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

Rodrigo Manuel Madeira Cunha Luís
rodrigo.luis@tecnico.ulisboa.pt
Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal
December 2021

Abstract—Often innovative and disruptive products do not have sufficient data to accurately predict the demand probability distributions used in traditional inventory planning solutions. This work studies the Newsvendor problem and proves that a Fuzzy formulation is a valid alternative to the probability methods with the advantage of easily integrating human-expertise knowledge and machine learning when it is necessary to make up for the insufficient data. 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, this paper introduces an uncertainty simulation procedure that uses possible demand scenarios to compare 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 %.

Index Terms—Newsvendor problem; Single-period inventory; Fuzzy Logic; Genetic algorithm; Cloud environment

1 Introduction

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” [1]. Based on this dilemma, a fundamental problem on inventory management takes shape: ”The Newsvendor Problem”.

The literature offers a wide range of solutions to solve the Newsvendor Problem (see [6]). 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.) to risk-averse techniques. However, they tend to use probability density functions to model the uncertain demand.

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 this limitations, being Fuzzy logic a suitable tool to perform this integration. A Fuzzy environment can use few data points to describe uncertainty through meaningful Membership Function(s) (MF). 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.

To design a framework capable of solving a real Multi-Item Newsvendor Problem (MINP) this thesis uses a fuzzy formulation, alongside an enhanced genetic algorithm based on the work of [2] and computation techniques that ensure reproducibility in bigger problems.

Contributions

The following topics summarize the added-value proposal:

Enhancements in the Credibility estimation: This thesis simultaneously improves the performance and running time of the Credibility estimation framework proposed in [2]. These enhancements are 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: This thesis 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: This work explains why scalability is essential in optimization problems, how to implement parallel computing in a 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: This thesis 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.

- *Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal*
- *E-mail: rodrigo.luis@tecnico.ulisboa.pt*

Manuscript received November, 2021; revised December, 2021.

2 CLASSICAL MULTI-ITEM NEWSVENDOR PROBLEM

2.1 Formulation

The classical formulation suggested in [3] uses a modified form of the original model proposed in 1964 by [4]. This form minimizes the expected cost function, being this minimization equivalent to maximize an "expected profit" function [3]. 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 1 represents the model described.

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

Subject to

$$\sum_{i=1}^{N} c_i x_i \leq B_G \tag{2}$$

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
- $F_i(D_i)$: 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

2.2 Case Studies

The author selected two works to serve as benchmarks for different case studies. The selected works are [3], [5]. These works are simple to understand, objective and have good comparisons to other works ([2], [6]).

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 .

2.2.1 *Exponential Demand Distribution*

The first scenario is the one proposed in [3]. Here, the **item demand is exponentially distributed**. Equations 3 and 4 respectively define the probability density function and cumulative distribution function of an exponential distribution with a mean value μ.

$$f(x;\mu) = \begin{cases} 0, & x < 0 \\ \frac{1}{\mu} e^{-\frac{x}{\mu}}, & x \geq \mu \end{cases} \tag{3}$$

$$F(x;\mu) = \begin{cases} 0, & x < 0 \\ 1 - e^{-\frac{x}{\mu}}, & x \geq \mu \end{cases} \tag{4}$$

The work of [3] studied this demand-type considering a problem with six items and a budget of 3500 CU. Table 1 presents the material data proposed:

TABLE 1
Exponential Distribution: Relevant Data

Item	v_i (CU)	h_i (CU)	c_i (CU)	μ_i
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 [3] 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 [3]. Table 2 shows the proposed solution to optimize the expected profit.

TABLE 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

2.2.2 *Normal Demand Distribution*

The second case study extracted from the literature is [5]. In this case, **normal distributions describe each item demand**. Equations 5 and 6 respectively define the probability density function and cumulative distribution function of a normal distribution with mean value μ and standard deviation σ.

$$f(x;\mu) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{5}$$

$$F(x;\mu) = \frac{1}{2}[1 + erf(\frac{x-\mu}{\sigma\sqrt{2}})] \tag{6}$$

Where:

$$erf(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^{x} e^{-t^2} dt \tag{7}$$

The study of this type of distribution included 17 materials and a budget of 2500 CU. Table 3 contains each material specific data:

TABLE 3
Normal Distribution: Relevant Data

Item	v_i (CU)	h_i (CU)	c_i (CU)	μ_i	σ_i
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 [3], the solution framework used in [5] introduced a way of ”deleting products, in ascending order, that have low marginal utility”. The reader can consult the overall solution procedure in [5]. Table 4 presents the proposed solution for maximizing the expected profit considering this case study.

TABLE 4
Normal Distribution: Benchmark Solution

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

3 Fuzzy Multi-Item Newsvendor Problem

3.1 Credibility Theory

Credibility theory was introduced by [7], [8]. [2] used some of the concepts proposed to define objective functions that describe multiple decision-making policies (section 3.3). This section helps the reader understanding these concepts by looking at their definition.

The Possibility, Necessity and Credibility of a fuzzy event. Formally, their definition is:

$$Pos\{\xi \geq r\} = sup\, \mu\{u\}\ ,\ u \geq r \tag{8}$$

$$Nec\{\xi \geq r\} = 1 - sup\, \mu\{u\}\ ,\ u \leq r \tag{9}$$

$$Cr\{\xi \geq r\} = \frac{1}{2}[Pos\{\xi \geq r\} + Nec\{\xi \geq r\}] \tag{10}$$

In plain English, the Possibility (equation 8) of a fuzzy variable being larger than a specified value *r* is equal to the largest membership grade found for values larger or equal than *r*. The Necessity of a fuzzy variable being larger than a specified value *r* (equation 9) is the standard complement [9] (unit minus the membership grade) of the largest membership grade found for values smaller or equal than *r*. Finally, the Credibility (equation 10) 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 [2]:

$$E[\xi] = \int_0^{\infty} Cr\{\xi \geq r\}\, dr - \int_{-\infty}^{0} Cr\{\xi \leq r\}\, dr \tag{11}$$

From a conceptual perspective, this definition can be perceived as the difference between the Credibility of ξ assuming positive values minus the Credibility of ξ assuming negatives values.

3.2 Multi-Item Fuzzy Extension

Section 3.1 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-product problems** because each product 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 products), 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 3.2.1). As a matter of curiosity, following the reasoning explained, a simple multi-product problem with ten products, where each product 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.

[2] 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 3.1.

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 3.2.3 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 3.2.4 explains how to use different Credibility samples to estimate the expected profit of a multi-item solution.

3.2.1 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 [2]) and the mean.

Let us assume $u_k = (u_{1k}, u_{2k}, ..., u_{nk})$ is a demand vector of n elements, $\mu(u_k)$ its estimated membership grade and $\mu(u_nk)$ the membership grade associated with each item proposed ordering quantity. The definition of the **minimum conjunctive operator** is:

$$\mu(u_k) = \mu(u_{1k}) \cap \mu(u_{2k}) \cap ... \cap \mu(u_{nk}) = min(\mu(u_{1k}), \mu(u_{2k}), ...\mu(u_{nk})) \tag{12}$$

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

$$\mu(u_k) = \mu(u_{1k}) \cap \mu(u_{2k}) \cap ... \cap \mu(u_{nk}) = \frac{\mu(u_{1k}) + \mu(u_{2k}) + ... + \mu(u_{nk})}{n} \tag{13}$$

3.2.2 Possibility and Necessity Estimation

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 mot 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:

$$\widetilde{Pos}\{F(x, u_k) \geq F_0\} = \max_{1 \leq k \leq N} \{\mu(u_k) | F(x, u_k) \geq F_0\} \tag{14}$$

$$\widetilde{Nec}\{F(x, u_k) \geq F_0\} = 1 - \max_{1 \leq k \leq N} \{\mu(u_k) | F(x, u_k) \leq F_0\} \tag{15}$$

Where $F(x, u_k)$ is the profit function and N the total number of random demand vectors. Recalling the definition proposed in equation 10, the estimated credibility is then:

$$\widetilde{Cr}\{F(x, u_k) \geq F_0\} = \frac{1}{2}[\widetilde{Pos}\{F(x, u_k) \geq F_0\} + \widetilde{Nec}\{F(x, u_k) \geq F_0\}] \tag{16}$$

3.2.3 Proposed Credibility Estimation Framework

Figure 1 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 NQM. Additionally, the levels are defined by 10% quantiles.
   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.

3.2.4 Expected Value Estimation

This section intends to extend the definition of the **expected value of a Fuzzy variable** ξ (equation 11) to multi-item problems. The rationale is similar to previous section 3.2.3 but instead of the Credibility, the estimation focus on the expected value.

As equation 11 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 17, 18 and 19 describe the steps to estimate the expected profit:

$$E_1 = -\sum \widetilde{Cr}\{F(x, u_k) \leq r_i\}, \quad if\ r_i < 0 \tag{17}$$

$$E_2 = E_1 + \sum \widetilde{Cr}\{F(x, u_k) \geq r_i\}, \quad if\ r_i \geq 0 \tag{18}$$

$$E = E_2 \times \frac{(r_{N_{cr}} - r_1)}{N_{cr}} + max(0, r_1) + min(0, r_{N_{cr}}) \tag{19}$$

Where:

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

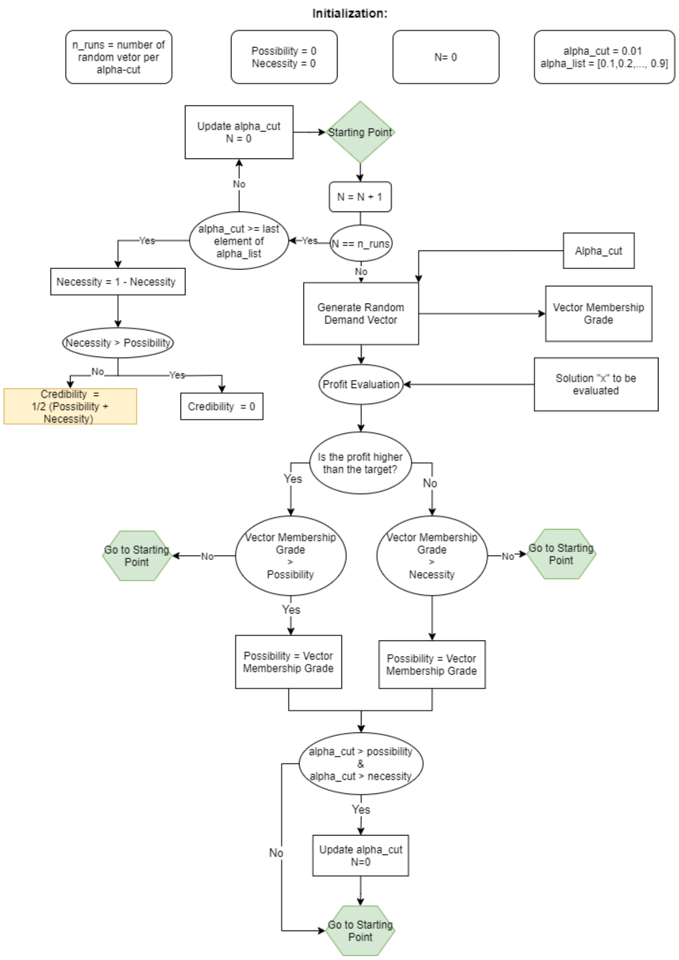

Fig. 1. Credibility Estimation for Fixed Profit Target

This work considers the number of credibility samples as a variable that must be studied being a crucial parameter to the algorithm performance . 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 1, the definition of minimum and maximum possible profits is:

$$P_{min} = -\sum_{i=1}^{N} c_i x_i = -B_G \quad (20)$$

$$P_{max} = \sum_{i=1}^{N} (v_i - c_i) x_i \quad (21)$$

$$P_{int} = [P_{min}, P_{max}] \quad (22)$$

Regarding the distribution between samples, it must ensure all samples as equally spaced between each other.

3.3 Fuzzy Decision-Making Policies

Section 3.2 proved 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.

3.3.1 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 20, 21 and 22
   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 17, 18 and 19 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

3.3.2 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 3.2.3) 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

3.3.3 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 21)
   b) Estimate credibility for current profit target using the procedure on section 3.2.3
   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

3.4 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.

3.4.1 Trapezoidal Membership Functions

A trapezoidal membership function is defined by four parameters (a, b, c, d) [10]. These four parameters define five different line segments alongside the complete universe of discourse. A succinct definition of a trapezoidal MF is:

$$m(x;a,b,c,d) = max(min(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}), 0) \tag{23}$$

3.4.2 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:

$$m(x;d) = \begin{cases} e^{d\times\frac{x-\mu}{\mu}}, & x \leq \mu \\ e^{-d\times\frac{x-\mu}{\mu}}, & x > \mu \end{cases} \tag{24}$$

3.4.3 Probability Mapping

The final way proposed to represent a probabilistic distribution in a Fuzzy environment is by using the mapping presented by [11] and summarized by [12]. The mapping consists of the following steps:

1) **Compute the probability of each element:**
   Apply the following operations to a finite number elements:

$$p(n) = \begin{cases} \int_{n-0,5}^{n+0,5} f(x)dx, & n \,\epsilon\, \mathbb{N}\backslash\{0\} \\ \int_{n}^{n+0,5} f(x)dx, & n = 0 \end{cases} \tag{25}$$

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

$$\Omega = \{w_1, w_2, .., w_I\}, \quad p(w_1) \geq ... \geq p(w_I) \tag{26}$$

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

$$m(w_1) = \sum_{k=1}^{I} p(w_k) \simeq 1 \tag{27}$$

$$m(w_i) = i.p(w_i) + \sum_{k=i+1}^{I} p(w_k), \quad i = 2, ...I-1 \tag{28}$$

$$m(w_I) = I.p(w_I) \tag{29}$$

4 OPTIMIZATION ALGORITHM

A genetic algorithm was chosen to solve this optimization problem. Section 4.1 defines the novel mechanisms introduced to fit the Fuzzy MINP better and section 4.2 presents the computational techniques used to reduce the running time.

4.1 Problem-Specific Enhancements

4.1.1 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) Compute resizing ratio: Apply equation 30
3) Apply resizing: Multiply all ordering quantities by the resizing ratio

The resizing ratio is given by:

$$R_{ratio} = \frac{B_G}{\sum_{i=1}^{N} c_i x_i} \tag{30}$$

Recalling the notation presented in section 2.1:

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

4.1.2 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 4.1.1)

4.1.3 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 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 31
2) Crossover or Mutation
3) Chromosome Restoration: Multiply normalized ordering quantity by its respective expected value

The proposed normalization is given by:

$$x_{ni} = \frac{x_i}{\mu_i} \tag{31}$$

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

4.2 Computational Performance

The proposed algorithm ran for the case study with exponentially distributed demands (revisit section 2.2.1) considering the different machines in table 5 and applying parallel computing. Figure 2 plots the relation between the number of vCPU and running time.

TABLE 5
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

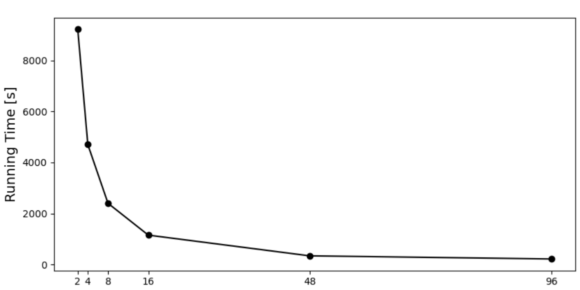

Fig. 2. Number of vCPU per machine and Running time

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 6 shows the running time with and without parallelism both using machine 5.

TABLE 6
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 6 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.

5 Results

This chapter uses examples from the literature to test the algorithm performance against classical solutions. These examples are the case studies presented in section 2.2.1 and 2.2.2. This chapter is organized in the following sections:

- **Section 5.1: Simulation procedure-** Explains how to simulate reality and extract metrics such as the average profit generated.
- **Section 5.2: Algorithm Tuning-** Tunes the algorithm to obtain the best possible performance.
- **Section 5.3: Case Studies Results-** Presents and discusses the most pertinent results for both case studies.

5.1 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. [2], [13], [14]) 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 remove this limitation. 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 3 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"*.

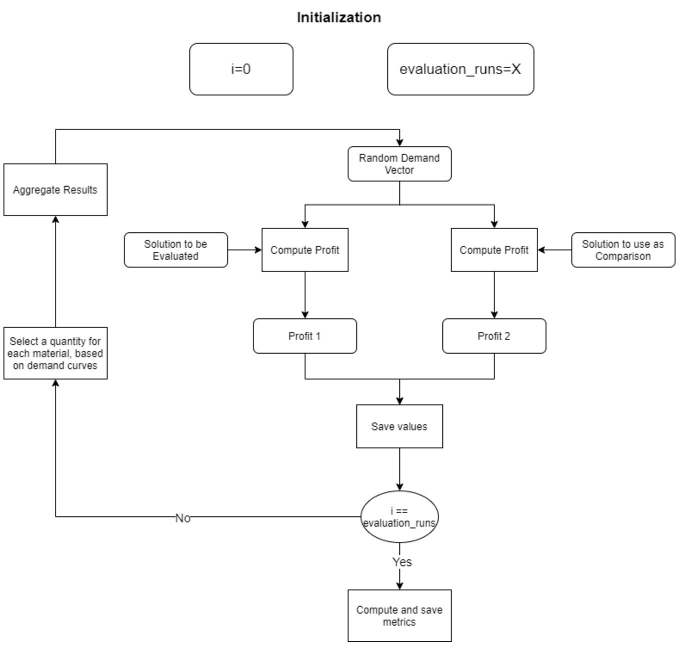

Fig. 3. Simulation Procedure

5.2 Algorithm Tuning

This section represents the necessary parameters tuning to have a reliable algorithm performance. For each case study presented in section 2.2, 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 3.2.3) aiming to find a good balance between accuracy and running time.
2) **Expected Profit Estimation:** As explained in section 3.2.4, 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.

From the overall tuning process two features must be highlighted: the solution resizing and the initialization with null values. Analyse tables 7, 8, 9 and 10.

TABLE 7
Exponential Distribution: Solution Resizing

Sol. Resizing	Fitness Av.	Av. Profit	Av. Unfeasible Sol.
True	**4167,9**	**2860,3**	**0**
False	3945,1	2293,2	250,4

TABLE 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**

TABLE 9
Normal Distribution: Solution Resizing

Sol. Resizing	Fitness Av.	Av. Profit	Av. Unfeasible Sol.
True	**3741,9**	**3797,3**	**0**
False	2208,8	2305,6	122,2

TABLE 10
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 for both case studies. 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 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 11 summarizes the parameters that result from the tuning applied in the exponentially distributed case study:

TABLE 11
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

Table 12 summarizes the parameters for the normal curves case study:

TABLE 12
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

5.3 Main Results

This section analyses the algorithm performance for the most expected profit maximization, the policy for which the benchmark solutions were designed. Here demand curves are perfectly described. Under these conditions, the only source of uncertainty comes from the curves' stochastic nature.

The remaining of this section includes two sections specific to each case study , exponential and normal distributions. All values result from 5 algorithm runs with constant parameters. The selected run will be the one with the highest fitness value.

5.3.1 Exponential Demand Distribution

This section illustrates the algorithm results for the exponentially distributed case study.

Table 13 shows the best candidate solution for the expected profit maximization considering the MF resultant from the probability mapping (section 3.4.3).

TABLE 13
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):

TABLE 14
Exponential Distribution with Expected Profit Maximization Results

Solution	Fitness (Map. MF)	Av. Profit
Classical Benchmark	1572,0	2875,9
Simple Fuzzy GA from [2]	1567,4	2870,9
Fuzzy GA with Novel Mechanisms	**2029,1**	**2906,5**

5.3.2 Normal Demand Distribution

This subsection aims to study the response considering the normally distributed case study.

The exponential MF were the candidates showing the highest average profit. Table 15 shows the solution with the highest fitness value when considering this MF type.

TABLE 15
Normal Distribution: Solution for Expected Profit Maximization

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

Comparing to the benchmark solution (table 4):

TABLE 16
Normal Distribution: Expected profit maximization results

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

6 Conclusion

The algorithm yields excellent results, especially for the expected profit-maximizing, the most important policy in a business context. Tables 14 and 16 show there is not any performance decrease when comparing the proposed solution to the analytical methods.

The novel mechanisms introduced in the GA helped improve performance, as tables 7, 9 and 10 prove. Additionally to these performance results, the author wants to reinforce the time reduction provided by integrating cloud and parallel computing techniques. Like table 6 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 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.

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 [15] and [16] can provide insights regarding this topic.

Improve Fitness Assessment Mechanisms: Section 3.2.3 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 3.2.4 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 [13] 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.

References

[1] T.-M. Choi, *Handbook of Newsvendor Problems. International Series in Operations Research & Management Science.* Springer, 2012, vol. 139.

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

[3] 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.

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

[5] 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.

[6] 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.

[7] 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

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

[9] L. A. Zadeh, "Fuzzy Sets," -, pp. 338–353, 1965.

[10] 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.

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

[12] 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.

[13] 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

[14] 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

[15] 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.

[16] 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.