_{2k}) + ... + m(u_{nk})}{n} \tag{3.9}$$

3.3.2 Possibility and Necessity Estimation

The Possibility corresponds to the highest membership grade found in elements that satisfy a Fuzzy event. Looking at figure 3.2, the Possibility of "item "i" generating a profit higher than 10" is 1 because there are elements that satisfy the Fuzzy event (in this case, generating a profit higher than 10) with a membership grade of 1.

On the other hand, the Necessity requires to find the element with the highest membership grade that do not belong to the Fuzzy event. This allows to access the complement, or in other words, the difference between the unit and the grade itself. Looking at figure 3.2, the Necessity is $\frac{2}{3}$, since the highest membership grade found for elements that do not satisfy the Fuzzy event (generating a profit higher than 10) is $\frac{1}{3}$, then complement is $\frac{2}{3} = 1 - \frac{1}{3}$.

Following this reasoning, the Possibility and Necessity estimation of multi-item solutions can **use a high enough number of random demand vectors**. This estimation requires to find, out of set of demand vectors:

1. The vector with the highest membership grade that **satisfies** the Fuzzy event.

2. The vector with the highest membership grade that **does not satisfy** the Fuzzy event.

For the MINP, the definition of estimated Possibility and Necessity of a solution generating a profit higher than F_0 is:

$$\widetilde{Pos}\{F(x, u_k) \geq F_0\} = \max_{1 \leq k \leq N} \{m(u_k) | F(x, u_k) \geq F_0\} \tag{3.10}$$

$$\widetilde{Nec}\{F(x, u_k) \geq F_0\} = 1 - \max_{1 \leq k \leq N} \{m(u_k) | F(x, u_k) \leq F_0\} \tag{3.11}$$

Where $F(x, u_k)$ is the profit function and *N* the total number of random demand vectors. Recalling the definition proposed in equation 3.3, the estimated credibility is then:

$$\widetilde{Cr}\{F(x, u_k) \geq F_0\} = \frac{1}{2}[\widetilde{Pos}\{F(x, u_k) \geq F_0\} + \widetilde{Nec}\{F(x, u_k) \geq F_0\}] \tag{3.12}$$

3.3.3 Proposed Credibility Estimation Framework

Figure 3.4 illustrates the proposed procedure to assess the Credibility of a solution generating a profit higher than a target.

Framework Enhancements:

Besides the properties mentioned in previous sections, this framework contains enhancements to improve performance and running time. These enhancements are:

1. **Early solution Rejection:** Solutions that do not respect the constraints (in this case, over budget solutions) discarded.

2. **Identification of inexplicable results:** Sometimes, for low credibility solutions, it is possible to estimate a Necessity value higher than the Possibility. In those cases, this feature automatically assign a credibility value of zero

3. **Adjustable α_{cut}:** In the generation of demand vectors, it is only considered quantities that have a membership grade higher than the α_{cut}. This threshold updates under two conditions:

   (a) **Threshold quantiles:** If after generating N random demand vectors the possibility and necessity did not overpass the next threshold level, the threshold is updated for the next threshold level. This N value is the Number of Random Demand Vectors per Grade Quantile and per Material (NQM). Additionally, the levels are defined by 10% quantiles, being:

   $$\alpha_{cut}\ levels = [1e-5,\ 0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9] \tag{3.13}$$

   (b) **Threshold minimum update:** The threshold should always be equal or greater than the minimum value between the highest membership grades found for both Possibility and Necessity.

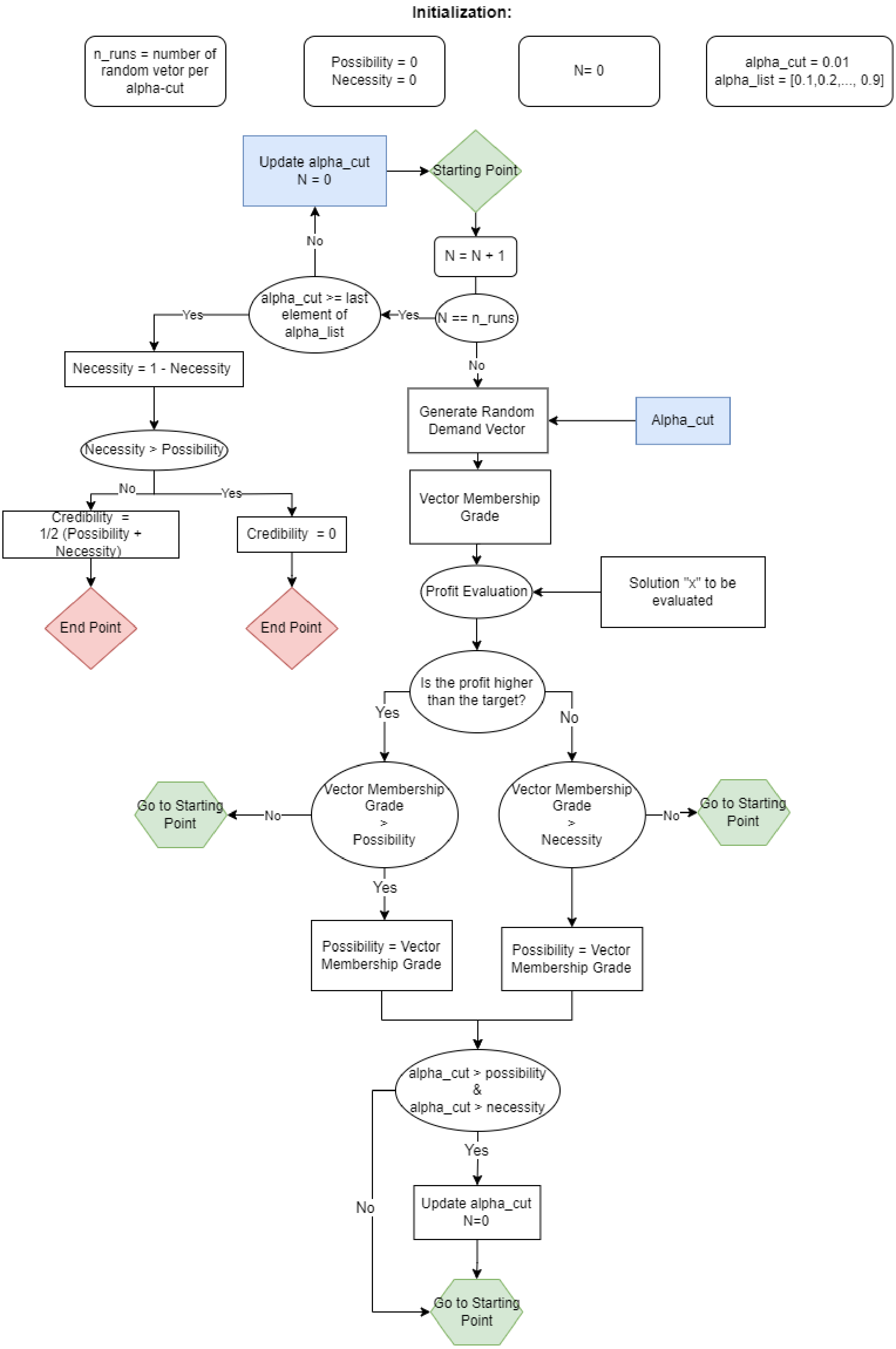

Figure 3.4: Credibility Estimation for Fixed Profit Target

3.3.4 Expected Value Estimation

This section intends to extend the definition of the **expected value of a Fuzzy variable** ξ (equation 3.4) to multi-item problems. The rationale is similar to previous section 3.3.3 but instead of the Credibility, the estimation focus on the expected value.

As equation 3.4 suggests, the expected value can be interpreted as the difference between a weighted average of Credibility values for positive profits minus a weighted average for negative profits, being the weights the corresponding credibility values. Since it is impossible for the computer to access an infinity number of credibility values, a finite number of samples must be calculated. These are the **Credibility samples**. Assuming the set of chosen profit values for the Credibility sampling is given by $r = (r_1, r_2 ... r_{N_{cr}})$, where $r_1 < r_2 < ... < r_{N_{cr}}$, equations 3.14, 3.15 and 3.16 describe the steps to estimate the expected profit:

$$E_1 = -\sum \widetilde{Cr}\{F(x, u_k) \leq r_i\}, \quad if\ r_i < 0 \tag{3.14}$$

$$E_2 = E_1 + \sum \widetilde{Cr}\{F(x, u_k) \geq r_i\}, \quad if\ r_i \geq 0 \tag{3.15}$$

$$E = E_2 \times \frac{(r_{N_{cr}} - r_1)}{N_{cr}} + max(0, r_1) + min(0, r_{N_{cr}}) \tag{3.16}$$

Where:

- N_{cr}: Total number of Credibility samples.

This work considers the number of credibility samples as a variable that must be studied being a crucial parameter to the algorithm performance (further see section 6.2.2). Contrary, the definition of the interval where to extract these samples is constant and based on the solution minimum and maximum possible profits. Using the same notion as in equation 2.1, the definition of minimum and maximum possible profits is:

$$P_{min} = -\sum_{i=1}^{N} c_i x_i = -B_G \tag{3.17}$$

$$P_{max} = \sum_{i=1}^{N} (v_i - c_i) x_i \tag{3.18}$$

$$P_{int} = [P_{min}, P_{max}] \tag{3.19}$$

Regarding the distribution between samples, it must ensure all samples as equally spaced between each other. As example, considering 3 credibility samples, with $c_i = (c_1, c_2, c_3) = (1, 2, 1)$ and $h_i = (h_1, h_2.h_3) = (5, 3, 2)$ (revisit the problem notation if needed in section 2.2), the expected profit calculation for the solution $x_i = (20, 30, 15)$ will consider the following interval:

$$P_{int} = [-(1 \times 20 + 2 \times 30 + 1 \times 15); ((5 - 1) \times 20 + (3 - 2) \times 30 + (2 - 1) \times 15)] = [-95; 140]$$

Then, the 3 equally distributed profit values used for the credibility sampling will be $\{-95; 22, 5; 140\}$. Finally, the estimated expected profit will be given by:

$$E = (-\widetilde{Cr}\{F(x, u_k) \leq -95\} + \widetilde{Cr}\{F(x, u_k) \geq 22.5\} + \widetilde{Cr}\{F(x, u_k) \geq 140\}) \times \frac{(140 - (-95))}{3} + 0 + 0$$

3.4 Fuzzy Decision-Making Policies

Section 3.3 shows that it is possible to estimate the Credibility and expected value of a Fuzzy event in a MINP. These properties allow to implement three **decision-making policies**:

1. Maximize the expected profit
2. Maximize Credibility with minimum profit target
3. Maximize profit with minimum Credibility target

The remaining of this section contains three sub-sections to explain how to implement these decision policies. Note that some notions of a GA such us population, fitness are slightly mentioned in this section but they will be explain in detail in chapter 4.

3.4.1 Expected Profit Maximization

This policy aims to find the solution that has the highest expected profit. These are the proposed steps to solve this optimization problem:

1. Define the number of credibility samples (and general algorithm's parameters)
2. Generate an initial population
3. Evaluate population fitness by following this procedure with each individual:
    (a) Define interval of interest by applying equation 3.17, 3.18 and 3.19

(b) Extract the profit values for each credibility sample, taking into account the total number of samples and that they must be equally distributed with the interval of interest.

(c) Compute the Credibility for each profit values

(d) Sequentially use equations 3.14, 3.15 and 3.16 to access the solution's expected profit. This value is the fitness value.

4. Apply reproductive methods to create the next generation, maintaining the best individual
5. Repeat steps 3 and 4 until it reaches the total number of generations
6. Select solution with the highest fitness

3.4.2 Credibility Maximization with Profit Target

This decision-making policy aims to find the solution that offers the highest Credibility of generating a profit higher than a given target. These are the proposed steps to solve this optimization problem:

1. Define the profit target (and general algorithm's parameters)
2. Generate an initial population
3. Evaluate population fitness by estimating the Credibility (section 3.3.3) of each solution generate profits higher than the target
4. Apply reproductive methods to create the next generation, maintaining the best individual
5. Repeat steps 3 and 4 until it reaches the total number of generations
6. Select solution with the highest fitness

3.4.3 Profit Maximization with Credibility Target

This policy aims to find the solution with the highest profit while ensuring a given credibility level. This policy can be implemented by:

1. Define the credibility target and profit reduction step (and general algorithm's parameters)
2. Generate an initial population
3. Evaluate population fitness by following this procedure with each individual:

   (a) Set profit target as maximum possible target (equation 3.18)

   (b) Estimate credibility for current profit target using the procedure on section 3.3.3

(c) If Credibility is equal or higher than the credibility target, the solution fitness is the current profit target. If not, repeat step 3b subtracting the profit reduction step to the profit target.

4. Apply reproductive methods to create the next generation, maintaining the best individual

5. Repeat steps 3 and 4 until it reaches the total number of generations

6. Select solution with the highest fitness

3.5 Membership Function Types

To assess the full potential of the proposed solution, while comparing it with classical approaches, the MF must represent the probability distributions accurately. This section presents the three types of MF studied and suggests parameterizations to help the MF tuning.

3.5.1 Trapezoidal Membership Functions

A trapezoidal membership function is defined by four parameters (a, b, c, d) [34]. These four parameters define five different line segments alongside the complete universe of discourse. A succinct definition of a trapezoidal MF is:

$$m(x; a, b, c, d) = max(min(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}), 0) \tag{3.20}$$

Figure 3.5 presents a trapezoidal membership function with parameters (20, 30, 60, 80):

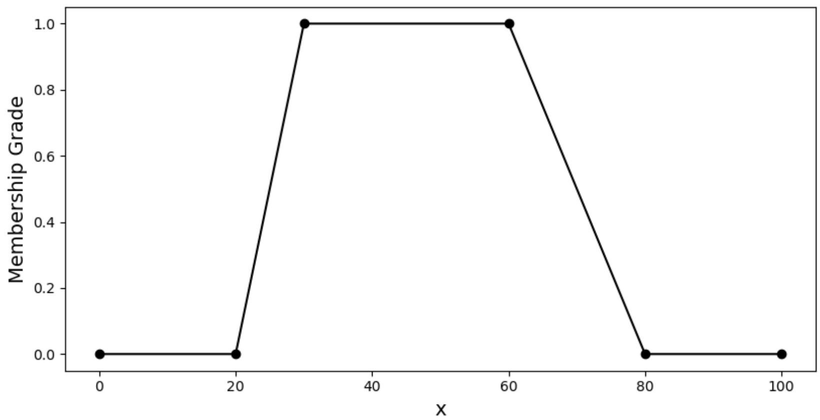

Figure 3.5: Trapezoidal MF with parameters (20,30,60,80).

Trapezoidal Membership Function Parametrization

Before analyzing the parameterization variables, it is beneficial to go over the following definitions presented in [34]:

Definition 3.1 (Support)**.** The **support** of a Fuzzy set A is the set of all points x in X that satisfies $m_A(x) > 0$:

$$support(A) = \{x|m_A(x) > 0\} \tag{3.21}$$

Definition 3.2 (Core)**.** The **core** of a Fuzzy set A is the set of all points x in X such that $m_A(x) = 1$:

$$core(A) = \{x|m_A(x) = 1\} \tag{3.22}$$

These two definitions originated the following parameterization variables:

- Support Coefficient (SC)
- Core Coefficient (CC)

Exponential Distribution: Trapezoidal Membership Function Parametrization

When representing exponential distributions (case study in section 2.3.1), the parameterization uses initial (a_i, b_i, c_i, d_i) points. These points are the ones suggested in [4] when studying the same case study.

$$a = max(0, a_i - a_i \times SC) \tag{3.23}$$

$$b = \frac{b_i + c_i}{2} - (\frac{b_i + c_i}{2} - b_i) \times CC \tag{3.24}$$

$$c = \frac{b_i + c_i}{2} - (\frac{b_i + c_i}{2} - c_i) \times CC \tag{3.25}$$

$$d = d_i + d_i \times SC \tag{3.26}$$

In section 6.2.4, figure 6.9, there is an example of how this parameterization changes a trapezoidal MF during the tuning process.

Normal Distribution: Trapezoidal Membership Function Parametrization

When representing normal distributions (case study in section 2.3.2) the parameterization uses the mean value μ and a standard deviation σ:

$$a = \mu - \sigma \times (CC + SC) \tag{3.27}$$

$$b = \mu - \sigma \times CC \tag{3.28}$$

$$c = \mu + \sigma \times CC \tag{3.29}$$

$$d = \mu + \sigma \times (CC + SC) \tag{3.30}$$

3.5.2 Exponential Membership Functions

Both exponential and normal distributions have a mean value μ, so exponential MF can have the same parameterization for both case studies. The **decay ratio *d*** is the parameterization variable, and it goes as follow:

$$m(x; d) = \begin{cases} e^{d \times \frac{x-\mu}{\mu}}, & x \leq \mu \\ e^{-d \times \frac{x-\mu}{\mu}}, & x > \mu \end{cases} \tag{3.31}$$

Figure 6.10 in section 6.2.4 contains the exponential MF with a six decay ratio used to represent the demand of material 1 in the normally distributed case study (section 2.3.2).

3.5.3 Probability Mapping

The final way proposed to represent a probabilistic distribution in a Fuzzy environment is by using the mapping presented by [35] and summarized by [10]. The mapping consists of the following steps:

1. **Compute the probability of each element:**

   Apply the following operations to a finite number elements:

$$p(n) = \begin{cases} \int_{n-0,5}^{n+0,5} f(x)dx, & n \in \mathbb{N}\backslash\{0\} \\ \int_{n}^{n+0,5} f(x)dx, & n = 0 \end{cases} \tag{3.32}$$

2. **Sort elements in descending order according to probability:**

$$\Omega = \{w_1, w_2, .., w_I\}, \quad where \quad p(w_1) \geq p(w_2) \geq ... \geq p(w_I) \tag{3.33}$$

3. **Perform the mapping using the following expressions:**

$$m(w_1) = \sum_{k=1}^{I} p(w_k) \simeq 1 \tag{3.34}$$

$$m(w_i) = i.p(w_i) + \sum_{k=i+1}^{I} p(w_k), \quad i = 2, ...I - 1 \tag{3.35}$$

$$m(w_I) = I.p(w_I) \tag{3.36}$$

4

Optimization Algorithm

Contents

4.1 Introduction

It is widespread to resort to meta-heuristics methods when analytical tools are incapable or too complex to solve highly non-linear problems. Meta-heuristics are nature-inspired optimization algorithms that further extend the capability of heuristic frameworks by guiding the generation and interpretability of solutions [36]. Most of the time, these modifications tend to yield better results since they provide a good trade-off between local and random searches. There are many examples of meta-heuristic methods applied to the Fuzzy newsvendor problem [11]. In the author's opinion, a meta-heuristic framework per se is not the only factor that influences the overall algorithm performance. The creativity in search procedure and the additional features added to better suit a given use case are also crucial to obtain the good results.

Figure 4.1 from [3] helps analysing the number of documents related to meta-heuristic methods published from 2011 until 2016. As figure 4.1 suggests, in the time of the study, the use of genetic algorithm overpassed any other meta-heuristic method including the Particle Swarm Optimization (PSO), Simulated Annealing (SA), Ant Colony Optimization (ACO), Immune Algorithms (IA), Tabu Search (TS). Additionally to this high popularity among the academia, the literature already presents studies that prove the effectiveness of GA in solving Fuzzy MINP ([4] [11]). Based on these reasons, a GA was the chosen algorithm to solve this optimization problem.

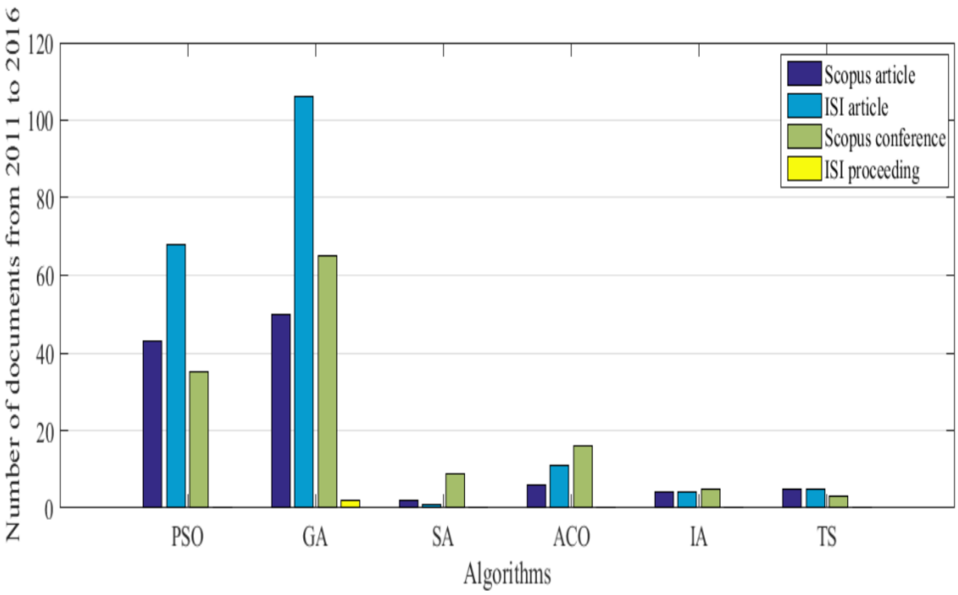

Figure 4.1: Number of Meta-heuristic algorithms from 2011 to 2016 (source [3])

The first genetic algorithm dates back to 1962 [37]. Despite its appearance in the early 1960s, this problem-solving technique (among other evolutionary algorithms) would not become popular until the '90s [38]. The main reason for that was the lack of computational power that made difficult to prove the theory in experimental testing. The reader has the work of [38] to broaden his knowledge in the history of evolutionary computation, particularly genetic algorithms.

The reaming of this chapter divides itself into two sections. The first section includes descriptions of the main GA mechanisms and the second section the novel features proposed to suit the MINP better.

4.2 Genetic Algorithm Core Concepts

4.2.1 Chromosome

A chromosome is a set of genes, and it represents a solution. Subsequently, a gene is the proposed ordering quantity for each item. Thus, an n-item newsvendor problem will have n-genes chromosomes. Figure 4.2 illustrates a chromosome of a MINP with three items.

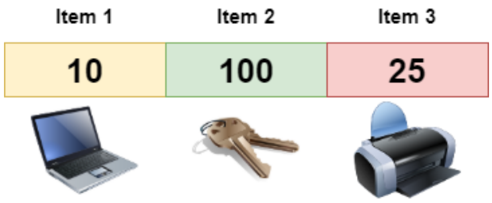

Figure 4.2: Chromosome Representation

In the author's opinion, this straightforward representation is a advantage of considering a GA for this problem. Two factors support this belief:

1. Other meta-heuristic frameworks [11] are not easy to adapt to the newsvendor problem
2. It is the foundation for clean and efficient solution generation processes. (see 4.2.2 and 4.2.3)

4.2.2 Crossover

The crossover mechanism generates new solutions, the children, from older individuals, the parents. The following sequential steps define the crossover process:

1. Randomly define the crossover length: The total number of ordering quantities to interchanged.
2. Randomly select the interchanged ordering quantities.
3. Interchange the selected ordering quantities.

Considering the example presented in figure 4.2, the crossover length can vary between 1 and 3 (inclusive). Figure 4.3 illustrates how the crossover mechanism looks like in this case.

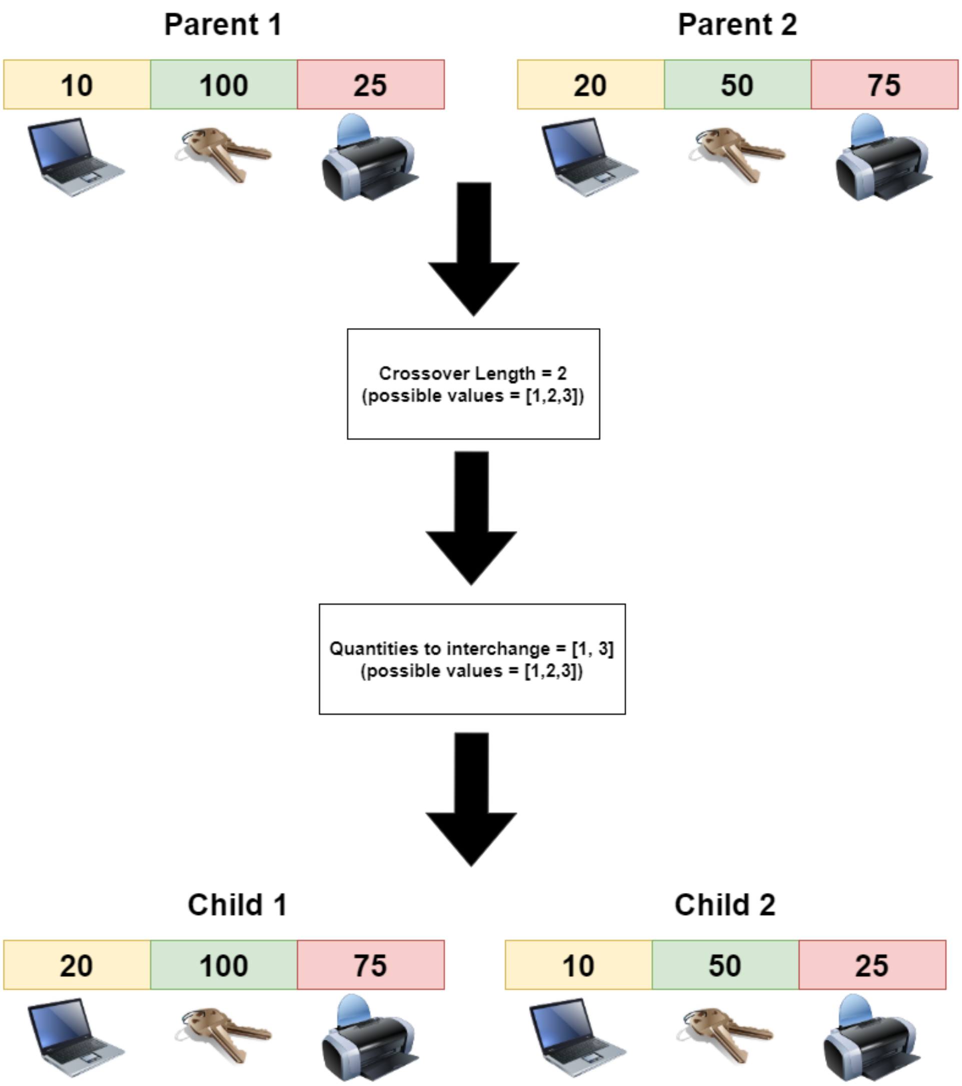

Figure 4.3: Crossover Process

The primary purpose of this reproductive method is to change previous solutions without entirely altering their configuration. This process can be considered a **local search** where the best solutions originate new solutions within the same neighbourhood.

4.2.3 Mutation

Mutation is another mechanism to generate new solutions. Contrary to the crossover, its purpose is to find new solution significantly different comparing to its ancestors. The Mutation aims to perform a **random search** exploring different solution spaces. The mutation process has the following steps:

1. Randomly define the total number of ordering quantities to mutate

2. Randomly select the ordering quantities to mutate according the total number previously defined

3. Mutate the ordering quantities

Figure 4.4 shows how to conduct this process:

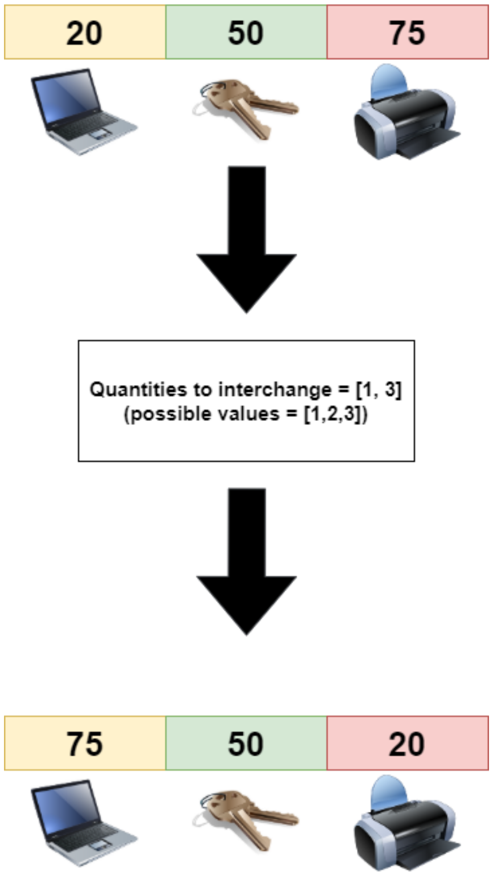

Figure 4.4: Mutation Process

4.2.4 Selection

The selection is responsible for picking the individuals that are the basis for the next generation. The selection starts by defining the tournament size through equation 4.1. Thus, the tournament size is a parameter indirectly determined by the population size and tournament coefficient. The selection process also guarantees a place in the next generation for the fittest solution (elitism).

$$T_{size} = \frac{Pop_{size}}{T_{Coef}} \tag{4.1}$$

The following sequential steps define the selection process:

1. Define T_{size}
2. Elitism
3. Random extraction
4. Tournament

Figure 4.5 illustrates the selection process:

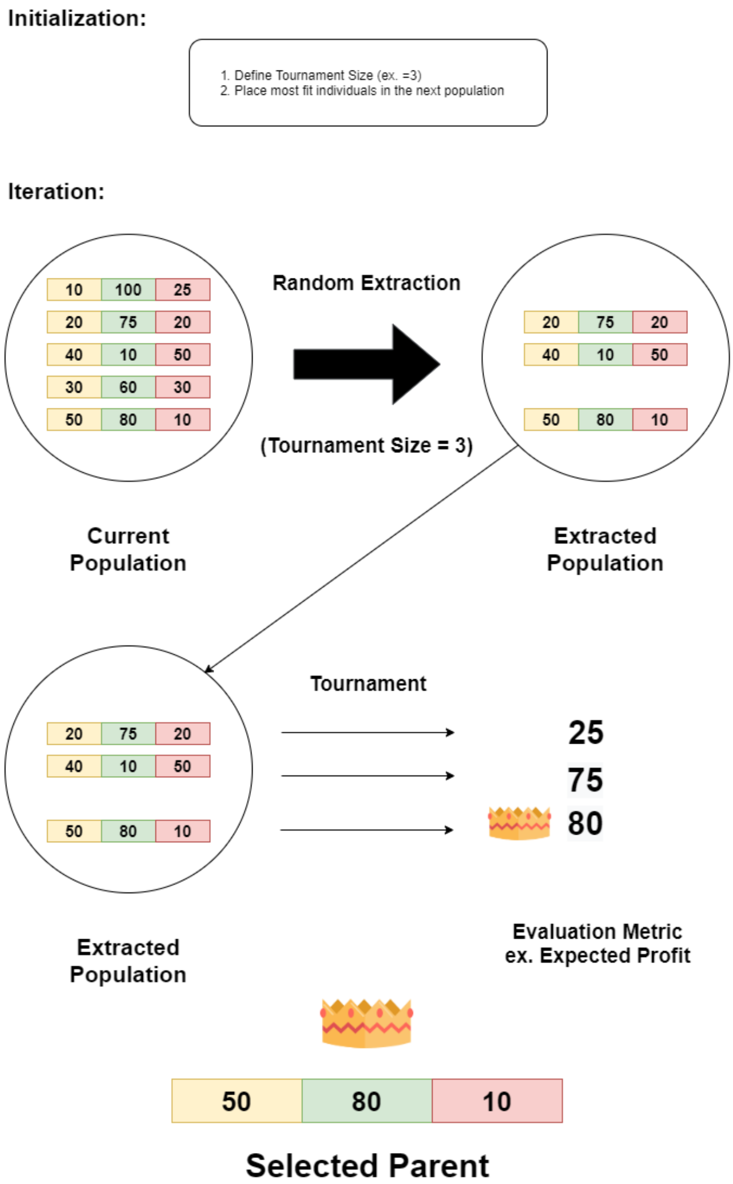

Figure 4.5: Selection Process

4.3 Problem-Specific Enhancements

4.3.1 Solution Resizing

The solution resizing feature transforms an unfeasible solution into a feasible solution without altering the relative proportions between the ordering quantities. This feature aims to increase the number of feasible solutions generated by taking into account the available profit. The following steps describe the solution resizing process:

1. Identify over-budget solutions: Solution that does not satisfy equation 2.2)
2. Compute resizing ratio: Apply equation 4.2
3. Apply resizing: Multiply all ordering quantities by the resizing ratio

The resizing ratio is given by:

$$R_{ratio} = \frac{B_G}{\sum_{i=1}^{N} c_i x_i} \tag{4.2}$$

Recalling the notation presented in section 2.2:

- x_i: Ordering quantity of item *i*
- c_i: Cost per unit of item *i*
- B_G: Budget available

As an example, let us assume with having a 3-item problem and a solution with the following parameters:

Table 4.1: Example: Solution Resizing

Variables:	**Item 1**	**Item 2**	**Item 3**
Ordering Quantity (x_i)	2	3	2
Cost (c_i) [CU/unit]	5	10	20
Available Budget (B_G)		50	

Computing the solution budget:

$$\sum_{i=1}^{N} c_i x_i = 2 \times 5 + 3 \times 10 + 2 \times 20 = 80 \text{ CU}$$

This solution is over the available budget, thus, it is a candidate for resizing. The resizing ratio is:

$$R_{ratio} = \frac{B_G}{\sum_{i=1}^{N} c_i x_i} = \frac{50}{80} = 0,625$$

By multiplying all ordering quantities x_i by the resizing ratio:

- $x_{r1} = 0,625 \times 2 = 1,25$
- $x_{r2} = 0,625 \times 3 = 1,875$
- $x_{r3} = 0,625 \times 2 = 1,25$

Considering the resized solution, the budget is:

$$\sum_{i=1}^{N} c_i x_{ri} = 1,25 \times 5 + 1,875 \times 10 + 1,25 \times 20 = 50 \quad \text{CU}$$

Note that:

$$\frac{2}{2+3+2} = \frac{1,25}{1,25+1,875+1,25}; \quad \frac{3}{7} = \frac{1,875}{4,375}; \quad \frac{2}{7} = \frac{1,25}{4,375}$$

To sum up, the solution resizing turns unfeasible solutions into a feasible ones without changing the the relative proportions between the ordering quantities.

4.3.2 Initialization with Null Values

The Initialization with Null Values uses, as initial population, chromosomes composed by null values, except in one item. From now on, these chromosomes are called null chromosomes. This feature aims to give the algorithm the capability of "understanding" which items are more profitable and naturally combine them. Without this feature, the selection of the initial population ordering quantities is purely random. Two steps define this feature:

1. Item selection: Select items that do not have a null chromosome representation
2. Generate the null chromosome with resizing (see section 4.3.1)

For instance, considering the example presented in the table 4.1, the null chromosome of item 2 is:

$$(x_1;\ x_2;\ x_3) = (0;\ 5;\ 0)$$

The ordering quantity is 5 because the available profit is 50 CU and the cost per unit 10 CU.

Note that if the population size is larger than the number of items, there will be a null chromosome for each item. Otherwise, the item selection for initialization is purely random.

4.3.3 Chromosome Normalization

The chromosome normalization suggests a solution generation procedure independent of the items' expected demand value. Very often the solutions generated present ordering quantities close to the expected demand values. In a case where expected demands are very different in terms of absolute value, the crossover or mutation suggested in section 4.2.2 and 4.2.3 can be compromised only generating solutions far away from the optimal. Therefore, chromosome normalization utilizes each ordering quantity deviation from the expected value to account for this problem. The following steps describe this process:

1. Chromosome Normalization: Apply equation 4.3
2. Crossover or Mutation: Follow sections 4.2.2 or 4.2.3
3. Chromosome Restoration: Multiply normalized ordering quantity by its respective expected value

The proposed normalization is given by:

$$x_{ni} = \frac{x_i}{\mu_i} \tag{4.3}$$

where:

- x_i: Ordering quantity of item *i*
- μ_i: Mean value of the probabilistic distribution of item *i*
- x_{ni}: Normalized ordering quantity of item *i*

Let us consider the following example to understand better the normalization proposed:

Table 4.2: Example: Chromosome Normalization

Variables:	**Item 1**	**Item 2**	**Item 3**
Ordering Quantity (x_i)	2	1200	15
Expected Demand Value (μ_i)	5	1000	20

Normalizing the ordering quantities:

- $x_{n1} = \frac{2}{5} = 0,4$
- $x_{n2} = \frac{1200}{1000} = 1,2$
- $x_{n3} = \frac{15}{20} = 0,75$

Mutating the first and the second items normalized quantities:

$$(x_{n1};\ x_{n2};\ x_{n3}) = (1,2;\ 0,4;\ 0,75)$$

Multiplying each normalized quantities by its most expected demand values:

$$(x_1;\ x_2;\ x_3) = (1,2 \times 5;\ 0,4 \times 1000;\ 0,75 \times 20) = (6;\ 400;\ 15)$$

In summary, with this chromosome normalization avoids high variations in the order quantities of each solution by taking into account their most expected demand value.

5

Computational Performance

Contents

5.1 Introduction

Scalability is crucial when designing industrial solutions. For example, it is very common for real inventory management problems to increase the number of items, thus, the number of decision variables. A problem can become impossible to solve in real-time if the computational power does not keep up with this increase in dimension. That is why computational efficiency is a critical aspect when implementing iterative optimization algorithms, such as a genetic algorithm (see [39–41]).

In an ideal scenario, computational time would be independent of the problem dimension. To achieve this, it is necessary to fully use the available computation power and scale it when needed. This flexibility is precisely what parallel computing and cloud computing can respectively offer.

Two sections compose the remaining of this chapter. The first section introduces parallel computing by explaining how this technique can reduce the computational time by half using a local machine with two Virtual Central Processing Units (vCPU). The second section studies the different types of machines offered via a cloud-based service.

5.2 Parallel Computing

Today, most desktops, laptops or data centres ship with dual-core processors, quad-core or even higher. The reason behind this popularity is energy efficiency. It is much easier to scale computing power by increasing the total processor number rather than increasing microprocessor clock frequencies [42].

An increase in computational power can only reduce running time if the implemented code can use its additional capacity. Parallel computing allows performing different tasks using multiple cores, improving computational efficiency. This technique contrasts with serial computing, where a single task allocates all available processing power.

Genetic Algorithms are excellent candidates to apply parallel computing. They exhibit several phases where they perform multiple independent tasks. On the one hand, these phases can be the fitness evaluation or creation of a population. On the other hand, each task correspond to the needed calculations per chromosome.

For the proposed solution, **the author decided to apply parallelism in the fitness evaluation procedure**, since this was the phase with the highest computational running time. For instance, using parameters that ensure a proper fitness evaluation for the second case study (further see section 6.2.5, table 6.17), the evaluation and generation of a population of 4 individuals took, in a local laptop, 65 and 0.5 seconds, respectively. Figure 5.1 shows this discrepancy between values.

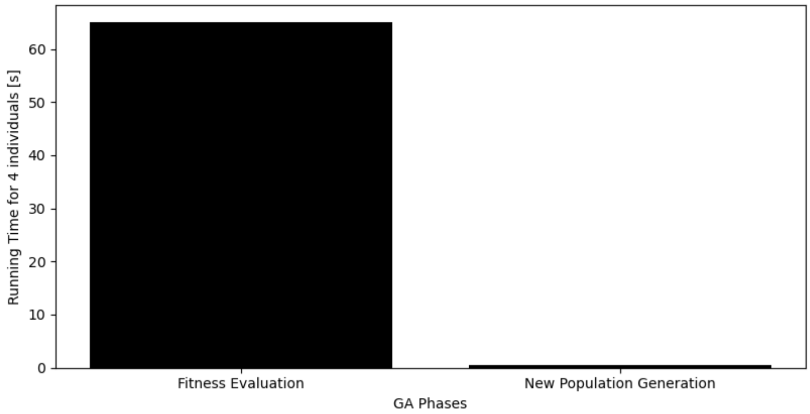

Figure 5.1: Running Time: Fitness Evaluation vs. Population Generation

Note that these values only consider four individuals per population. This is a small population size, meaning that a reasonably sized problem would have an even higher discrepancy.

Figure 5.2 uses the example previously mentioned to illustrate **how parallelism reduced the running time by nearly half** when using dual-core local machine.

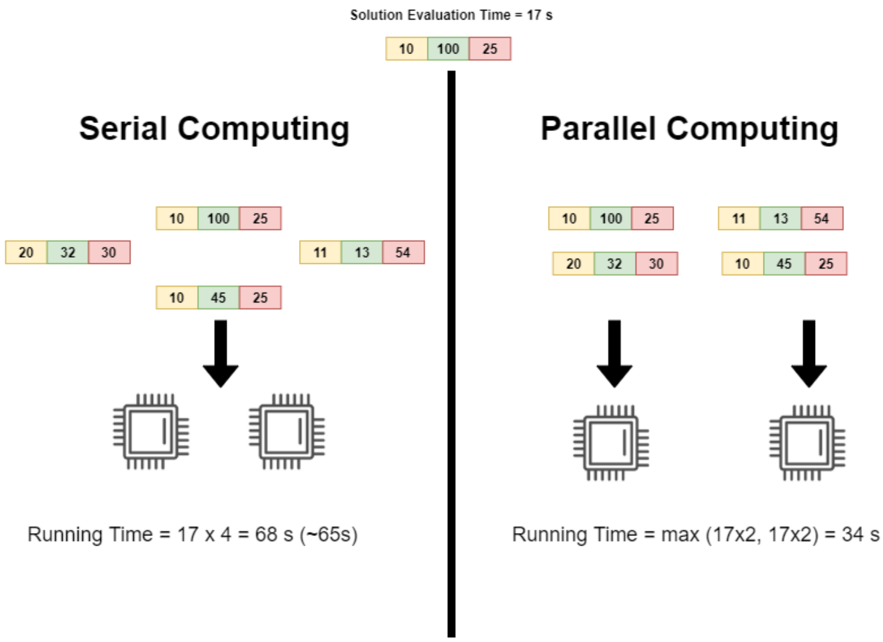

Figure 5.2: Parallel Computing Framework

In the case presented, **the running time experienced a reduction by half since the machine had two cores.** More cores would result in even more significant time reductions, being that the topic of discussion in section 5.3. Finally, notice the time fluctuations experienced (65s $\approx$ 68s). The reason behind this is the unstable environment offered by a local machine due to background tasks. A cloud environment solves this problem and also allows to have higher flexibility in terms of computational power.

5.3 Cloud Environment

Nowadays, "cloud computing" is still a buzzword in the academic and industrial worlds [43]. This term refers to the on-demand delivery of computational resources over the internet [44]. Compared to traditional processing data centres, the advantages of these services are the higher flexibility in terms of computational power and the possibility of using an isolated environment without maintaining it. Furthermore, with cloud services, it is possible to access computing power, storage and databases on an as-needed basis. Given these advantages, **the proposed framework implementation used cloud resources**. This section intends to explain, at a high level, how this implementation was possible and the results of this choice.

Every program running in the cloud has an environment previously defined by the user. This environment is called a **container**. The container has in it all the necessary packages as well as the source code to be executed. Additionally, the container receives all required variables (in the MINP, the crossover probability, population size, etc.) as environment variables. This results in an isolated and constant environment that allows running a given program for different input variables. With the container defined, it is possible to pass its image to a machine of choice. The device will then execute the tasks described in the source code. Moreover, given the well-defined environment, there is not any operating incompatibilities during execution. Figure 5.3 illustrates the overall containerization life cycle.

The ability to perform the same tasks in different machines without altering the source code or conditions is an advantage of containerization. Next, table 5.1 presents the characteristics of a set of virtual machines available in the cloud. This thesis studied these machines to find the one exhibiting the best ratio between computational power and performance.

Table 5.1: Virtual Machines Properties

	vCPU	Memory [GiB]
Machine 1	2	8
Machine 2	4	16
Machine 3	8	32
Machine 4	16	64
Machine 5	48	192
Machine 6	96	384

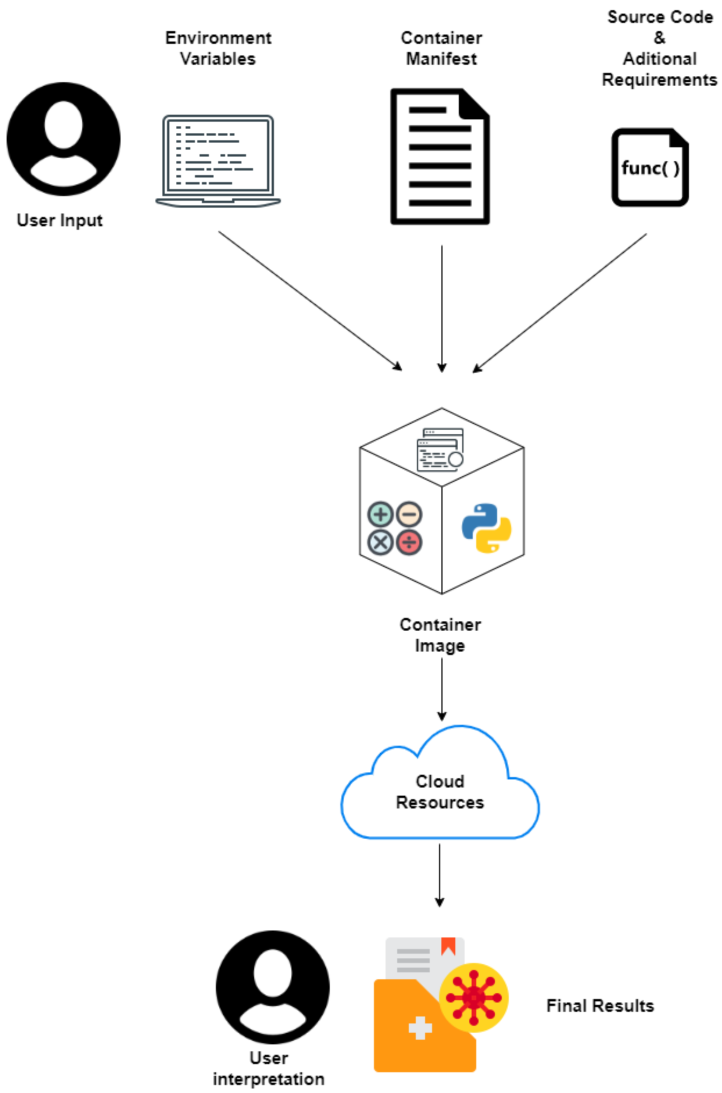

Figure 5.3: Containerization Life Cycle

The proposed algorithm ran for the case study with exponentially distributed demands (revisit section 2.3.1) considering the different machines in table 5.1 and applying parallel computing. Since the algorithm parameters change the total running time, it is important to mention that for this study the algorithm used the parameters present in table 6.14 (parameters justification in chapter 6). Figure 5.4 plots the relation between the number of vCPU and running time.

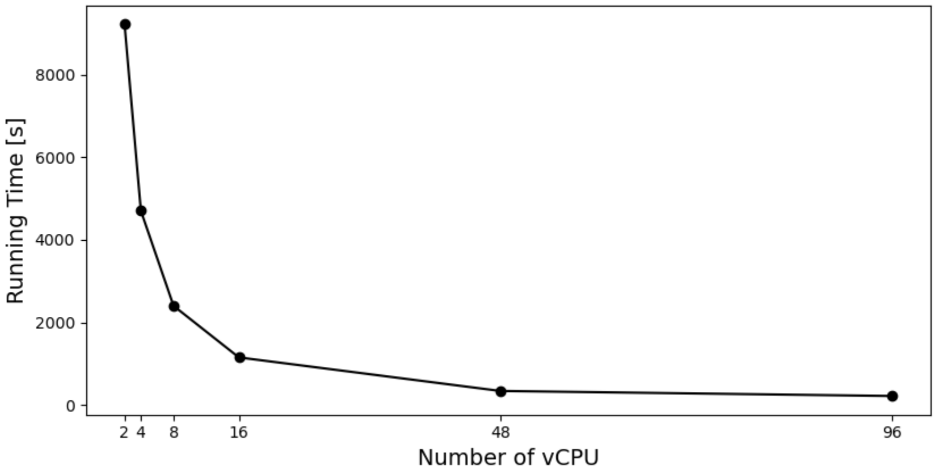

Figure 5.4: Number of vCPU per machine and Running time

Machine 1 had a running time of 9216 seconds, while machine 5, the one with the best ratio between power and performance, needed 340 seconds to complete the job. The running time reduction between these machines was 96%.

Finally, bear in mind that these results were only possible because the source code applied parallelism. Table 5.2 shows the running time with and without parallelism both using machine 5.

Table 5.2: Machine 5 Performance with and without Parallelism

Machine	**Parallelism**	**Running Time [s]**
5	Yes	340
5	No	20342

Without parallel computing, the use of powerful machines becomes irrelevant. Knowing this, table 5.2 helps concluding that the integration of parallel computing and cloud resources **reduced the computation time by 98,3%** (cloud be even more with the 96 vCPU of machine 6). This drastic reduction makes feasible problems that were almost impossible to solve due to time constraints. Moreover, note that for the normally distributed case study (revist setion 2.3.2) with **20 materials**, unstructured studies shown stable results in **less than 8 minutes**.

6

Results

Contents

This chapter uses examples from the literature to test the algorithm performance against classical solutions. These examples are the case studies presented in section 2.3.1 and 2.3.2. The reader should notice this work intention is not to compare itself against other approaches but rather to prove its suitability while delivering advantages for a wider variety of cases (further discussed in chapter 7). This chapter is organized into the following sections:

- **Section 6.1: Simulation procedure-** Explains how to simulate reality and extract metrics such as the average profit generated.
- **Section 6.2: Algorithm Tuning-** Tunes the algorithm to obtain the best possible performance.
- **Section 6.3: Case Studies Results-** Presents and discusses the most pertinent results for both case studies.

6.1 Simulation Procedure

One of the main contributions of this work is to provide a suitable evaluation framework for the proposed solutions. Until now, studies on the Fuzzy multi-item newsvendor problem (e.g. [4, 11, 45]) have been focusing on evaluating the performance of their solutions solely based on the maximization of an objective function (most often the expected value). This approach raises questions, such as: ”Is the objective function a good representation of reality?” or ”Will the solution generate the expected results in a real scenario?”.

The proposed evaluation method uses pseudo-random demand vectors to answer the questions mentioned. The demand vectors randomness depends on the materials probabilistic demand curves. The idea is to represent reality by regenerating possible demand vectors based on the items demand curves (values with higher probability tend to be selected most often). Once done, this generation will result in a sound panoply of results, where likely results will have a higher representation, but less common results are also present. This methodology makes it possible to evaluate metrics such as:

- Average profit
- Number of times the profit is higher than a given target

Figure 6.1 contains a flow chart to illustrate how to calculate the metrics mentioned above. Notice the proposed procedure generates a number of X random demand vectors and then uses this set of vectors to access the aforementioned metrics in the block named *”Compute and save metrics”*.

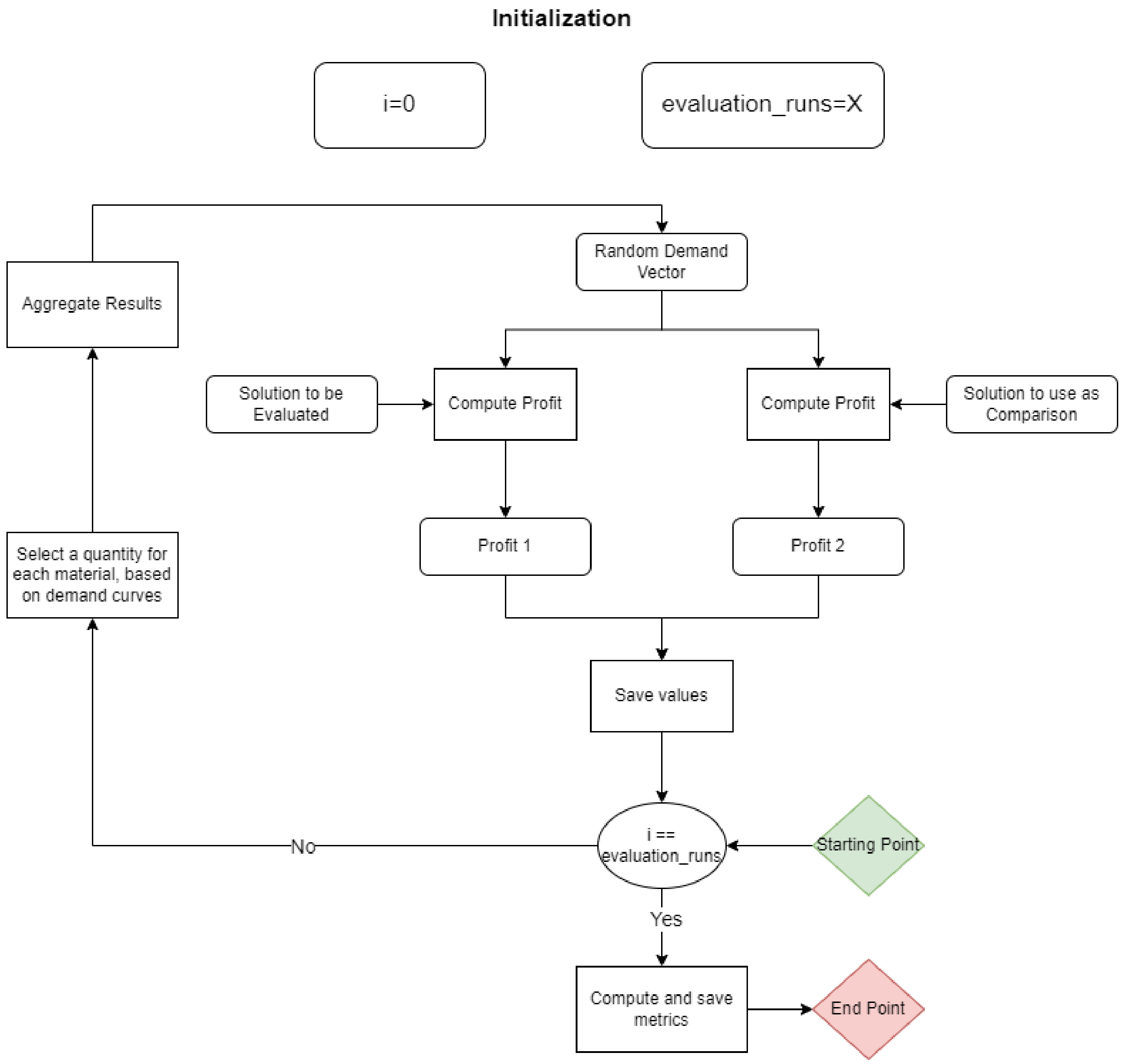

Figure 6.1: Simulation Procedure

Number of Demand Vectors

Only a high enough number of random demand vectors ensures a trustful simulation. Figures 6.2 and 6.3 illustrate, using each of the case studies benchmark solutions (tables 2.2 and 2.4, respectively), how the average profit standard deviation and computational time change with the number of random demand vectors. The number of random demand vectors was set to $100000 = 10^5$ for all simulations performed.

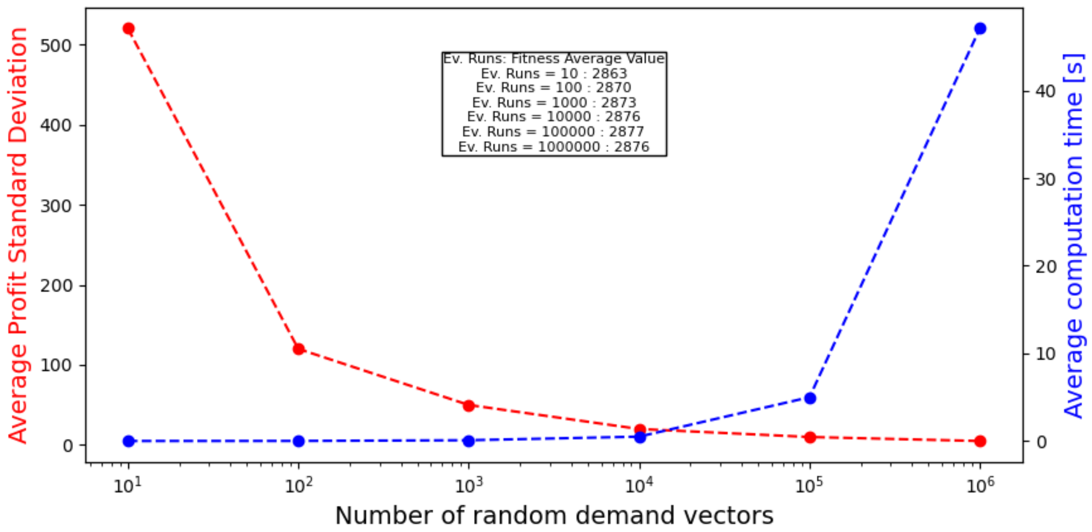

Figure 6.2: Exponential Distribution: Number of Random Demand Vectors, Profit Standard Deviation and Average Computational Time

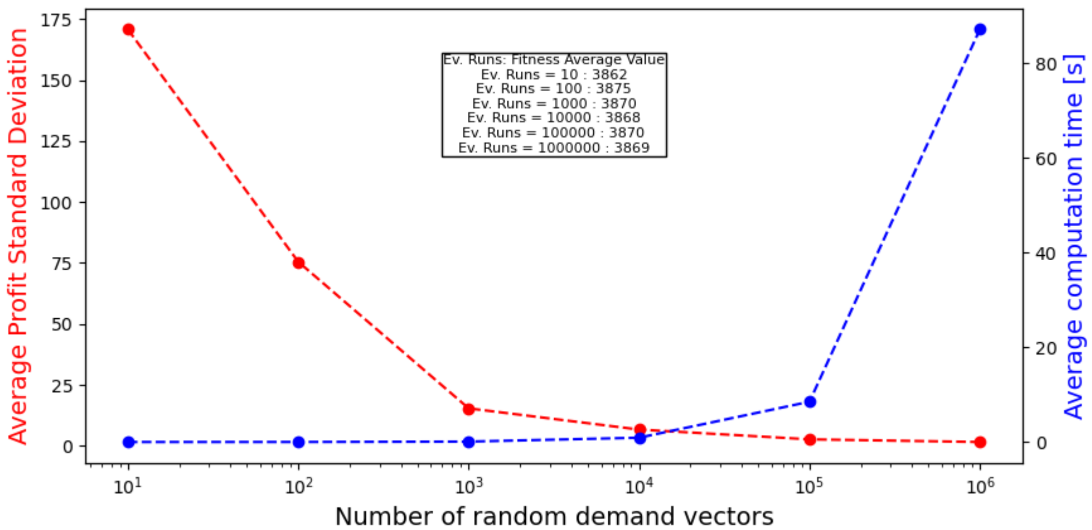

Figure 6.3: Normal Distribution: Number of Random Demand Vectors, Profit Standard Deviation and Average Computational Time

6.2 Algorithm Tuning

This section is dedicated to tune the necessary parameters to have a reliable algorithm performance. For each case study presented in section 2.3, the tuning process followed five phases:

1. **Credibility Estimation:** The credibility estimation relies on the random generation of possible demand vectors. The estimation accuracy increases with the number of random vectors generated, but so does the computational effort. This phase tunes the NQM (revisit section 3.3.3) aiming to find a good balance between accuracy and running time.
2. **Expected Profit Estimation:** As explained in section 3.3.4, the expected profit estimation uses credibility samples with different profit targets. This phase tunes the number of credibility samples to find the best trade-off between expected profit estimation stability and the computational effort.
3. **Solution Fitness Stability:** This phase tunes the population size and the number of generations to ensure the algorithm works close to its full capacities without compromising the running time.
4. **Membership Function Selection:** This section finds the best membership functions to describe the probabilistic demand curves for each case study.
5. **Other Parameters:** The tuning of the remaining solution generation/interpretation and GA-specific features goes in this section.

6.2.1 Credibility Estimation

When estimating the Credibility of a Fuzzy event, it is crucial to balance performance and computational effort. The credibility estimation relies on a random generation of possible demand vectors to have a good representation of the universe of discourse. If this random selection is not sufficiently large, the credibility estimation is compromised, leading to significantly different estimations for the same solution. The Credibility estimation stability can be tested by iteratively estimating the Credibility of a given solution and computing the standard deviation between estimations.

Additionally, bear in mind the larger the number of demand vectors generated, the longer the algorithm will take to estimate a credibility value. Since the proposed framework depends on the capability to estimate multiple credibility values, it is critical to have the fastest possible performance without compromising the accuracy, otherwise, the relevance of this solution in a industrial context is compromised.

In summary, the goal is to find the NQM (revisit section 3.3.3) value that gives the best trade-off between credibility estimation stability and computational time. There are two variables playing a role in the credibility estimation, begin them:

1. Selected profit target
2. NQM value.

Given the different number of items, each use case received its unique Credibility estimation tuning analysis. The **tuning process used the benchmark solutions** presented in tables 2.2 and 2.4. The following steps describe the course of action in credibility estimation tuning:

1. Study different profit targets with a constant NQM value. Select the profit target with the highest standard deviation.
2. Study the different NQM values for the most variant profit target.

Exponential Distribution: Credibility stability

Shao [4] selected trapezoidal MF to represent the exponential distributions. This is also the MF used in the tuning process. Table 6.1 shows its parameters:

Table 6.1: Exponential Distribution: Considered membership functions (based on [4])

Material	MF Parameters
1	[180, 190, 210, 220]
2	[210, 220, 230, 240]
3	[100, 110, 115, 125]
4	[80, 90, 110, 120]
5	[60, 70, 80, 90]
6	[20, 25, 35, 40]

To find the profit target with the highest standard deviation in the credibility estimation process, a large set of profits targets between 0 CU and 9000 CU (with 1000 CU steps) was initially used. A study conducted 20 Credibility estimations for each of these targets. The estimations used a NQM value of 10 to access the standard deviation of the estimations. The appendix figure A.1 presents the results from of this study, being the 5000 CU profit target the only target showing a not null standard deviation. Figure 6.4 shows a closer analysis between 4500 CU and 5500 CU with steps of 100 CU units steps:

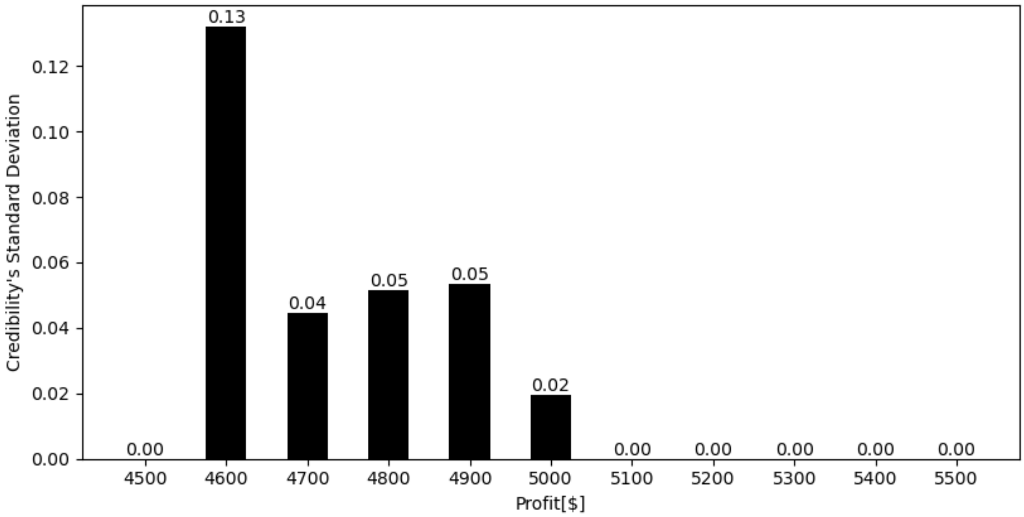

Figure 6.4: Exponential Distribution: Credibility Standard Deviation vs. Profit Targets from 4500 to 5500 CU

From figure 6.4, it is possible to see that the 4600 CU is the profit target showing a higher deviation value. Thus, **4600 CU is the target considered for the NQM tuning**. Figure 6.5 illustrates how the different NQM values change the credibility standard deviation and computational time for 20 runs when considering a 4600 CU profit target:

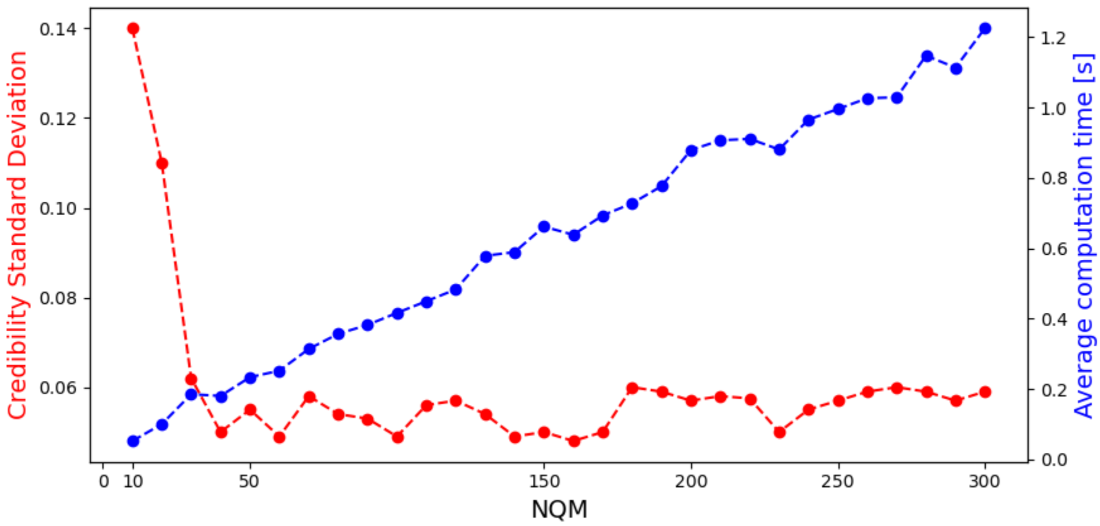

Figure 6.5: Exponential Distribution: NQM value, Credibility Standard Deviation and Average Computational Time

Based on the information carried by figure 6.5, the decision was to set the **NQM value as 50**. This choice ensures a credibility deviation of 0,054 and a credibility assessment time of 0,22 seconds. Theoretically, with these parameters, a credibility optimization for a fixed profit target with 1000 individuals per population and five generations would take around 20 min (results without considering parallelism and cloud computing).

Normal Distribution: Credibility stability

This analysis used the curves resultant from the mapping introduced in the section 3.5.3. Like in the previous section 6.2.1, the goal here is to find the profit target with the highest standard deviation, using a constant NQM value of 10. After finding this profit target, a study compares the standard deviation for different NQM values.

This is exactly the same procedure as in section 6.2.1. As a result, its results are in the appendix section A.2. The chosen **NQM value was 10**.

6.2.2 Expected Profit Estimation

As explained in section 3.3.4, the expected profit estimation relies on computing the weighted average for different profit targets, where each weight corresponds to the credibility of the Fuzzy event. Estimating values for all possible elements is a highly demanding task, thus a universe of discourse sampling is required. In other words, to ensure an equilibrium between performance and computational effort, it is necessary to find the best number of Credibility samples to estimate the expected profit of a given solution. This section shows how to tune the expected profit estimation by changing the number of Credibility samples.

Exponential Distribution: Credibility Samples

Figure 6.6 illustrates how the total number of credibility samples influence the solution expected profit stability and computational time (when considering the benchmark solution and MF) for the exponentially distributed case study:

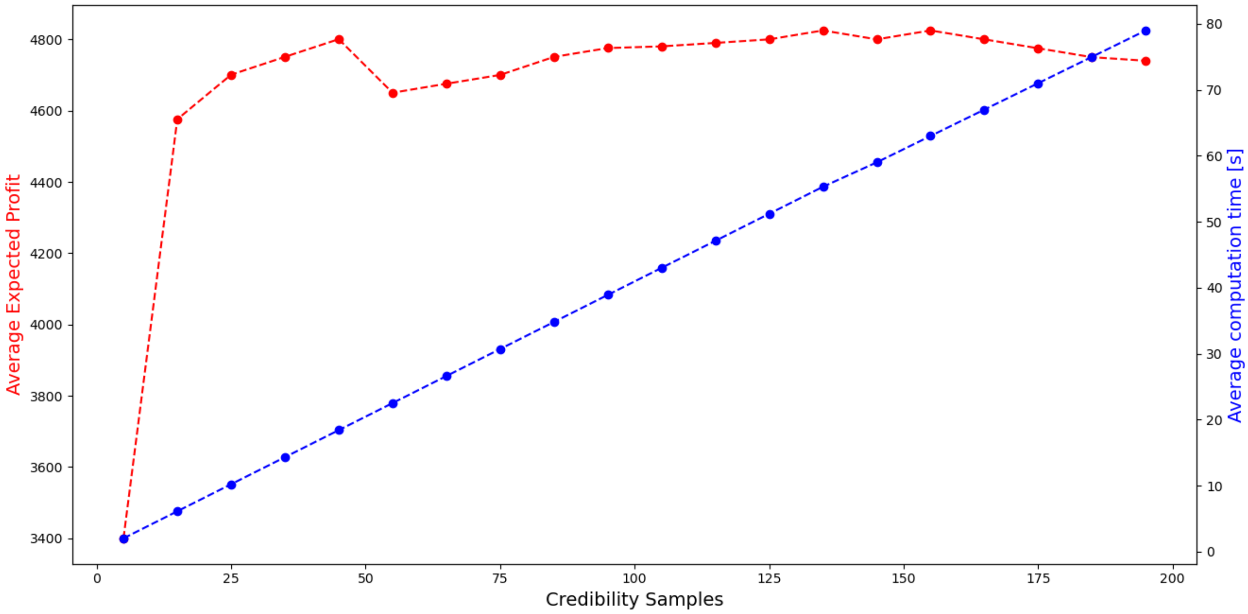

Figure 6.6: Exponential Distribution: Number of Credibility Samples, Expected Profit and Computational Time

Given the results presented, **25 was the choice for the total number of credibility samples**. This value ensures an expected profit assessment time of around 10 seconds, which means a problem with 1000 genes and 5 generations how to take 13 hours and 54 minutes to be completed, just considering fitness calculation time without parallel or cloud computing. Finally, notice that the 25 value can introduce (in rare situations) some instability in the expected profit computation in rare cases, but 10 seconds per solution is already a high computational effort to account for.

Normal Distribution: Credibility Samples

The same study as in section 6.2.2 was applied to the case study considering a normally distributed demand. Results are in the appendix section A.3. The **chosen Credibility samples number was 30.** For 5 runs, this value had a mean fitness value of 4312, a standard fitness deviation of 38,1 and an average computational time of 10,9 seconds (without parallel or cloud computing).

6.2.3 Fitness Stability

After defining the NQM value (revisit section 3.3.3) and the number of credibility samples, it is possible to study the population size and the total number of generations. Combining these two variables is crucial to ensure the algorithm evolves to a point close to its full capabilities without compromising the computational effort. The goal of this section is to tune this combination for each case study.

This analysis used population sizes of 5, 10, 25, 50, 75 and 100, keeping other parameters constant. Among these constant parameters, there was a total number of 20 generations. Apart from the variables previously studied(NQM and Credibility Samples), all other variables result from an empirical. Naturally, these variables need to be further validated (section6.2.5) , but in the author's opinion, they constitute a good starting point to ensure results according to the algorithm capabilities. Table 6.2 shows these parameters:

Table 6.2: Empirical Parameters from Unstructured Testing

Parameters	Value
NQM	50 (Exponential) // 10 (Normal)
Credibility Samples	25 (Exponential) // 30 (Normal)
Resizing	True
Initialization with Null Values	True
Chromosome Normalization	False
Interception Rule	Minimum
Generations	20
Tournament Coefficient	10
Crossover Probability	0,8
Mutation Probability	0,2

Exponential Distribution: Population Size and Generations

Figure 6.7 presents the relation between population size, average fitness value and average computational time. The values shown are the results of 5 algorithm runs per population size.

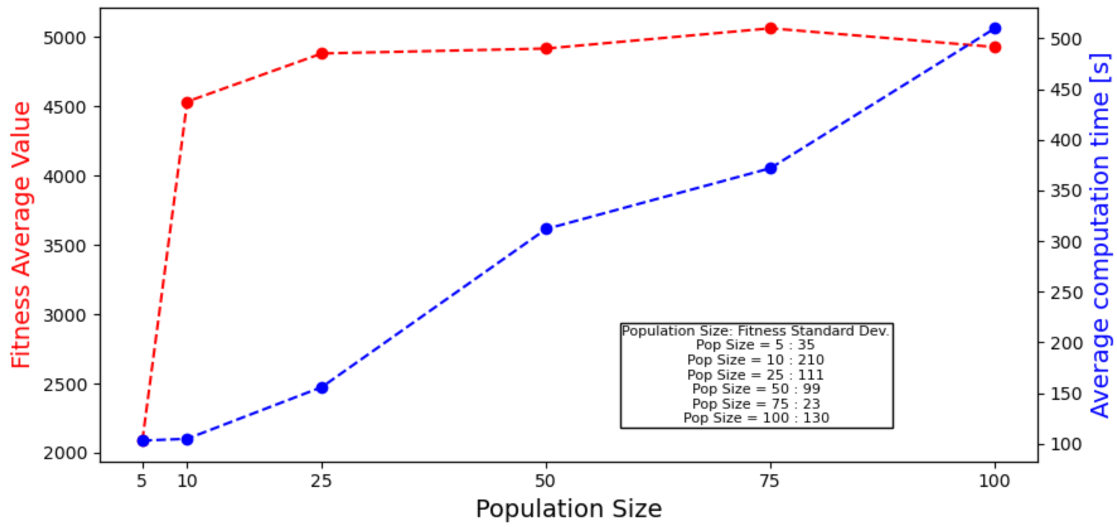

Figure 6.7: Exponential Distribution: Population Size, Fitness and Computational Time

The **selected value for the population size was 75**. Figure 6.8 uses this value and does a similar study but now comparing the number of generations.

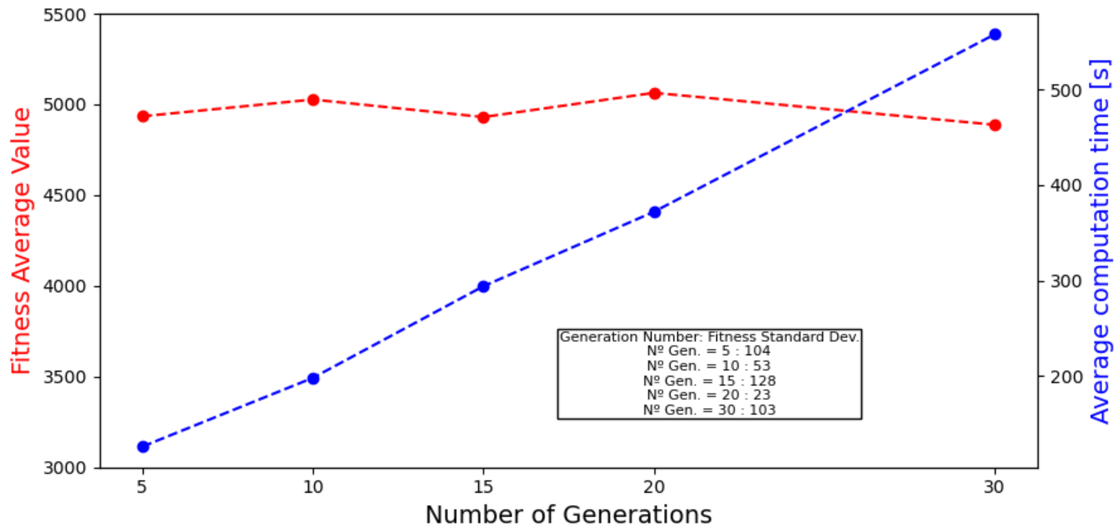

Figure 6.8: Exponential Distribution: Number of Generations, Fitness and Computational Time

Based on this information, **chosen number of generations was 20**.

Normal Distribution: Population Size and Generations

Following the same rationale applied for the exponentially distributed case study, a study tuned the population size and generations number, now for the normally distributed case study. Appendix section A.4 contains the two figures that illustrate this. The study concluded that a **population size of 50 and generations number of 15** were the best values to use.

6.2.4 Membership Function Selection

The parameters previously defined ensured a proper balance between the solution stability and computational time. To have the best algorithm response it is necessary to select the most suitable membership function type and hyper-parameters to represent each probabilistic curve. To do this, a study included the three MF types proposed in section 3.5. The goal is to find the best Membership Functions (MFs) type based on an objective metric: the average profit. The membership function selection bases upon two steps:

1. **MF individual tuning:** Tune each of the membership functions presented in section 3.5.

2. **MF Selection:** Select the MF type and corresponding parameters with the highest average profit.

The MF selection is first done on the exponentially distributed case study, followed by the normally distributed case study. All results were obtained with averages of 5 algorithm runs.

Exponential Distribution: Membership Function Selection

Starting with the **trapezoidal MF**, the values proposed by [4] and presented in table 6.1 served as baseline. Then, different SC and CC values were tested using the expressions in section 3.5.1. In table 6.3 it is possible to observe the different values tested and their results:

Table 6.3: Exponential Distribution: Support and Core Coefficients for Trapezoidal MF

Support C.	Core C.	Average Profit	Average Fitness	Fitness Standard Deviation
0	1	2829,3	4030,9	25,5
0,5	1	2837,8	4525,1	32,7
1	1	2814,5	4921,1	120,3
0	0	2845,8	4160,9	26,4
0	**0,5**	**2860,3**	**4167,6**	**71,4**

The selected combination between the SC and CC was 0 and 0,5, respectively.

In the case of the **exponential MF**, there is only one parameter to be studied: the decay ratio. Table 6.4 shows how this parameter influences the average profit:

Table 6.4: Exponential Distribution: Decay Ratio for Exponential MF

Decay Ratio	Average Profit	Average Fitness	Fitness Standard Deviation
0,01	2485,2	3890,7	92,2
0,5	2836,7	3271,5	26,0
1	**2846,1**	**3666,9**	**79,2**
1,5	2791,4	3876,4	103,2
2	2769,8	4162,9	84,6
3	2789,1	4476,5	94,1

The **chosen value for the decay ratio was 1** with an average profit of 2846,1.

Finally the MF from the mapping in section 3.5.3 are independent of any parameter, generating a unique MF from each probability curve. Table 6.5 shows the results observed:

Table 6.5: Exponential Distribution: Probability Mapping

Av. Profit Generated	Average Fitness	Fitness Standard Deviation
2836,1	**1902,2**	**99,1**

Observing the results from tables 6.3, 6.4 and 6.5, it is possible to see that the trapezoidal MF had the highest average profit generation for SC and CC values of 0 and 0,5, respectively. Thus, this is the choice to represent the probabilistic exponential curves in a Fuzzy environment. The final parameters for the MF are:

Table 6.6: Exponential Distribution: Selected Membership Functions Parameters

Material	MF Parameters
1	[180, 195, 205, 220]
2	[210, 222.5, 227.5, 240]
3	[100, 111.25, 115, 125]
4	[80, 95, 105, 120]
5	[60, 72.5, 77.5, 90]
6	[20, 27.5, 32.5, 40]

Recalling the initial parameters presented in table 6.1, figure 6.9 shows the selected MF for material 1 alongside its original MF from [4].

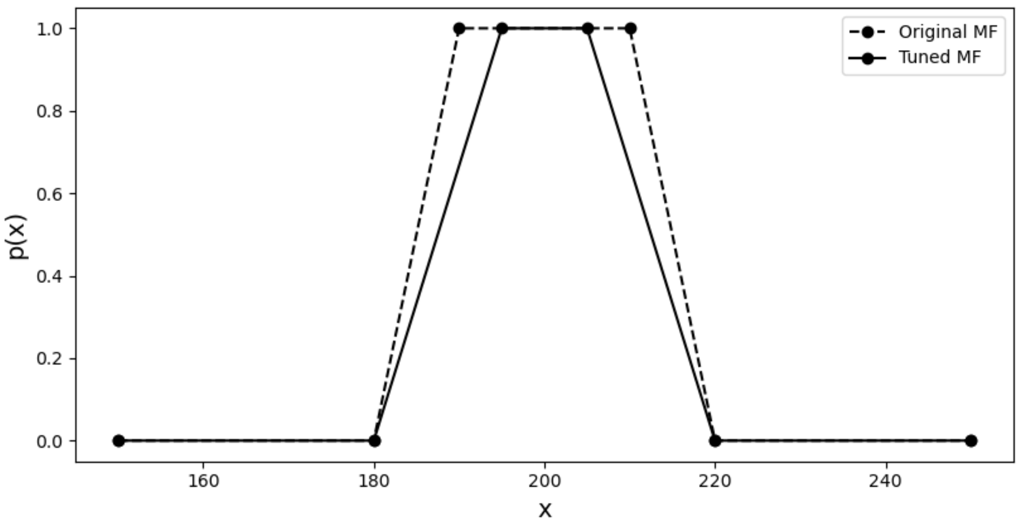

Figure 6.9: Original (from [4]) vs. Tuned Membership Function for Material 1

Normal Distribution: Membership Function Selection

This section aims to find the MF that best describes the probability curves of the normally distributed demand. This is the same procedure as in the section 6.2.4, but for the normally distributed case study. Due to this, Appendix section A.5 contains its results.

Among the three MF types tested, **the exponential MF with a decay ratio of 6 yielded the highest average profit**, being the MF selected for the features tuning. As an example, figure 6.10 displays the MF of material 1, assuming these parameters:

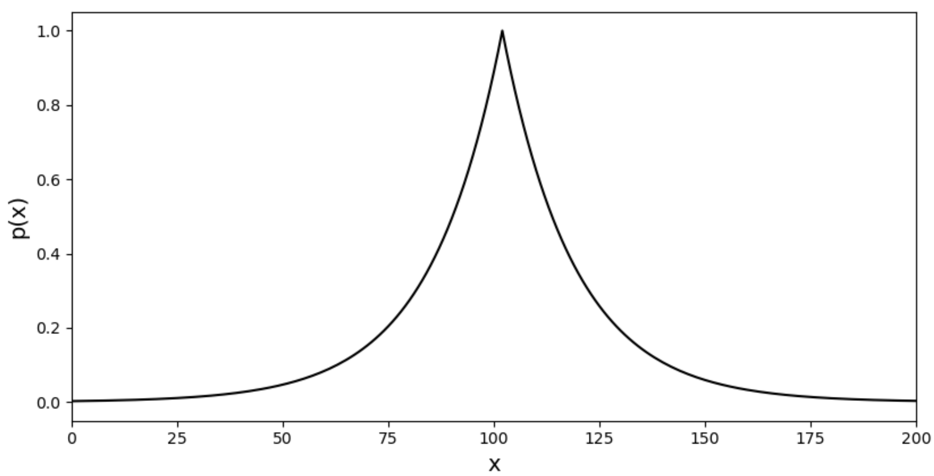

Figure 6.10: Normal Distribution: Selected Membership Function for Material 1

6.2.5 Other Parameters

Having selected the best MF to describe each case study probabilistic demand, it is possible to study the algorithm features that were empirically determined (table 6.2). Moreover, based on how these features influence the algorithm response/behaviour, there are two categories for these features: the solution modification/interpretation features and GA-specific features.

The **solution generation/interpretation features** include:

- Solution Resizing
- Initialization with Null Values
- Chromosome Normalization
- Interception Rule

The **GA-specific features** are:

- Tournament Coefficient (revist equation 4.1)
- Crossover Probability
- Mutation Probability
- Population Size & Number of Generations (studied in section 6.2.3)

The best parameters are the ones that yield the highest fitness values, except in case these parameters alter the way the fitness value is perceived (see interception rule in tables 6.10 and A.5). All results were obtained with averages of 5 algorithm runs.

Exponential Distribution: Other Parameters Tuning

Considering the exponentially distributed case study, the first feature analyzed was the solution resizing. These are the values with and without this feature:

Table 6.7: Exponential Distribution: Solution Resizing

Solution Resizing	Fitness Average	Average Profit	Average Unfeasible Solutions
True	**4167,9**	**2860,3**	**0**
False	3945,1	2293,2	250,4

The **solution resizing proved to be a relevant feature** to increase the solution fitness.

The next algorithm parameter studied was the Initialization with Null Values. Table 6.8 shows how this parameter influences the solution fitness:

Table 6.8: Exponential Distribution: Initialization with Null Values

Initialization with Null Values	Fitness Average	Average Profit
True	4167,9	2860,3
False	**4940,9**	**2853,0**

The average fitness value decreased with the Initialization with Null Values. Thus, it was a disadvantage for the exponential curves case study. Despite this, the smaller fitness value yields a slightly higher average profit. This proves the MF does not perfectly represent the stochastic demand, what was expected behaviour. Section 7.1 discusses this in detail. However, the goal is still to find the algorithm parameters that maximize the fitness value. Due to this reason, **the Initialization with Null values was discarded** for the exponentially distributed case study.

Table 6.9 illustrates the impact the chromosome normalization has on the fitness value maximization.

Table 6.9: Exponential Distribution: Chromosome Normalization

Chromosome Normalization	Fitness Average	Average Profit
True	4889,5	2827,4
False	**4940,9**	**2853,0**

The **chromosome normalization did not have a significant impact** in the fitness value maximization, being discard as a feature in this case study.

Section 3.3.1 presented two types of interception rules: the minimum and the mean interception rule. Table 6.10 reveals how these interception rules influence the fitness and profit average values.

Table 6.10: Exponential Distribution: Interception Rules

Interception Rule	Fitness Average	Average Profit
Minimum	**4940,9**	**2853,0**
Mean	4985,1	2820,2

Contrary to the previously studied features that modified the generation of the solutions, the interception rule changes how the algorithm "interprets" each solution. Consequently, the same solution has different fitness values depending on the interception rule considered. Due to this reason, the decision-making factor, in this case, was the average profit and not the fitness value. As a result, the **minimum interception rule is the selected rule** for the exponentially distributed case study.

Looking at the GA-specific features, the analysis started with the tournament coefficient. This parameter controls the number of individuals randomly selected to a tournament (revisit section 4.2.4, equation

4.1). Table 6.11 shows how different coefficient values influence the fitness.

Table 6.11: Exponential Distribution: Tournament Coefficient

Tournament Coefficient	Average Fitness	Fitness Standard Deviation
2	5006,5	84,7
5	4904,1	69,2
10	4940,9	123,1
15	5017,9	118,1
20	**5106,6**	**57,1**
30	4974,8	97,1
50	4753,3	33,3

The value of **20 was selected as the tournament coefficient value**, due to the higher average fitness yielded.

Another GA-specific feature is the crossover probability. This feature controls how often two tournament winners combine their solutions. For further detail, revisit section 4.2.2. Table 6.12 shows how this probability changes the average fitness and its standard deviation:

Table 6.12: Exponential Distribution: Crossover Probability

Crossover Probability	Average Fitness	Fitness Standard Deviation
0,1	4892,9	159,6
0,3	4882,8	69,1
0,5	4950,3	126,3
0,7	4987,9	103,1
0,8	**5106,6**	**57,1**
0,9	5067,2	60,8

Given the higher fitness average, the **value of 0,8 was kept as crossover probability**.

Finally, table 6.13 displays how changing the mutation probability influences the fitness value.

Table 6.13: Exponential Distribution: Mutation Probability

Mutation Probability	Average Fitness	Fitness Standard Deviation
0,05	4987,1	98,1
0,1	4923,5	104,9
0,2	**5106,6**	**57,1**
0,3	4892,9	64,1
0,4	5001,7	100,8
0,5	5100,1	22,2
0,7	4967,2	135,4
0,9	5015,8	69,2

The **0,2 mutation probability yield the higher average fitness**, thus it is the selected value.

Table 6.14 summarizes the parameters that result from the tuning applied in the exponentially distributed case study:

Table 6.14: Exponential Distribution: Final Parameters

Parameters	**Value**
NQM	50
Credibility Samples	25
Solution Resizing	True
Initialization with Null Values	False
Chromosome Normalization	False
Interception Rule	Minimum
Population Size	75
Generations	20
Tournament Coefficient	20
Crossover Probability	0,8
Mutation Probability	0,2

Normal Distribution: Other Parameters Tuning

Here it goes the tuning of the remaining parameters for the normally distributed case study. Since this tuning process was already presented, its results are in the appendix section A.6, except for two variables. These variables are the **Solution Resizing and Initialization with Null Values**. Tables 6.15 and 6.16 respectively show the impact of this features.

Table 6.15: Normal Distribution: Solution Resizing

Solution Resizing	**Fitness Average**	**Average Profit**	**Average Unfeasible Solutions**
True	**3741,9**	**3797,3**	**0**
False	2208,8	2305,6	122,2

Table 6.16: Normal Distribution: Initialization with Null Values

Initialization with Null values	**Fitness Average**	**Average Profit**
True	**3741,9**	**3797,3**
False	2229,9	2446,6

On the one hand, the solution resizing proved to be a valuable feature but now for the normal distributions. Its inclusion increased the fitness value by 69% and the average profits by 65%. On the other hand, the **initialization with null values proved to be relevant for the normal distributions** increasing the fitness value by 68% and the average profits by 55%. This counters the results presented in table 6.8.

In the author's opinion, the reason why the Initialization is vital in the normally distributed case study is that this case study uses a **low budget** when compared with the overall problem. Therefore, the algorithm naturally selects materials with the highest profit margins by utilizing vectors with null values, rejecting lower margins due to budget constraints.

Table 6.17 summarizes the parameters for the normal curves case study:

Table 6.17: Normal Distribution: Final Parameters

Parameters	Value
NQM	10
Credibility Samples	30
Solution Resizing	True
Initialization with Null Values	True
Chromosome Normalization	False
Interception Rule	Minimum
Population Size	50
Generations	15
Tournament Coefficient	10
Crossover Probability	0,8
Mutation Probability	0,2

6.3 Case Studies Results

This section analyses the algorithm performance for the three decision-making policies proposed in section 3.4. Here demand curves are perfectly described. Under these conditions, the only source of uncertainty comes from the curves' stochastic nature.

Moreover, the analysis uses the algorithm parameters shown in table 6.14 and 6.17, but all MF types best candidates. The use of candidates from all MF types is justified because the MF selection has based on the average profit (correlated with the expected profit). The best candidate can be different for other policy criteria, thus, the study considers all MF best candidates. To ease the reader interpretation, the data used for the membership function selection is presented in appendix chapter B.

The remaining of this section includes two sections specific to each case study (exponential and normal distributions). Each of these sections contain three subsections accounting for each of the three decision-making policies presented in section 3.4, being them:

- Expected Profit Maximization
- Credibility Maximization with Profit Target
- Profit Maximization with Credibility Target

All values result from 5 algorithm runs with constant parameters. The selected run will be the one with the highest fitness value.

6.3.1 Exponential Demand Distribution

This section illustrates the algorithm results for the exponentially distributed case study. The best MF candidate selection for each decision-making policy (appendix B) uses the parameters derived in the tuning procedure (table 6.14).

Exponential Distribution: Expected Profit maximization

Table 6.18 shows the best candidate solution for the expected profit maximization considering the MF resultant from the probability mapping (section 3.5.3).

Table 6.18: Exponential Distribution: Solution for Expected Profit Maximization

Item	**1**	**2**	**3**	**4**	**5**	**6**
Quantity	41,67	35,58	16,07	116,84	75,47	36,91

Comparing to the benchmark solution (table 2.2):

Table 6.19: Exponential Distribution: Expected Profit Maximization Results

Solution	Fitness (Mapping MF)	Average Profit
Classical Benchmark	1572,0	2875,9
Simple Fuzzy GA from [4]	1567,4	2870,9
Fuzzy GA with Novel Mechanisms	**2029,1**	**2906,5**

Exponential Distribution: Credibility maximization for profit target

Table 6.20 shows the best candidate solution for Credibility maximization with a **2000 CU profit target**.

Table 6.20: Exponential Distribution: Solution for Credibility Maximization with 2000 CU Profit Target

Item	1	2	3	4	5	6
Quantity	101,80	101,76	39,09	54,17	40,19	22,41

Comparing to the benchmark solution (table 2.2):

Table 6.21: Exponential Distribution: Credibility maximization results for 2000 CU profit target

Solution	Fitness (Exponential MF)	Profit $>$2000 CU
Classical Benchmark	**0,71**	**75 %**
Simple Fuzzy GA from [4]	0,71	75 %
Fuzzy GA with Novel Mechanisms	0,78	74 %

Appendix section C.1 contains the MF candidates results for the 2500CU and 3000CU profit targets.

Exponential Distribution: Profit Maximization with Credibility Target

Table 6.22 shows the solution with the highest fitness among the exponential membership functions, when considering the profit maximization with a **0,5 Credibility target**.

Table 6.22: Exponential Distribution: Solution for Profit Maximization with 0,5 credibility target

Item	1	2	3	4	5	6
Quantity	130,42	92,35	34,69	56,86	32,70	26,58

Comparing to the benchmark solution (table 2.2):

Table 6.23: Exponential Distribution: Results for Profit maximization with 0,5 Credibility Target

Solution	Fitness (Mapping MF)	Average Profit	Profit >2000 CU
Classical Benchmark	**4876,5**	**2875,9**	**75 %**
Simple Fuzzy GA from [4]	4854,3	2870,9	75 %
Fuzzy GA with Novel Mechanisms	5405,1	2854,5	70 %

Appendix section C.2 contains the MF candidate results for the 0,75 and 0,9 credibility targets.

6.3.2 Normal Demand Distribution

This subsection aims to study the response considering the normally distributed case study. Similar to subsection 6.3.1, the best candidates from each MF (see subsection 6.2.4) are tested for each decision policy and attached in appendix section B.2. Afterwards, the outperforming candidate is compared against the benchmark solution.

Normal Distribution: Expected Profit Maximization

The exponential MF were the candidates showing the highest average profit. Table 6.24 shows the solution with the highest fitness value when considering this MF type.

Table 6.24: Normal Distribution: Solution for Expected Profit Maximization

Item	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Quantity	0	0	0	0	0	114,23	0	0	0	0	0	87,00	44,25	0	0	0	0

Comparing to the benchmark solution (table 2.4):

Table 6.25: Normal Distribution: Expected profit maximization results

Solution	Fitness (Exponential MF)	Average Profit
Classical Benchmark	**3763,2**	**3870,0**
Simple Fuzzy GA replicated from [4]	2208,8	2305,6
Fuzzy GA with Novel Mechanisms	3792,1	3827,3

Normal Distribution: Credibility Maximization with Profit Target

Table 6.26 shows the best solution, among the exponential membership functions, for the Credibility maximization considering a **profit target of 2000CU:**

Table 6.26: Normal Distribution: Solution for Credibility Maximization with 2000 CU Profit Target

Item	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Quantity	3,66	2,10	16,8	6,68	5,27	32,27	11,98	5,33	0	5,35	11,28	5,33	12,86	3,92	0	35,16	11,49

Comparing to the benchmark solution (table 2.4):

Table 6.27: Normal Distribution: Credibility maximization results for 2000 CU profit target

Solution	Fitness (Exponential MF)	Profit >2000 CU
Classical Benchmark	0,95	96 %
Simple Fuzzy GA replicated from [4]	0,96	97 %
Fuzzy GA with Novel Mechanisms	**0,97**	**99** %

Appendix section C.3 contains the candidates results for the profit targets 2500CU, 3000CU, 3500CU and 4000CU.

Normal Distribution: Profit maximization with credibility target

Table 6.28 shows the solution with the highest fitness, among the probability mapping MF, for the profit maximization with a **0,75 Credibility target**,.

Table 6.28: Normal Distribution: Solution for Profit Maximization with 0,75 Credibility Target

Item	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Quantity	0	0	0	0	0	63,51	0	22,66	0	0	24,38	50,37	33,12	19,44	0	9,58	12,89

Comparing to the benchmark solution (table 2.4):

Table 6.29: Normal Distribution: Profit Maximization Results for 0,75 Credibility Target

Solution	Fitness (Mapping MF)	Average Profit	Profit >2500 CU
Classical Benchmark	2228,3	3870,0	93 %
Simple Fuzzy GA replicated from [4]	2343,1	2515,8	93 %
Fuzzy GA with Novel Mechanisms	3231,7	3061,5	98 %

Appendix section C.4 contains the MF candidates results for the 0,5 and 0,9 credibility targets.

6.3.3 Discussion

The proposed framework performed at the same level as the benchmark solutions for both case studies and decision-making policies proving its effectiveness. Despite this, comparison are only suitable for the expected profit maximization since the benchmark solution accounts only for this policy.

Table 6.19 shows the proposed framework slightly outperformed the benchmark solution by 1 % when considering an exponentially distributed demand. In contrast, table 6.25 shows the benchmark solution outperforming the proposed framework by 1 % in the normally distributed case study.

From the mechanisms introduced, the author highlights the **solution resizing** and the **initialization with null values**. On the one hand, tables 6.7 and 6.15 prove the solution resizing usefulness by increasing the average profit by 25 % in the exponentially distributed case study and 65 % in the normally distributed case study. On the other hand, the **Initialization with Null Values proved to be very useful in low budget scenarios** as is the case of the normally distributed case study. Table 6.16 confirms this by presenting an average profit increase of 55 % when introducing this mechanism.

7

Conclusions

Contents

7.1 Main Conclusions

This thesis developed a scalable multi-objective optimization algorithm to solve Newsvendor problems. For this purpose, the designed framework implemented a Fuzzy formulation instead of a classical formulation to solve cases where there is insufficient data to predict the demand distributions and it is necessary the integration of human-expertise knowledge and machine learning. The problem was then solve using a genetic algorithm enhanced by parallel computing in a cloud environment that drastically reduced the running time and made the solution applicable in a real-life scenario.

The algorithm yields excellent results, especially for the expected profit-maximizing, the most important policy in a business context. Tables 6.19 and 6.25 show there is not any performance decrease when comparing the proposed solution to the analytical methods.

From the results derived, it is possible to say the challenge proposed by Siemens was accomplished. Moreover, the framework can integrate real human expertise and machine learning inputs to design the uncertainty around item demand and optimize the decision-making.

Examining the added-value proposal in section 1.2, it is possible to confirm that this thesis accomplished all points. The novel mechanisms introduced in the GA helped improve performance, as tables 6.7, 6.15 and 6.16 prove. Additionally to these performance results, the author wants to reinforce the time reduction provided by integrating cloud and parallel computing techniques. Like table 5.2 exhibits, these techniques introduced a time reduction of 98,3%, which ensures this solution is scalable.

As limitations, there is the inevitable unmatching between membership functions and probabilistic demand curves. This unmatching reduced the framework performance when compared to the other solutions. Table 6.8 is proof of this since a higher fitness value did not yield a higher average profit. Thus, although membership functions can be directly derived from real data, the algorithm performance will always dependent on the quality of the uncertainty assessment.

In conclusion, both author and company are very optimistic after this work and looking forward to implementing this idea in a real factory scenario.

7.2 Future Work

To implement this idea in a real-world scenario, there are areas where the framework directly or indirectly improve. The following points describe these areas:

Integration with Predicting Agent

The algorithm performance is dependent on the quality of the membership functions. Accurate MF can only be obtained with a good predicting agent. This agent can be either a human or a machine learning model, and its main objective is to provide insights regarding the likelihood of each item consumption. For instance, a trust interval and a most likely demand quantity can define a trapezoidal MF.

Different Objective Metrics

This work focused on optimizing the decision-making process based on profit, but other factors, such as service level or customer satisfaction, are also crucial in inventory planning. For this solution to be relevant, those use-cases must be taken into account. The work of [46] and [47] can provide insights regarding this topic.

Improve Fitness Assessment Mechanisms

Section 3.3.3 introduced enhancements in the Credibility estimation process. Despite this, there is still room for improvements in the overall fitness assessment, specially for the expected profit maximization. These improvements are not related to Credibility estimation itself but rather with the profit targets selection.

As section 3.3.4 shown, the expected value is a weighted average of credibility values for different equally spaced profit targets. In reality, there is no need to compute the credibility for smaller profit targets if there is already a higher target with a Credibility equally or very close to 1. In these cases, expected value estimation cloud be optimized by assuming the credibility for smaller profit targets is 1 and focus the computational resources on higher profits target, which is where there are significant variations in the Credibility values. This rationale extrapolates to the profit maximization with a Credibility target by first using a dispersed backward target search and then focusing on the profits that yield Credibility values close to the optimization target.

Different Optimization Algorithms

In the author's opinion, selecting a GA as an optimization algorithm proved to be fruitful. However, the work of [11] shows that there are other options when it comes to meta-heuristic algorithms. His work even suggests that algorithms such as the Bee Colony Optimization would yield better results comparing to a GA.

Bibliography

[1] "misapplying-the-newsvendor-problem @ www.datascienceish.net." [Online]. Available: https://www.datascienceish.net/blog/2020/1/9/misapplying-the-newsvendor-problem

[2] E. Shekarian, N. Kazemi, S. H. Abdul-Rashid, and E. U. Olugu, "Fuzzy inventory models: A comprehensive review," *Applied Soft Computing Journal*, vol. 55, pp. 588–621, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2017.01.013

[3] M. Quintero-Duran, J. E. Candelo, and V. Sousa, "Recent trends of the most used metaheuristic techniques for distribution network reconfiguration," *Journal of Engineering Science and Technology Review*, vol. 10, no. 5, pp. 159–173, 2017.

[4] Z. Shao and X. Ji, "Fuzzy multi-product constraint newsboy problem," *Applied Mathematics and Computation*, vol. 180, no. 1, pp. 7–15, 2006.

[5] T.-M. Choi, *Handbook of Newsvendor Problems. International Series in Operations Research & Management Science.* Springer, 2012.

[6] "Extra! Extra! Here's What You Need to Know about the Newsvendor Problem — by Opex Analytics — The Opex Analytics Blog — Medium." [Online]. Available: https://medium.com/opex-analytics/extra-extra-heres-what-you-need-to-know-about-the-newsvendor-problem-be1d462822c

[7] E. B. Wilson, "The Mathematical Theory of Investment," *Science*, vol. 42, no. 1077, pp. 248–249, mar 1888. [Online]. Available: http://www.jstor.org/stable/2979084

[8] M. Mu, J. Chen, Y. Yang, and J. Guo, "The Multi - product Newsvendor Problem : Re- view and Extensions," *2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing*, pp. 6–9, 2019.

[9] "What is Industry 4.0? Here's A Super Easy Explanation For Anyone." [Online]. Available: https://www.forbes.com/sites/bernardmarr/2018/09/02/what-is-industry-4-0-heres-a-super-easy-explanation-for-anyone/?sh=7795220b9788

[10] D. Petrović, R. Petrović, and M. Vujošević, “Fuzzy models for the newsboy problem,” *International Journal of Production Economics*, vol. 45, no. 1-3, pp. 435–441, 1996.

[11] A. A. Taleizadeh, F. Barzinpour, and H. M. Wee, “Meta-heuristic algorithms for solving a fuzzy single-period problem,” *Mathematical and Computer Modelling*, vol. 54, no. 5-6, pp. 1273–1285, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.mcm.2011.03.038

[12] “Newsvendor model.” [Online]. Available: https://en.wikipedia.org/wiki/Newsvendor{_}model

[13] G. Hadley and T. M. Whitin, “Analysis of Inventory Systems.” *Journal of the American Statistical Association*, vol. 59, no. 305, 1964.

[14] S. Nahmias and C. P. Schmidt, “Problem with a Single Constraint,” -, vol. 3, pp. 463–474, 1984.

[15] H. S. Lau and A. Hing-Ling Lau, “The multi-product multi-constraint newsboy problem: Applications, formulation and solution,” *Journal of Operations Management*, vol. 13, no. 2, pp. 153–162, 1995.

[16] H. S. Lau and A. H. L. Lau, “The newsstand problem: A capacitated multiple-product single-period inventory problem,” *European Journal of Operational Research*, vol. 94, no. 1, pp. 29–42, 1996.

[17] G. L. Vairaktarakis, “Robust multi-item newsboy models with a budget constraint,” *International Journal of Production Economics*, vol. 66, no. 3, pp. 213–226, 2000.

[18] I. Moon and E. A. Silver, “The multi-item newsvendor problem with a budget constraint and fixed ordering costs,” *Journal of the Operational Research Society*, vol. 51, no. 5, pp. 602–608, 2000.

[19] L. Abdel-Malek, R. Montanari, and L. C. Morales, “Exact, approximate, and generic iterative models for the multi-product Newsboy problem with budget constraint,” *International Journal of Production Economics*, vol. 91, no. 2, pp. 189–198, 2004.

[20] L. L. Abdel-Malek and R. Montanari, “An analysis of the multi-product newsboy problem with a budget constraint,” *International Journal of Production Economics*, vol. 97, no. 3, pp. 296–307, 2005.

[21] B. Zhang, X. Xu, and Z. Hua, “A binary solution method for the multi-product newsboy problem with budget constraint,” *International Journal of Production Economics*, vol. 117AZ, no. 1, pp. 136–141, 2009.

[22] M. Khouja, “The single-period (news-vendor) problem: Literature review and suggestions for future research,” *Omega*, vol. 27, no. 5, pp. 537–553, 1999.

[23] L. A. Zadeh, “Fuzzy Sets,” -, pp. 338–353, 1965.

[24] H. Ishii and T. Konno, “A stochastic inventory problem with fuzzy shortage cost,” *European Journal of Operational Research*, vol. 106, no. 1, pp. 90–94, 1998.

[25] L. Li, S. N. Kabadi, and K. P. Nair, “Fuzzy models for single-period inventory problem,” *Fuzzy Sets and Systems*, vol. 132, no. 3, pp. 273–289, 2002.

[26] P. Dutta, D. Chakraborty, and A. R. Roy, “A single-period inventory model with fuzzy random variable demand,” *Mathematical and Computer Modelling*, vol. 41, no. 8-9, pp. 915–922, 2005.

[27] ——, “An inventory model for single-period products with reordering opportunities under fuzzy demand,” *Computers and Mathematics with Applications*, vol. 53, no. 10, pp. 1502–1517, 2007.

[28] R. Martí and G. Reinelt, *The Linear Ordering Problem. Applied Mathematical Sciences.* Springer, 2011.

[29] B. Liu, *Theory and Practice of Uncertain Programming. Studies in Fuzziness and Soft Computing.10.1007/978-3-7908-1781-2*. Springer-Verlag, 2002, vol. 4, no. 3. [Online]. Available: http://marefateadyan.nashriyat.ir/node/150

[30] ——, *Uncertainty Theory. Studies in Fuzziness and Soft Computing. doi:10.1007/978-3-540-39987-2*. Springer, 2004.

[31] B. Liu and Y. K. Liu, “Expected value of fuzzy variable and fuzzy expected value models,” *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 4, pp. 445–450, 2002.

[32] L. A. Zadeh, “F u z z y s e t s as a basis f o r a theory of possibility* l.a. z a d e h,” *Fuzzy Sets and Systems*, vol. 1, no. 1, pp. 3–28, 1978.

[33] W. Pedrycz and F. Gomide, *Fuzzy Systems Engineering*. IEEE, 2007.

[34] J. Jang, C. Sun, and E. Mizutani, “Neuro-Fuzzy and Soft Computing-A Computational Approach to Learning and Machine Intelligence [Book Review],” pp. 1482–1484, 1997.

[35] D. Dubois and H. Prade, “Fuzzy sets and statistical data,” *European Journal of Operational Research*, vol. 25, no. 3, pp. 345–356, 1986.

[36] F. Glover, “Future paths for integer programming and links to artificial intelligence,” *Computers and Operations Research*, vol. 13, no. 5, pp. 533–549, 1986.

[37] J. H. Holland, “Outline for a Logical Theory of Adaptive Systems,” *Journal of the ACM (JACM)*, vol. 9, no. 3, pp. 297–314, 1962.

[38] K. De Jong, D. B. Fogel, and H.-P. Schwefel, “A history of evolutionary computation,” *Handbook of Evolutionary Computation*, no. January, 1997.

[39] M. Oral and O. Kettani, “Reformulating nonlinear combinatorial optimization problems for higher computational efficiency,” *European Journal of Operational Research*, vol. 58, no. 2, pp. 236–249, 1992.

[40] Y. NESTEROV, “Efficiency of cordinate descent methods on huge-scale optimization problems,” *Society*, vol. 31, no. 2, pp. 1082–1119, 2012.

[41] A. N. Andrianov, A. S. Anikin, I. V. Bychkov, and A. Y. Gornov, “Numerical solution of huge-scale quasiseparable optimization problems,” *Lobachevskii Journal of Mathematics*, vol. 38, no. 5, pp. 870–873, 2017.

[42] Microsoft, “The Manycore Shift: Microsoft Parallel Computing Initiative Ushers Computing into the Next Era,” -, no. November, pp. 1–10, 2007.

[43] A. Regalado, “Who Coined 'Cloud Computing'?” [Online]. Available: https://www.technologyreview.com/2011/10/31/257406/who-coined-cloud-computing/

[44] Aws, “What is cloud computing?” [Online]. Available: https://aws.amazon.com/what-is-cloud-computing/

[45] A. A. Taleizadeh, H. Shavandi, and R. Haji, “Constrained single period problem under demand uncertainty,” *Scientia Iranica*, vol. 18, no. 6, pp. 1553–1563, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.scient.2011.11.012

[46] A. J. Brito and A. T. De Almeida, “Modeling a multi-attribute utility newsvendor with partial backlogging,” *European Journal of Operational Research*, vol. 220, no. 3, pp. 820–830, 2012.

[47] A. L. Guiffrida and R. Nagi, “Fuzzy set theory applications in production management research: A literature survey,” *Journal of Intelligent Manufacturing*, vol. 9, no. 1, pp. 39–56, 1998.

A

Extra tuning metrics

A.1 Normal Distribution: Credibility Stability

Figure A.1 presents the values of the variances for the credibility assessment of profits between 0 CU and 9000 CU, with an NQM value of 50:

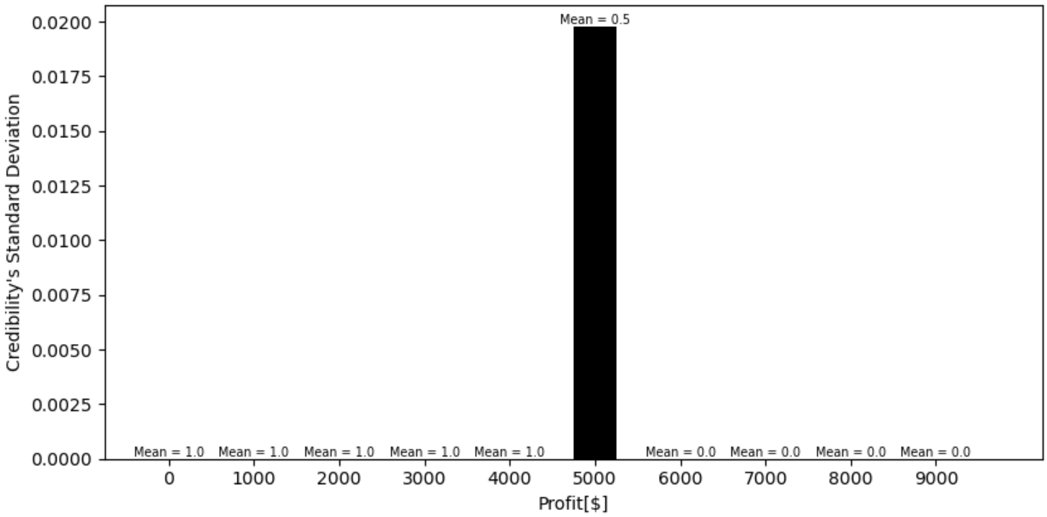

Figure A.1: Exponential Distribution: Credibility Standard Deviation vs. Profit Targets from 0 to 9000 CU

A.2 Normal Distribution: Credibility Stability

Figure A.2 presents the values of the variances for the credibility assessment of profits between 0 and 4500, with an NQM value of 50:

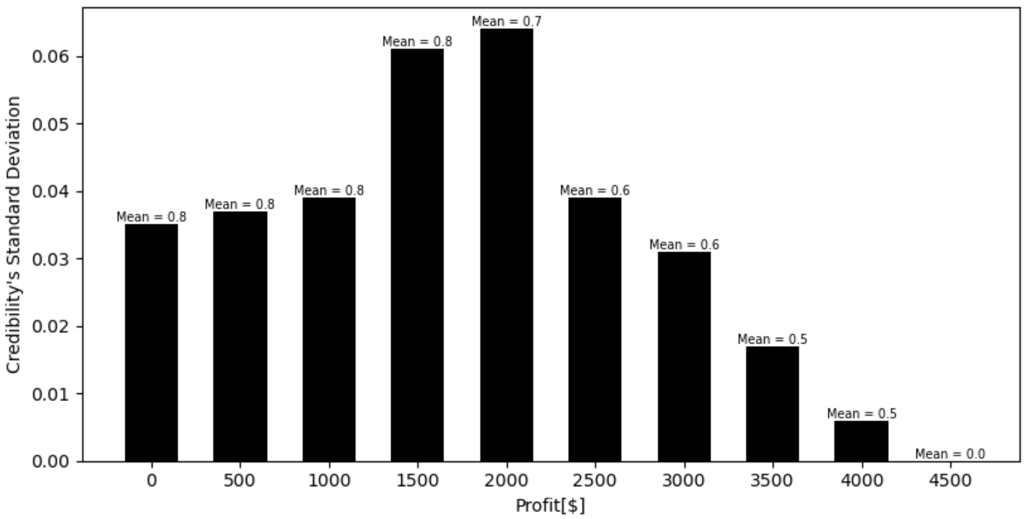

Figure A.2: Normal Distribution: Credibility Standard Deviation vs. Profit Targets from 0 to 4500 CU

The values between 1000 and 2500 shown higher variance values. Due to this, figure A.3 plots all variance between these values with a step of 100 units:

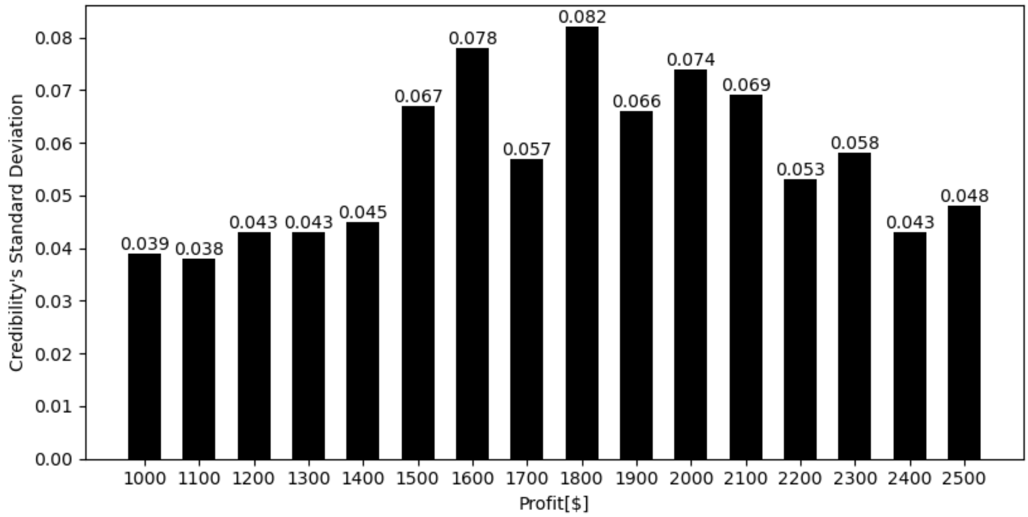

Figure A.3: Normal Distribution: Credibility Standard Deviation vs. Profit Targets from 1000 to 2500 CU

The **profit of 1800 CU was the one showing the highest variance**, thus this is the value for the NQM tuning study. Figure A.4 shows the relation between the NQM values and the credibility standard deviation. Once again, the standard deviations are calculated based on 5 computations per NQM value.

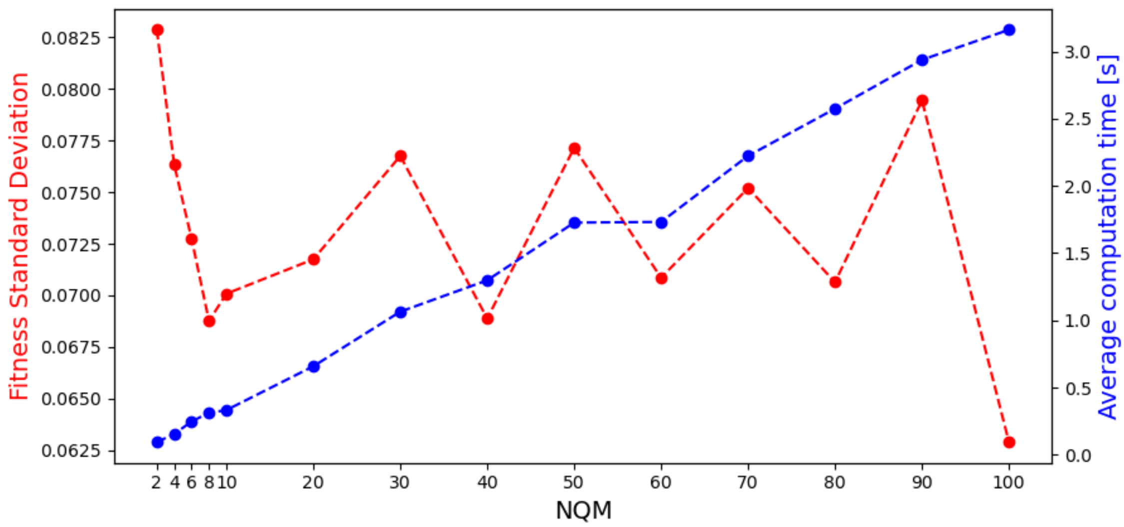

Figure A.4: Normal Distribution: NQM value, Credibility Standard Deviation and Computational Time

With the given information, the NQM value of 10 was chosen. This choice ensures a credibility assessment time similar to the one observed for the case of the exponential curves (when considering an NQM value of 50).

A.3 Normal Distribution: Credibility Samples

Figure A.5 shows how the credibility sampling influences the stability of the expected profit estimation in the case of the normal curves:

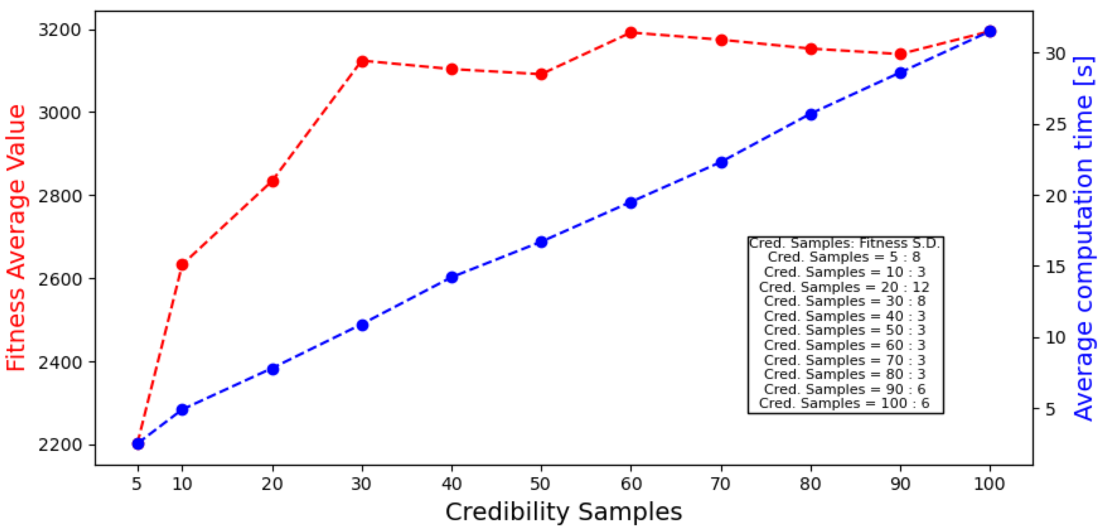

Figure A.5: Normal Distribution: Number of Credibility Samples, Expected Profit and Computational Time

Based on figure A.5, the **chosen value for the credibility samples was 30.**

A.4 Normal Distribution: Population Size and Generations

Figure A.6 contains the population size study fr the normal distributions case-study.

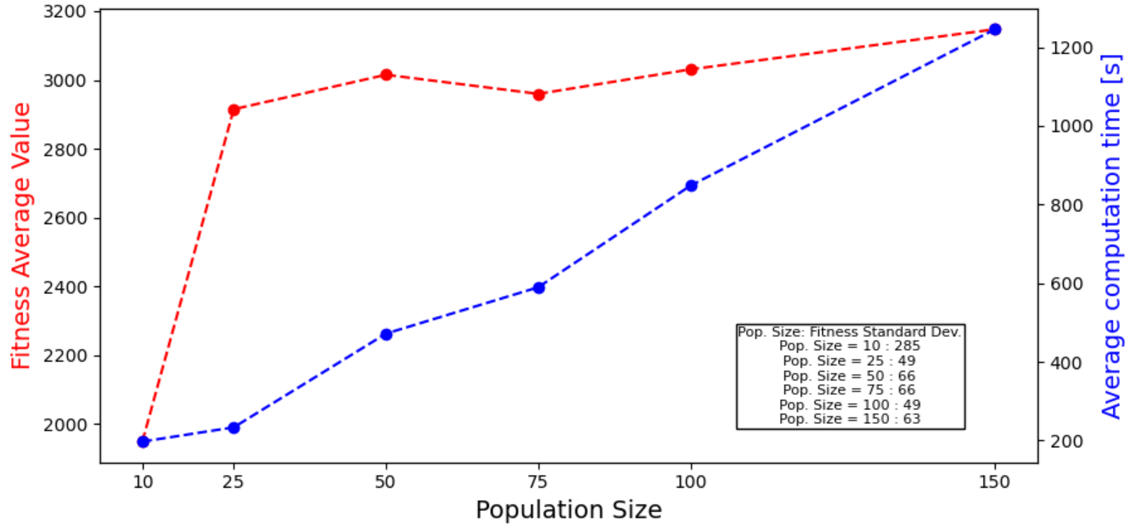

Figure A.6: Normal Distribution: Population Size, Fitness and Computational Time

The **selected population size was 50**, maintaining a close performance time when comparing to the exponential curves example.

Figure A.7 illustrates how the total number of generations affect the fitness value, considering a population size of 50.

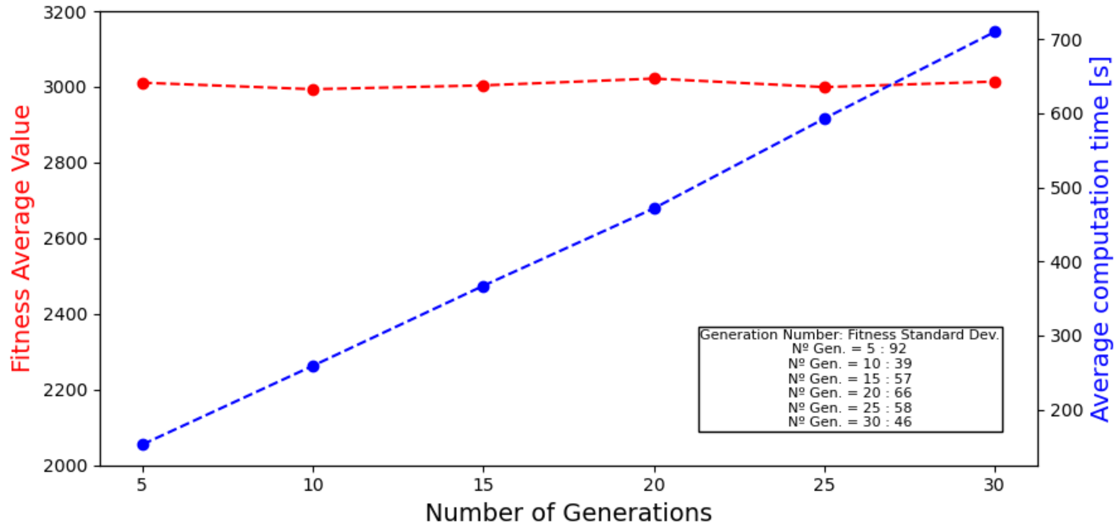

Figure A.7: Normal Distribution: Number of Generations, Fitness and Computational Time

Figure A.7 suggests the total number of generations does not have a significant influence until the 5 generations. Despite this, a **total number of 15 generations was selected** to ensure fitness stability and computational times similar to the case of the exponential curves.

A.5 Normal Distribution: Membership Function Selection

This section contains the tuning results for the normal distributed case-study.

Table A.1 shows how different combinations of SC and CC can influence the average profit generated by the solutions found with the algorithm parameters already defined.

Table A.1: Normal Distribution: Support and Core Coefficients for Trapezoidal Membership Functions

Support C.	Core C.	Average Profit	Average Fitness	Fitness S.D.
0	**1**	3679,5	3545,9	66
0,5	**1**	3582	3385,4	92,3
1	**1**	3541,4	3163,1	72,5
2	**1**	3504,7	2756,5	50,1
4	**1**	3557,7	2280,2	47
0	**0,5**	3705,4	3971,2	85,8
0	**0**	3683,5	4166,1	39,3

With an average profit of 3705,4, the chosen values for the **SC and CC are 0 and 0,5, respectively.** Remember the way the SC and CC modifies the trapezoidal MF is different, depending on each type of probabilistic curve being represented. For further detail see sub-section 3.5.1.

For the exponential MF, it only required to analyse the decay ratio. Table A.2 shows the results, when considering different values for this parameter:

Table A.2: Normal Distribution: Decay Ration for Exponential Membership Functions

Decay Ratio	Average Profit	Average Fitness	Fitness S.D.
0,01	3246,2	1997,4	101,2
0,5	3604,8	2192,2	37,5
1	3660,3	2619,4	64,1
2	3655,9	3085,5	58,3
4	3739,2	3570,2	37,2
6	3797,2	3741,9	44
8	3686,3	3751,6	40,6
10	3725,8	3864,8	31,6

The highest average profit (3797,2) was observed when the for decay ratio equals to 6. Due to this, **the considered decay ratio is 6.**

Finally, table A.3 presents the average profit obtained when applying the mapping presented in section 3.5.3:

Table A.3: Normal Distribution: Probability Mapping

Average Profit	Average Fitness	Fitness S.D.
3702,2	3003,9	57,6

A.6 Normal Distribution: Parameters Tuning

Table A.4 shows how the chromosome normalization effect the variables in study.

Table A.4: Normal Distribution: Chromosome Normalization

Chromosome Normalization	Fitness Average	Average Profit
False	3741,9	3797,3
True	3713,7	3755,7

The **chromosome normalization did not significantly impact the fitness value.** This feature is once again discard from the final algorithm parameters.

Table A.5 introduces the results using either the minimum interception rule or the mean interception rule.

Table A.5: Normal Distribution: Interception Rules

Interception Rule	Fitness Average	Average Profit
Minimum	3741,9	3797,3
Mean	3614,1	3722,8

The selected interception rule was the minimum one, since it shown a higher average profit.

Looking at the GA-specific features, table A.6 presents the results for different tournament coefficient values.

Table A.6: Normal Distribution: Tournament Coefficient

Tournament Coefficient	Average Fitness	Fitness Standard Deviation
2	3684,9	78,2
5	3703,8	33,6
10	3741,9	44
15	3713,4	40,7
20	3646	116,2
30	3629,8	21,7

The average fitness had its highest when the **tournament coefficient equals to 10**, being this the selected value.

Table A.7 reports the results for several crossover probability values.

Table A.7: Normal Distribution: Crossover Probability

Crossover Probability	Average Fitness	Fitness Standard Deviation
0,1	3465,2	157,8
0,3	3644,8	89
0,5	3650,1	20,8
0,7	3657	16,3
0,8	3741,9	44
0,9	3724,5	41,2

The **0,8 crossover probability was the value with the highest average fitness**, thus it is the chosen one.

Lastly, table A.8 presents the results for the mutation probability.

Table A.8: Normal Distribution: Mutation Probability

Mutation Probability	Average Fitness	Fitness Standard Deviation
0,02	3653,6	43,5
0,05	3692,9	28,3
0,1	3698	34,1
0,2	3741,9	44
0,3	3715,6	41,4
0,5	3728,4	27,9
0,7	3701,2	67,5
0,9	3701	45,3

The **0,2 probability mutation probability yielded the highest fitness on average**. With the choice of this value it is possible to see that all the empirical parameters proved to be the most efficient for the normal curves example.

B

Membership Function Selection Data for All Decision Policies

B.1 Exponential Distribution: Membership Function Selection

Exponential Distribution: Expected Profit Maximization

Table B.1 contains the results of each MF type candidate for the expected profit maximization considering the exponentially distributed case-study.

Table B.1: Exponential Distribution: Expected profit maximization

MF Type:	Average Fitness	Average Profit	Best Fitness	Best Fitness Profit
Trapezoidal	5044,9	2820,4	5120,4	2834,4
Exponential	3658,6	2873,9	3725	2859,3
Mapping	**1949,1**	**2886,1**	**2029,1**	**2906,5**

Exponential Distribution: Credibility Maximization for Profit Target

Table B.2 summaries the candidates results when considering a **profit target of 2000CU:**

Table B.2: Exponential Distribution: Credibility Maximization with a 2000 CU profit target

MF Type:	Best Solution Fitness	Best Solution >2000 CU	Average Fitness
Trapezoidal	1	70 %	1
Exponential	**0,78**	**74 %**	**0,78**
Mapping	0,49	53 %	0,48

Exponential Distribution: Profit Maximization with Credibility Target

Table B.3 summaries the MF candidate results when considering a **0,5 credibility target**:

Table B.3: Exponential Distribution: Profit maximization for 0,5 credibility target

MF Type:	Average Fitness	Best Solution Fitness	Av. Profit	Best Sol. >2000 CU
Trapezoidal	5524,3	5634,8	2820,7	68 %
Exponential	**5405,1**	**5474,4**	**2854,5**	**70 %**
Mapping	248	314,6	1125,6	0 %

B.2 Normal Distribution: Membership Function Selection

Normal Distribution: Expected Profit Maximization

Table B.4 contains the results of each MF type candidate for the normally distributed case study, when maximizing the expected profit.

Table B.4: Normal Distribution: Expected profit maximization

MF Type:	Average Fitness	Average Profit	Best Fitness	Best Fitness Profit
Trapezoidal	3948,3	3697,5	4005,7	3741,9
Exponential	**3751,2**	**3785,9**	**3792,1**	**3827,3**
Mapping	2954,8	3627,2	3027,30	3709,8

Normal Distribution: Credibility Maximization with Profit Target

Table B.5 summaries the MF candidates results when considering a **profit target of 2000CU:**

Table B.5: Normal Distribution: Credibility optimization with 2000 profit target

MF Type:	Best Solution Fitness	Best Solution $>$2000 CU	Average Fitness
Trapezoidal	1	81 %	1
Exponential	**0,97**	**99 %**	**0,97**
Mapping	0,98	99 %	0,97

Normal Distribution: Profit Maximization with Credibility Target

Table B.6 summaries the MF candidates results when considering a **0,75 credibility target**:

Table B.6: Normal Distribution: Profit maximization for 0,75 credibility target

MF Type:	Average Fitness	Best Solution Fitness	Av. Profit	Best Sol. $>$2500 CU
Trapezoidal	3832	3930,1	3655,4	92 %
Exponential	3781,7	4036,8	3583,9	91 %
Mapping	**2926,9**	**3231,7**	**3061,5**	**98 %**

C

Extra Results

C.1 Exponential Distribution: Credibility Maximization with Profit Target

Table C.1 summaries the results when considering a **profit target of 2500CU:**

Table C.1: Exponential Distribution: Credibility Maximization with 2500 Profit Target

MF Type:	Best Solution Fitness	Best Solution $>$2500 CU	Average Fitness
Trapezoidal	1	39 %	1
Exponential	**0,76**	**50 %**	**0,76**
Mapping	0,48	41 %	0,48

The benchmark solution had profits above the 2500CU 62% of the times across all simulations.

Table C.2 summaries the results when considering a **profit target of 3000CU:**

Table C.2: Exponential Distribution: Credibility optimization with 3000 profit target

MF Type:	Best Solution Fitness	Best Solution $>$3000 CU	Average Fitness
Trapezoidal	1	18 %	1
Exponential	**0,74**	**30 %**	**0,74**
Mapping	0,47	27 %	0,47

The benchmark solution had profits above the 3000CU 48% of the times across all simulations.

C.2 Exponential Distribution: Profit Maximization with Credibility Target

Table C.3 summaries the results when considering a **0,75 credibility target**:

Table C.3: Exponential Distribution: Profit maximization for 0,75 credibility target

MF Type:	Average Fitness	Best Solution Fitness	Av. Profit	Best Sol. $>$2000 CU
Trapezoidal	**4948**	**5026,5**	**2818,1**	**73 %**
Exponential	2612,4	2690,4	2433,9	72 %
Mapping	176,1	241,5	1006	0 %

Table C.4 summaries the results when considering a **0,9 credibility target**:

Table C.4: Exponential Distribution: Profit maximization for 0,9 credibility target

MF Type:	Average Fitness	Best Solution Fitness	Av. Profit	Best Sol. $>$2000 CU
Trapezoidal	**4800,5**	**4841,1**	**2797,9**	**73 %**
Exponential	984,4	1025,3	2306,1	72 %
Mapping	136,7	163,3	1058,4	0 %

C.3 Normal Distribution: Credibility Maximization with Profit Target

Table C.5 summaries the results when considering a **profit target of 2500CU:**

Table C.5: Normal Distribution: Credibility optimization with 2500 profit target

MF Type:	Best Solution Fitness	Best Solution $>$2500 CU	Average Fitness
Trapezoidal	1	72 %	1
Exponential	**0,97**	**85 %**	**0,96**
Mapping	0,93	97 %	0,92

Table C.6 summaries the results when considering a **profit target of 3000CU:**

Table C.6: Normal Distribution: Credibility optimization with 3000 profit target

MF Type:	Best Solution Fitness	Best Solution $>$3000CU	Average Fitness
Trapezoidal	1	64 %	1
Exponential	**0,96**	**92 %**	**0,95**
Mapping	0,79	89 %	0,75

Table C.7 summaries the results when considering a **profit target of 3500CU:**

Table C.7: Normal Distribution: Credibility optimization with 3500 profit target

MF Type:	Best Solution Fitness	Best Solution $>$3500CU	Average Fitness
Trapezoidal	1	64 %	1
Exponential	**0,96**	**92 %**	**0,95**
Mapping	0,79	89 %	0,75

Table C.8 summaries the results when considering a **profit target of 4000CU:**

Table C.8: Normal Distribution: Credibility optimization with 4000 profit target

MF Type:	Best Solution Fitness	Best Solution $>$4000CU	Average Fitness
Trapezoidal	0,5	38 %	0,5
Exponential	**0,85**	**62 %**	**0,76**
Mapping	**0,51**	**62 %**	**0,5**

C.4 Normal Distribution: Profit Maximization with Credibility Target

Table C.9 summaries the results when considering a **0,5 credibility target**:

Table C.9: Exponential Distribution: Profit maximization for 0,5 credibility target

MF Type:	Average Fitness	Best Solution Fitness	Av. Profit	Best Sol. $>$2500 CU [%]
Trapezoidal	4556,9	4584	3570,7	79 %
Exponential	**4238**	**4299,2**	**3695,5**	**86 %**
Mapping	4501,1	4541,2	3597,5	81 %

Table C.10 summaries the results when considering a **0,9 credibility target**:

Table C.10: Exponential Distribution: Profit maximization for 0,9 credibility target

MF Type:	Average Fitness	Best Solution Fitness	Av. Profit	Best Sol. $>$2500 CU
Trapezoidal	2386	2489,6	2570,3	97 %
Exponential	**3212,6**	**3557,3**	**3275,3**	**96 %**
Mapping	3832	3985	3655,4	91 %