ERASMUS UNIVERSITY ROTTERDAM

ERASMUS SCHOOL OF ECONOMICS

MASTER THESIS

ECONOMETRICS & MANAGEMENT SCIENCE

with specialisation

OPERATIONS RESEARCH & QUANTITATIVE LOGISTICS

A Fuzzy Multi-Echelon Inventory Control Model with Automatic Knowledge Base Generation

ILIA PIPEROV

481413

Supervisor: Dr. O. Karabağ

Second assessor: Prof. Dr. Ir. R. Dekker

April 15, 2022

ERASMUS UNIVERSITEIT ROTTERDAM

Abstract

Globally optimal control policies for multi-echelon inventory systems are either unknown or immensely difficult to derive. Since the inventories of multi-echelon systems are interconnected and thus rely on each other, a global and centralised policy also requires a perfect exchange of real-time data between all echelons. However, in practice, data often turns out to be vague, imperfectly measured, or incomplete. A very effective tool for handling highly complex systems with imperfect data are fuzzy rule-based systems. In such systems, uncertain variables are modelled by fuzzy sets and control actions are taken based on simple fuzzy if-then rules.

This thesis presents a procedure for automatically deriving fuzzy rule-based systems that are capable of managing general arborescent multi-echelon inventory systems in a reliable and cost-effective way. The procedure uses a modified version of Wang and Mendel's method, simulated annealing, and genetic tuning to determine optimal fuzzy parameters. Moreover, two suitable policies for multi-echelon inventory control are introduced: an exact policy and a heuristic that relies on a partial exchange of data but is considerably faster. The applicability of both policies was assessed by conducting a simulation experiment and a sensitivity analysis. In addition, the heuristic was tested in a real-world application.

As the results demonstrate, fuzzy rule-based systems are an effective and promising tool for controlling multi-echelon inventory systems facing a rapidly changing environment with unreliable data. They lead to almost perfect fill rates while mostly achieving acceptable costs, regardless of the level of uncertainty. This makes them potentially suitable for industries in which the availability of goods is of great importance.

Contents

1 Introduction

Chapman et al. (2016) introduce inventory as usually accounting for about 20% to 60% of the total assets of a manufacturing firm. This makes inventory management a crucial component of increasing a firm's profit (Chapman et al., 2016). At the same time, it is a highly challenging task since globalisation requires companies to establish wide and interconnected supply chains with multi-echelon inventories to meet customer demands (Christopher & Peck, 2004).

In general, optimal control policies for multi-echelon systems are unknown or very difficult to derive (Axsäter, 2015). According to Axsäter (2015), the reason is that, due to the interdependence of all echelons, only centralised decision systems can lead to a globally optimal policy. Such systems, however, are complex and usually require a perfect exchange of real-time data between all echelons (Axsäter, 2015). This also includes that the data which is used for predictions at each echelon is fully reliable (Giannoccaro et al., 2003). As practice shows, however, data often turns out to be vague, imperfectly measured, or incomplete and, therefore, is not always fully reliable (Giannoccaro et al., 2003; Zafari, 2014). This, and also the complexity of global multi-echelon policies in general, causes companies to resort to simple local control policies, which in turn may lead to sub-optimal results (Axsäter, 2015).

A way of tackling the complexity and imperfect data in multi-echelon systems is the use of fuzzy logic as a foundation for decision-taking. The concept of fuzzy logic was originally introduced by L. A. Zadeh (1975) and aims to break down complex decisions into a set of linguistic, easy-to-understand if-then rules. The crucial point is that such rules do not follow the simple Boolean logic, which says that a value is either true or false, but can also adopt continuous truth values in between (e.g., from completely true to completely false) (L. A. Zadeh, 1975). This allows a system to slightly adapt to unexpected changes and, therefore, makes fuzzy logic a valuable tool for inventory control applications.

In this thesis, a new global inventory control policy for general multi-echelon systems is presented. In addition, a heuristic is proposed that only partially considers information along the supply chain but is considerably faster. In this context, general multi-echelon systems are associated with arborescent multi-stage, multi-product supply chains (SCs) consisting of one or more suppliers, plants, warehouses, distribution centers, and customers. The introduced method is capable of handling multiple uncertain input parameters, which are chosen to be: demand, inventory position, lead time, and prices of raw materials. The parameters are modelled by means of fuzzy sets, and model decisions are taken by applying fuzzy logic.

The main purpose of this thesis is to provide an integrated inventory control model that is ready to be applied by practitioners in real-world SCs of any structure and size. Hence, the model is designed to be intuitive, easy-to-implement, and comprehensible for human decision-takers while still achieving comparable or better results than existing local order policies. To guarantee the model's applicability, a procedure to automatically extract an optimal fuzzy knowledge base by using past data is proposed. This thesis combines multiple innovative and widely applied concepts into one completely new general procedure that is broadly applicable in practice. The innovations of the proposed method are further elaborated in Section 2. A more detailed explanation of the method itself can be found in Section 5.

The control model being presented in this thesis was developed by considering the following research questions:

- *Is a fuzzy rule-based inventory control system with an automatically optimised knowledge base applicable to general multi-echelon systems?*

  To determine the applicability of the presented inventory control model, two factors are seen to be defining: 1) lower total costs compared to local order policies and 2) a reasonable computation time. This leads to the following two sub-questions:

  - *Does the application of fuzzy rule-based inventory control systems lead to lower total inventory costs while meeting customer satisfaction requirements compared to a local order policy?*

    By applying simulation, the developed control policies are compared to a local order policy. A local policy is applied which determines batch quantities by using the deterministic EOQ formula first and, based on that, identifies reorder points considering stochastic demand and lead time. Performance metrics of interest are: 1) the total SC cost, (2) the SC fill rate, and (3) the SC holding cost.

  - *Up to what size of general multi-stage supply chains can a general fuzzy rule-based inventory control model be applied such that the training of an automatically generated knowledge base still takes reasonable time?*

    To answer this, the complexity of the proposed algorithms is evaluated. Based on that, upper bounds for the number of inventories to be used in both models are recommended.

- *Is a fuzzy rule-based inventory control system with an automatically optimised knowledge base still applicable to general multi-echelon systems when unexpected uncertainty occurs?*

  The robustness of an inventory control model is essential for its applicability in practice. A robust model is supposed to handle high amplitudes of uncertainty at any given stage of the SC. This results in the following two sub-questions:

  - *To what extent can a fuzzy rule-based control model handle extreme amplitudes of uncertainty at any stage of a general multi-echelon inventory system such that requirements for customer satisfaction are still met?*

    To answer this question, a sensitivity analysis is conducted by applying multiple scenarios, providing different levels of uncertainty at different echelons of the SC.

  - *Under which conditions does a fuzzy rule-based inventory control model for general multi-echelon networks perform better than a local order policy?*

    Similarly, a sensitivity analysis with identical scenarios is applied to a local order policy. By comparing the results to the presented global policy, the conditions necessary for maintaining a high model performance are deduced.

The rest of this thesis is structured as follows: In Section 2, a detailed review of the current literature related to the application of fuzzy logic and evolutionary algorithms in multi-echelon inventory systems is presented. After that, some relevant background knowledge on key concepts related to fuzzy rule-based models, as used in this thesis, is provided in Section 3. It follows a thorough problem description, including all model assumptions in Section 4. In Section 5, the methodology for solving a fictive case study by applying the newly introduced general approach is described. Finally, the results of the case study are displayed in Section 6, and the model's strengths and shortcomings are discussed in Section 7.

2 Literature Review

The optimisation of multi-echelon inventory systems has been extensively researched since the beginning of the last century, when Harris developed the economic order quantity (EOQ) formula in 1913 (Erlenkotter, 1990). With introducing the concept of echelon stock, Clark & Scarf (1960) set a milestone for holistic approaches determining optimal policies for multi-echelon inventory systems. Clark & Scarf (1960) developed a way of obtaining an optimal policy for two-echelon systems by considering inventory information from all stages. Since then, building on this approach, a plethora of methods for solving multi-echelon systems have been invented. Bessler & Veinott Jr (1966), for instance, expanded the method of Clark & Scarf (1960) to the case of general tree-structured inventory systems. Sherbrooke (1968) developed a mathematical model (METRIC) for coordinating base-depot supply chains with repairable items where demand is assumed to originate from a compound Poisson process.

However, as Gümüs & Güneri (2007) found, literature is mostly dedicated to models with a limited number of echelons and, further, mainly deals with serial systems instead of tree-structured ones. Furthermore, the majority of multi-echelon optimisation models only include a small number of variables such as demand or lead time and assume those to originate from known distributions. According to Giannoccaro et al. (2003), probabilistic assumptions usually do not reflect reality properly because they are based on past data, which in many cases is incomplete or not reliable.

To face this ambiguity of data, some researchers resorted to a new way of defining input parameters by using fuzzy sets, a concept that was originally invented by Zadeh (1965). Fuzzy sets allowed to model uncertain variables with linguistic terms, which classify the value of variables by their degree of membership to a chosen function instead of assigning them strict values (Zadeh, 1965). Since the 1980s, the fuzzy set theory has been successfully implemented in inventory problems many times. According to Aengchuan & Phruksaphanrat (2018), most of the related contributions can be categorised into three groups: mathematical, fuzzy logic, and adaptive neuro-fuzzy inference system (ANFIS) models. Mathematical models usually address the classical EOQ model by considering some of the input variables as fuzzy numbers and applying fuzzy arithmetic. Park (1987), for instance, applied the EOQ formula using a fuzzy cost structure. Many other researchers used fuzzy sets to model demand, lead time, deterioration rate, etc. (Aengchuan & Phruksaphanrat, 2018). However, as Aengchuan & Phruksaphanrat (2018) argue, mathematical fuzzy models can be complex and difficult to implement and therefore, may not be applied by industries to real-world applications. ANFIS models, on the other hand, utilise the learning capability of artificial neural networks to define the parameters of fuzzy inference systems (Aengchuan & Phruksaphanrat, 2018). Such models are highly robust (Salleh et al., 2017) and have effectively been applied to inventory control problems, e.g., by Samanta & Al-Araimi (2003) who propose a periodic review model considering fuzzy demand and inventory level. However, as Salleh et al. (2017) argue, they are hardly applicable to systems with large inputs due to their complex structure and high computational cost. Fuzzy logic control models can solve this issue as they use a predefined set of simple if-then rules to decide on actions. This approach is not only easier to execute, but also creates more comprehensible and transparent decisions by mimicking the reasoning of human experts.

A thorough literature research yields that Petrovic & Sweeney (1994) were the first to implement the concept of fuzzy logic in inventory problems. They propose a model for determining order quantities by including fuzzy demand, inventory level, and lead time for a single-echelon system. However, as the authors suggest, the model also has some drawbacks since it does not offer a strict procedure for determining boundaries for the membership functions and also lacks guidance for selecting optimal logical connective definitions. Subsequently, Petrovic et al. (1999) expanded on the previous work and developed a fuzzy control model for serial production supply chains, which also includes a simulation framework for testing the supply chain's performance. The model takes uncertain customer demand and supplier reliability as an input to derive optimal order quantities. After that, Petrovic (2001) refined the model by adding fuzzy lead times. However, it must be emphasised that the proposed method considers a decentralised control respectively a partial coordination of echelons, i.e., inventory information between the echelons is not or only partially included. Giannoccaro et al. (2003) proposed a solution to this potential issue by creating a serial multi-echelon policy which incorporates the echelon stock concept invented by Clark & Scarf (1960). However, contrary to the work of Petrovic (2001), the inventory model provided by Giannoccaro et al. (2003) did not include lead time and supplier reliability as fuzzy parameters, which may result in a wrong estimation of delays along the supply chain and consequently could amplify the bullwhip effect (Xiong & Helo, 2006). To solve this issue, Xiong & Helo (2006) developed a fuzzy inventory control model to counteract demand fluctuation resulting from the bullwhip effect. In their work, they provide a general framework for applying fuzzy control in general multi-echelon supply-demand networks by enabling information sharing along the whole chain.

However, the model of Xiong & Helo (2006) was found to be the last approach aiming to generalise the use of fuzzy logic in multi-echelon inventory systems. In more recent literature, researchers instead rely on the further development of fuzzy single-echelon control systems by incorporating new combinations of uncertain parameters. Tanthatemee & Phruksaphanrat (2012), for instance, considered uncertain demand and availability of supply to find both optimal order quantities and optimal reorder points.

All the previously mentioned fuzzy inventory control models have one thing in common: the model's knowledge base is either established by expert knowledge or derived from simple observations of past data. Even though these approaches prove to produce acceptable results for single-echelon systems including few fuzzy parameters, problems could arise for systems with higher complexity. One reason is that the number of fuzzy rules increases exponentially with the number of included input parameters (Cordón, 2011). Hence, it becomes increasingly difficult for human experts to capture the system's complexity when determining a rule base. Also, the risk of having irrelevant rules increases, which consequently reduces a model's comprehensibility (Cordón, 2011).

A concept widely used to automatically create efficient fuzzy models is genetic tuning. Genetic tuning takes a predefined structure of membership functions and rules and adapts its parameters optimally by using genetic algorithms (Cordón, 2011). Even though this concept has been broadly applied to fuzzy logic-based models in general, its usage in the field of inventory management is not well-established to this point. Rotshtein & Rakityanskaya (2006) were found to be the first to apply genetic tuning to a single-echelon fuzzy inventory control model. The authors use both the trend of demand and the current inventory level as fuzzy inputs to determine how much to increase or decrease the level of inventory in each period. To achieve this, they suggest a tuning procedure for finding an optimal knowledge base that solves a non-linear problem by applying a genetic algorithm combined with a neural network. Wiecek (2016) uses a genetic algorithm for optimising the rule base of a fuzzy model which predicts order quantities and reorder points considering uncertain demand and lead time in a single-echelon inventory.

Based on the fuzzy logic inventory control models mentioned in this review, no method could be found that incorporates uncertain demand, lead time, and commodity prices while considering information between echelons along the whole supply chain. Furthermore, no fuzzy-logic based control model for general multi-echelon systems could be found that offers an automatic knowledge base generation procedure and therefore makes an application to real-world supply chains possible. Thus, to the best of our knowledge, the method presented in this thesis is the first general fuzzy logic-based inventory control model that treats demand, lead time, and commodity prices at the same time and also provides an automatic optimisation of the fuzzy model's parameters.

3 Background Knowledge on Fuzzy Logic Models

A key aspect of developing efficient inventory optimisation policies in general is the way uncertainty is taken into account. Uncertainty typically arises when parameters such as demand, lead time, or costs are estimated in order to determine optimal order quantities and reorder points. Typically, uncertainty is considered by assuming uncertain model parameters to originate from probabilistic distributions that are derived from past data (Giannoccaro et al., 2003). It is important to realise that such an approach usually presumes the data to be perfect. However, as Giannoccaro et al. (2003) point out, practice shows that there can be various other reasons for uncertainty related to the quality of data that are difficult to capture. Data can be ambiguous, vague, imperfectly measured, or even missing, and therefore, is not always fully reliable (Giannoccaro et al., 2003; Zafari, 2014). Two interrelated concepts which make it possible to cope with imprecise and ambiguous parameters in inventory applications are fuzzy sets and fuzzy logic. Together, they are applied in so-called fuzzy inference systems (FIS) to transform vague input parameters into precise (crisp) outputs.

This section gives an introduction to fuzzy sets, fuzzy logic, and fuzzy inference systems. Note that only key concepts relevant to this thesis are presented. A complete and thorough collection of knowledge about fuzzy sets, fuzzy logic, and fuzzy systems is offered in L.-X. Wang (1999).

3.1 Fuzzy sets

Let us consider the decision of when to place a new order to stock up the inventory of a certain product. Fundamentally, a new order may be set when the inventory level is low but not too low such that no stock out can occur while costs are kept low. Apparently, such a definition is not precise enough since the question arises as to how exactly ”low” and ”too low” are defined in this context. A common way to objectively specify such criteria would be to determine an exact reorder point by considering other factors potentially affecting a stock out, such as the future demand or the expected lead time. Hence, for each point in time, it is precisely defined whether to place a new order or not, i.e., the vague criterion of ”low but not too low” is replaced by a specific reorder point. However, such an exact criterion only makes sense if one can fully rely on the correct computation of a reorder point. This not only requires the right mathematical method but also assumes correct data. In particular, the latter is not always given in practice. As a consequence, the definition of a reorder point is not exact anymore but has become vague.

A mathematical tool for dealing with vague expressions provides the concept of fuzzy sets, which was first introduced by Zadeh (1965). The author formalises the definition of a fuzzy set as:

Definition 1. *A fuzzy set (class) A in X is characterised by a membership (characteristic) function $\mu_A(x)$ which associates with each point in X a real number in the interval [0,1].*

Hence, an element is no longer strictly a member of a class or not, but can have a continuous degree of membership to a class as long as the class is defined as a fuzzy set. This also implies that an element can have partial memberships, i.e., it can be part of multiple fuzzy sets at the same time (Chen et al., 2001). This property makes fuzzy sets perfectly suitable for classifying vague and imprecise inputs, since these might not be uniquely assignable to a certain class of interest.

As used by Wiecek (2016), a formal mathematical representation of Definition 1 is given by:

$$A = \{(x, \mu_A(x)) : x \in X\}, \tag{1}$$

where:

$$\mu_A : X \to [0, 1].$$

Hence, for each element $x \in X$ (non-empty space), a degree of membership is assigned to the fuzzy set A, i.e., the fuzzy collection A in a non-empty space X is a set of of pairs as given in (1) (Wiecek, 2016). Using representation (1), one can distinguish between the following three cases:

Case 1: $\mu_A(x) = 1$ (an element x is a full member of fuzzy set A ($x \in A$))

Case 2: $\mu_A(x) = 0$ (an element x is no member of fuzzy set A ($x \notin A$))

Case 3: $0 < \mu_A(x) < 1$ (an element x is partially a member of fuzzy set A)

Returning to our example of finding the right reorder level for a certain product, a fuzzy set could be used to represent the set of low inventory levels. For instance, low inventory levels could be defined as all possible stock quantities in the range of 0 and 100 (universe of discourse). Hence, stock levels that do not belong to the universe of discourse are clearly categorised as not being low. Whereas, all other values are classified as low with a certain degree. To fully define this degree, the shape of a membership function must also be specified. As shown in Figure 1, this function may be linear for low inventory levels, but essentially can be of any arbitrary shape (Sadollah, 2018). Other commonly used functions are triangular, trapezoidal, or Gaussian (Sadollah, 2018).

Initially, the parameters of a membership function were supposed to be designed by experts (Giannoccaro et al., 2003). Based on their experience, managers are asked to decide on suitable shapes for membership functions and, further, define lower and upper limits for given fuzzy (sub)sets. In particular, such limits may be set by estimating which values are not possible for a certain variable of interest (Giannoccaro et al., 2003). However, to this date, various methods for automatically generating and tuning the parameters of membership functions have been invented. For instance, in this thesis, genetic tuning is applied to a data sample in order to determine the parameters of membership functions. Further details can be found in Section 5.

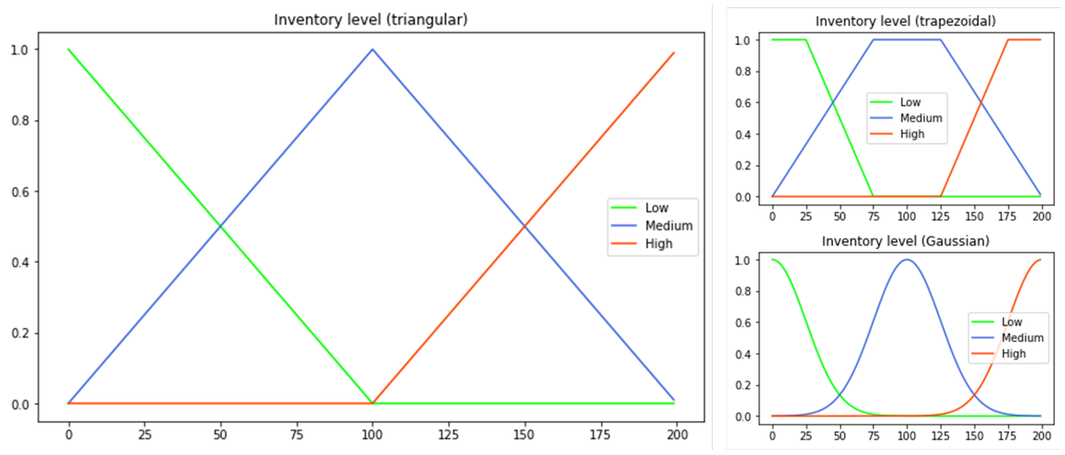


Figure 1: Common shapes of fuzzy membership functions

3.2 Fuzzy logic

Real-world applications often require humans and corporations to take rational decisions in an environment of uncertainty and imprecision (Chen et al., 2001). It is known that experienced humans have a strong capability of making accurate decisions while facing problems with vague information (L. A. Zadeh, 1988). Certainly, this is only achievable up to a certain complexity. Therefore, humans typically resort to computer-supported decision systems that are normally based on classical two-valued logic, as in the form of: IF x_1 is true AND x_2 is false AND ... AND x_n is false THEN y is false (Chen et al., 2001). That is, only propositions that are either true or false are used. However, as L. A. Zadeh (1973) argues, so-called two-valued logic becomes less sufficient in capturing the complexity of a vague and imprecise real-world system as its complexity rises. The author described this phenomenon as the "principle of incompatibility".

3.2.1 The concept of approximate reasoning

Fuzzy logic, a new type of logic introduced by L. A. Zadeh (1975), provides a framework for approximating human reasoning more accurately than two-valued logic. In fuzzy logic, truth-values are fuzzy subsets of the unit interval with linguistic labels such as "true", "false", "very true", "quite true", "not very true", "not very false", etc. Additionally, the use of fuzzy quantifiers such as "most", "many", "several", "few", "much of", "frequently", "occasionally" or "about" is permitted (L. A. Zadeh, 1988). Consequently, truth tables and rules of inference do not provide exact values anymore, but produce imprecise linguistic expressions as an approximation instead. The degree of approximation depends on the definitions of the fuzzy subsets representing the linguistic labels, that is, their membership functions and universe of discourse (L. A. Zadeh, 1975).

To be able to cope with subjective linguistic expressions in a mathematical context, L. A. Zadeh (1975) developed an extensive set of mathematical tools. A thorough summary of all mathematical innovations, including fuzzy syllogisms and basic rules of inference, is given in L. A. Zadeh (1988).

3.2.2 Meaning representation

A way of understanding the implications of fuzzy sets as a representation of truth-values provides the possibility theory (L. A. Zadeh, 1988). Concretely, it holds that

$$Poss\{X = u\} = \mu_A(u), u \in U, \tag{2}$$

where μ_A is the membership function of A and $Poss\{X = u\}$ is the possibility that X may take u as its value. Hence, the possibility distribution of X is the set of possible values of X, which implies that possibility is a matter of degree (L. A. Zadeh, 1988). It should be emphasised that possibility is not equatable to probability in this context. In fact, possibilities represent similarities of objects to imprecisely defined properties, whereas probabilities convey information about relative frequencies (Bezdek, 1994; Giannoccaro et al., 2003). However, there exist methods to transform possibility distributions into probability distributions (Giannoccaro et al., 2003). Further information about this topic can be found in Dubois et al. (1993).

3.2.3 Linguistic variables

Another crucial concept in fuzzy logic is the linguistic variable. In general, linguistic variables are words or sentences that are represented by fuzzy membership functions (possibility distributions)

(L. A. Zadeh, 1988). For instance, a linguistic variable may be ”inventory level” with its values (labels) ”low”, ”medium”, or ”high” as already introduced in our example. In general, the number of values can be chosen arbitrarily, but it is common to use an uneven number of values in order to ensure a symmetric distribution of values around a center value (Cordón, 2011; Cordón et al., 2001). A common use case of linguistic variables are control systems which can be modelled by fuzzy rule-based systems as shown in Section 3.3. In such systems, an output variable Y (controllable variable) is controlled as a function of different state variables (input variables) X_1, X_2,...,X_n (L. A. Zadeh, 1988) by applying fuzzy if-then rules (Section 3.2.4). Both input and output variables are typically represented by linguistic variables (L. A. Zadeh, 1988). For instance, in addition to the linguistic variable ”inventory level” (X_1) in our example, a linguistic variable ”demand” (X_2) with values ”very low”, ”low”, ”medium”, ”high”, and ”very high” may be used to control how much to order — ”order quantity” (Y_1) — with e.g., values ”small”, ”medium”, and ”large”.

3.2.4 Fuzzy if-then rules

Fuzzy if-then rules are rules whose antecedents, consequences, or both are fuzzy rather than crisp (Dubois & Prade, 1996). Typically, they express the relationship between one or more linguistic variables to each other (Zafari, 2014). A basic structure is given by (Zafari, 2014):

IF <antecedent> THEN <consequence>

In general, there are several different types of fuzzy rules (Dubois & Prade, 1996). In this thesis, Mamdani-type rules (Mamdani & Assilian, 1975) are applied. Their most usual structure is given as (Cordón, 2011):

IF X_1 is A_1 and ... and X_n is A_n THEN Y is B

In Mamdani-type rules, both antecedents and consequences are linguistic variables (Cordón, 2011). They are used in this thesis since their simple structure enables the design of accurate and highly interpretable fuzzy rule-based systems (Section 3.3).

3.3 Mamdani-type fuzzy rule-based systems (FRBSs)

As previously shown, one of the main advantages of fuzzy logic is its ability to approximate human reasoning by applying fuzzy if-then rules. However, most real-world applications have crisp input values and, furthermore, require crisp output values to clearly define actions that need to be taken

(K. Wang, 2001). This can be guaranteed by so-called fuzzy rule-based systems (FRBS). FRBSs are used to model systems making use of fuzzy logic with fuzzy predicates (fuzzy rules) (Cordón, 2011) while accepting crisp input values and producing crisp outputs. As shown in Figure 2, a FRBS consists of two interrelated components (Cordón, 2011): (1) a fuzzy inference system (FIS) and (2) a knowledge base (KB). Both components are explained in the following.

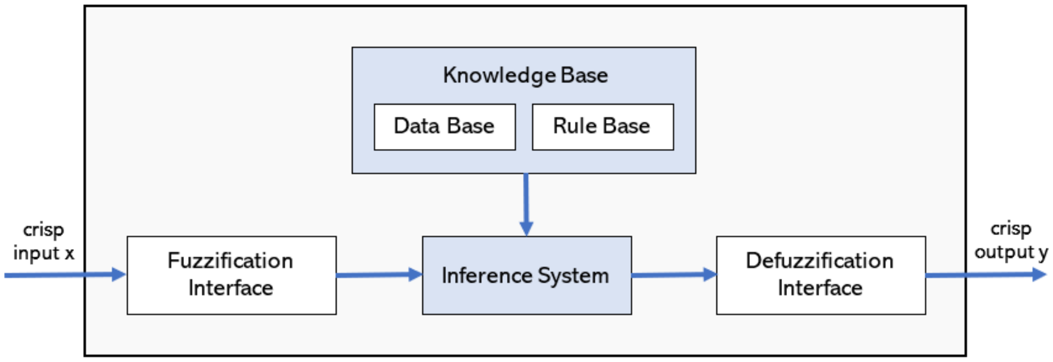


Figure 2: Framework of a Mamdani-type FRBS. Adapted from Cordón (2011).

3.3.1 Fuzzy knowledge base (KB)

The KB of a Mamdani-type FRBS is a collection of fuzzy if-then rules (Section 3.2.4) which qualitatively describe a system (Cordón, 2011). In Mamdani-type FRBSs, the fuzzy sets describing the linguistic variables being part of the rules are uniformly defined for all the rules in the KB (Cordón, 2011), i.e., for a particular fuzzy set, the same membership function is applied to all the rules of a KB. This improves the readability and thus the interpretability of a system for human beings (Cordón, 2011). Typically, a KB is further divided into two sub-components: (1) the fuzzy rule base (RB), which contains all fuzzy rules; and (2) the data base (DB), including the membership functions of the fuzzy sets describing the linguistic variables being part of the rules (Cordón, 2011).

In general, the membership functions and rules of a KB are designed by experts (Chen et al., 2001). However, in recent decades, methods for automatically determining the components of a KB based on past data have been developed. Usually, genetic algorithms are utilised to both optimise the parameters of membership functions and extract the most relevant rules from a predefined KB (Riza et al., 2015; Cordón, 2011). More information on genetic algorithm-based Mamdani FRBSs can be found in Cordón (2011). On the other hand, an initial KB can be determined by employing clustering. A particular example of an automatic procedure, including clustering and genetic tuning, is provided in the methodology of this thesis (Section 5).

3.3.2 Fuzzy inference system (FIS)

A FIS is the central unit of a FRBS. It is responsible for transforming crisp input values into crisp output values by applying all the information stored in the fuzzy KB (Cordón, 2011). A FIS consists of three main steps: (1) fuzzification, (2) inference, and (3) defuzzification, which are further explained by using Figure 3. In Figure 3, a typical Mamdani-type FIS is displayed. Two linguistic variables are considered to be the system's input: "A" and "B". Furthermore, a linguistic variable "C" is used to model the system's output. Each of the three variables are described by three values (A_1, A_2, A_3; B_1, B_2, B_3; C_1, C_2, C_3), which are represented by fuzzy subsets with triangular membership functions. Finally, there exist n different fuzzy if-then rules (RB) describing the relations of the linguistic variables to each other. In the following, the particular steps of the process are explained.

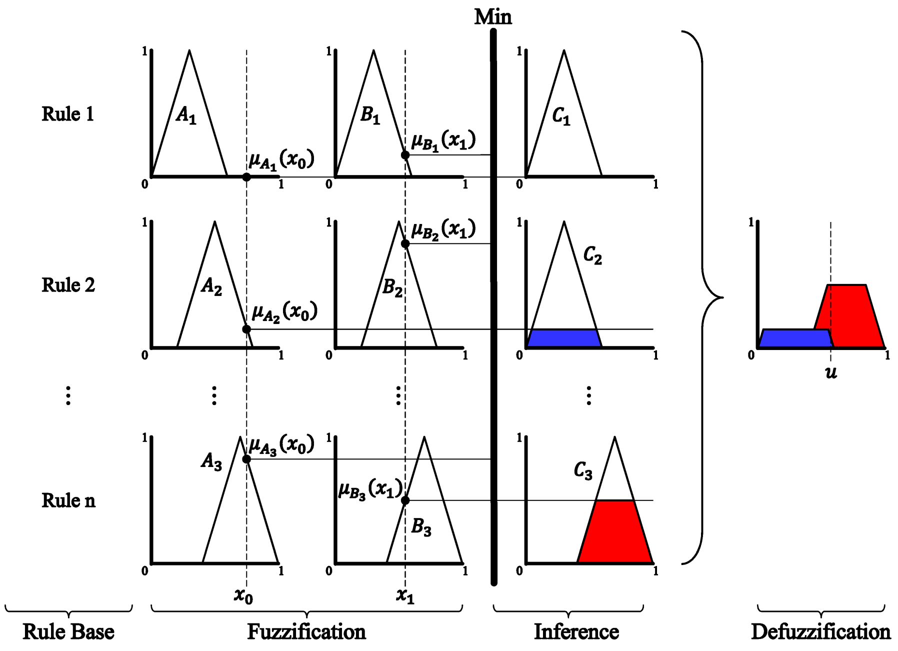


Figure 3: Steps of a typical Mamdani-type FIS. Adapted from Salgado-Plasencia et al. (2020).

Fuzzification

Fuzzification is the transformation of a crisp input value into a fuzzy input by applying the information in the KB (Kayacan & Khanesar, 2015). Concretely, given an input (x_0, x_1 in Figure 3), all membership functions related to input variables are evaluated for all rules of the RB, as shown in (1). The "fuzzified" input values ($\mu_{A_1}(x_0)$, $\mu_{A_2}(x_0)$, $\mu_{A_3}(x_0)$, $\mu_{B_1}(x_1)$, $\mu_{B_2}(x_1)$), $\mu_{B_3}(x_1)$ in

Figure 3) are further used to derive decisions by conducting inference, as shown in the next step.

Inference

Inference is the fuzzy logic reasoning process that determines the outputs corresponding to fuzzified inputs (K. Wang, 2001). Logical conclusions in a FIS are thereby determined by fuzzy relations between the fuzzy sets that are part of the RB. Fuzzy relations are typically fuzzy expressions that are connected by logical operators (K. Wang, 2001). Three commonly used operators in fuzzy set theory are shown in Table 1. Note that A and B are fuzzy sets with respective membership functions $\mu_A(x)$ and $\mu_B(x)$.

Description	**Operator**	**Fuzzy operation**
Union of A and B	OR	$\mu_A(x) + \mu_B(x) = \max[\mu_A(x), \mu_B(x)] = \mu_A \vee \mu_B$
Intersection of A and B	AND	$\mu_A(x) \cdot \mu_B(x) = \min[\mu_A(x), \mu_B(x)] = \mu_A \wedge \mu_B$
Complement of A	NOT	$\neg\mu_A(x) = 1 - \mu_A(x)$

Table 1: Basic logical operations with fuzzy sets. The definitions origin from Zadeh (1965).

A widely used method for conducting inference in Mamdani-type FRBSs is the max-min composition (Chen et al., 2001). In this method, all rules of the RB are evaluated by applying the logical operations as shown in Table 1. To be exact, the results of the fuzzy operations on the rule antecedents are mapped into the fuzzy subsets representing the rule consequences. The result of the inference is then the area under the mapped values, as marked in red and blue in Figure 3.

Defuzzification

It is important to realise that the output of inference is a linguistic variable, that is, a fuzzy set. Hence, the fuzzy output must be converted to a crisp value in a final step called defuzzification (Aengchuan & Phruksaphanrat, 2018). In the process of defuzzification, first an aggregate membership function is developed by taking the union of the outputs of each rule. This means that the overlapping area of the fuzzy output set is counted once (Zalnezhad & Sarhan, 2015). After that, a crisp value is computed by applying a defuzzification method. A commonly used method is the centroid method, which takes the center of area (marked as "u" in Figure 3) of the aggregate fuzzy set obtained from the first step (K. Wang, 2001).

4 Problem Description

This section defines the problem being addressed in this thesis. First, the domain of application of the proposed inventory policy is clearly characterised by introducing the network and its underlying model assumptions. Finally, the objectives and fuzzy parameters of the presented FRBS are discussed.

4.1 Network description

In this thesis, a cost-effective fuzzy rule-based inventory control model for general arborescent multi-echelon inventory systems is developed. The main objective is to automatically generate fuzzy KBs for each inventory that minimise the total SC costs in the network while maintaining a reasonable fill rate. Hence, there are two opposed optimisation factors — costs and fill rate — that are considered. This is essential since solely focusing on minimising the costs may lead to lower customer satisfaction, which in turn may negatively influence a company's success in the long run.

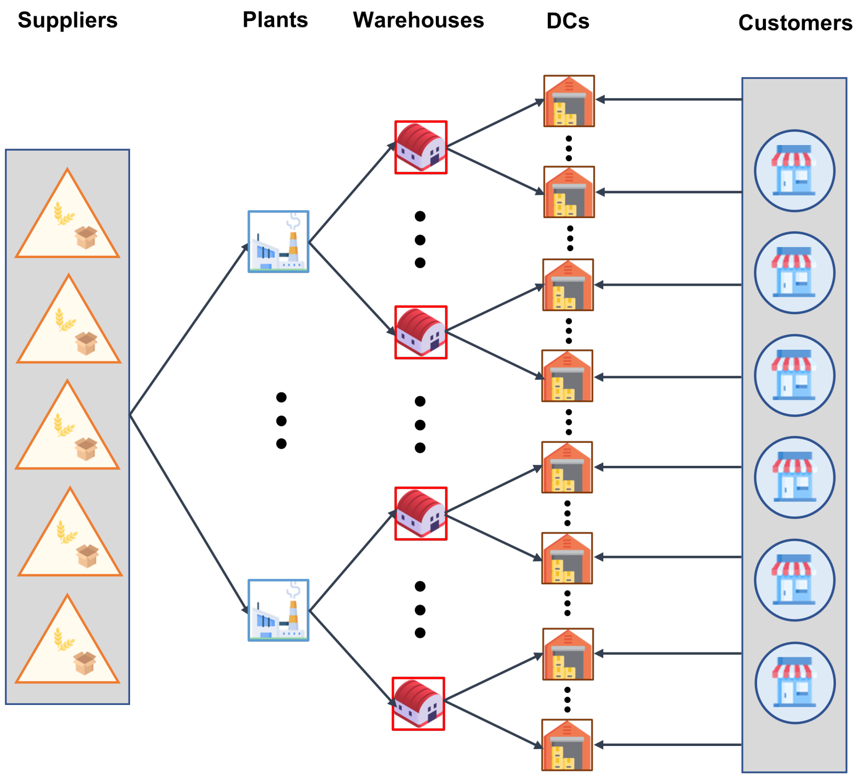


Figure 4: SC with general arborescent inventory structure

As shown in Figure 4, the considered systems are associated with multi-stage supply chains consisting of the following stages: (1) suppliers (S) — responsible for the replenishment of raw materials and sub-components that are needed in production; (2) plants (P) — the company's production points at which end-products are assembled; (3) warehouses (WH) — store end-products that are newly produced (assembled); (4) distribution centers (DC) — regional depots that store end-products waiting for final transport to the end-customers; and (5) customers (C) — e.g., hypermarkets, grocery chains, or individual grocers. Since the system is arborescent, installation points have at most one single predecessor (Axsäter, 2015). Moreover, as shown, each echelon can consist of multiple installations and, further, is defined as a stocking point meaning that it supplies lower echelons and gets supplied by higher echelons. There is no limit to the number of installation points per echelon (stage). This type of SCs is not uncommon for companies that manufacture and distribute consumer goods, e.g., a producer of packaged cereals, as used in the case study of this thesis. Therefore, the presented model can be of high importance for real-world applications.

4.2 Model assumptions

Plants in the production stage can potentially produce multiple types of end-products, which again may consist of multiple sub-components (raw materials) that are directly sourced from the raw material market (suppliers). It is assumed that plants have limitless access to raw materials and, further, have unlimited production capacity. Moreover, all inventories of the SC are assumed to have unlimited storage capacity. This is an assumption usually made in fuzzy rule-based inventory control applications since storage capacity constraints would overcomplicate the derivation of proper fuzzy rules and membership functions of the output variable.

Customer demand is uncertain and can only be satisfied directly from the distribution center inventories. Unfulfilled customer demand is considered to be lost, i.e., back-ordering is not allowed. Every time an order cannot be satisfied, a fixed penalty cost is incurred. It is also assumed that moving goods between echelons of the SC does not happen instantly, i.e., there is a lead time between all echelons. Moreover, suppliers are assumed to be fully reliable, and market prices of raw materials are not constant. Lead time and spot market prices of raw materials are assumed to be uncertain. As the type of distribution is not relevant for the derivation of fuzzy rule-based models, lead time and market prices can follow any distribution (general distribution).

Supply chain operating costs play a crucial role when determining a cost-effective global order policy. In the proposed FRBS, costs arising between the suppliers and the customers are assumed to be known in advance, i.e., they have been determined beforehand. The particular cost components are thoroughly explained by using the cost function in the next section (4.3).

It is assumed that lateral transshipments are not allowed. This is essential since the existence of multiple predecessors of an inventory would significantly increase the complexity of the model. In such a case, a particular inventory would require the determination of multiple order quantities at once. This, in turn, would require the execution of fuzzy multi-input multi-output (MIMO) models. However, MIMO models are seen to be outside the scope of this thesis.

Finally, it is assumed that inventories of each echelon are replenished by applying periodic review policies on a daily basis, i.e., at the beginning of each day, the required order quantity for each inventory is determined. Since it is decided whether an order must be placed every day, the model does not require the determination of a reorder level.

4.3 Model objectives

As already introduced, the proposed multi-echelon FRBS has two opposed objectives: (1) minimising the total SC costs arising throughout the entire network while (2) maintaining a reasonable fill rate. The model must thereby process information along the whole chain and, furthermore, it needs to provide reasonable computation times.

Objective (1) is measured by the total SC costs C which are given by

$$C = C_H + C_L + C_O + C_S + C_F + C_P + C_U. \tag{3}$$

As shown in Equation (3), the cost of operating a supply chain (C) includes holding costs (C_H), penalty costs (C_L), fixed order costs (C_O), production setup costs in case of plants (C_S), transport costs between echelons (C_F), production costs (C_P), and purchase costs of raw materials (C_U). The calculation of the separate components is displayed in equations (4) to (8).

$$C_H = \sum_{t \in T} \sum_{v \in V} I_v(t) h_v \tag{4}$$

The total holding costs C_H are obtained by adding the holding costs of each inventory $v \in V$ over the entire time horizon $T = \{t \in T : t = 1, , , , T\}$. The total holding cost per day t at inventory v is calculated by multiplying the corresponding holding cost h_v (per unit and day) by the inventory level $I_v(t)$ on this day. The holding cost h_v is defined as a fixed percentage of the average unit cost of material v_p — the product (material or end-product) corresponding to inventory v.

$$C_L = \sum_{o \in O : o_k = CUST} (1 - u_o) b_{o_v} \tag{5}$$

A fixed penalty cost b_{o_v} is incurred for each unfulfilled customer order (where o_v is the inventory (v) linked to a particular order o). The binary variable u_o indicates whether an order $o \in O$ (where O is the set of all orders in the network) is satisfied. Moreover, the parameter o_k specifies from which type of echelon (k) an order o was placed. Concretely, $k \in K = \{P, WH, DC, CUST\}$, where K is the set of the system's echelon types, which includes: plant (P), warehouse (WH), distribution center (DC), and customer (CUST).

$$C_O + C_S = \sum_{o \in O : o_k \in \{P, DC\}} c_{o_v} + \sum_{o \in O : o_k \in \{WH\}} s_{o_v} \tag{6}$$

A fixed order cost c_{o_v} (c_v) is incurred for each order o that comes from a plant (P) or a distribution center (DC). Similarly, a fixed production setup cost c_{o_v} must be paid for each order placed by a warehouse (WH).

$$C_F = \sum_{t \in T} \sum_{v \in V} \sum_{w \in S_v} D_{vw}(t) f_{vw} \tag{7}$$

When an inventory v fulfills an order of one of its successors $w \in S_v$ (set of successors of v), transport costs of f_{vw} per unit of demand are incurred. The parameter $D_{vw}(t)$ denotes the demand from inventory w met by inventory v on the date t.

$$C_P + C_U = \sum_{o \in O : o_k = WH} q_o p_{o_v} + \sum_{o \in O : o_k = P} q_o m_{o_t} \tag{8}$$

Finally, there must be paid a production cost p_{ov} and a material cost m_{ot} (material price at time t) per unit for each order o set by a warehouse inventory $(o_k = WH)$ or a plant inventory $(o_k = P)$. Hence, these costs are multiplied by the order quantity q_o of order o.

Objective (2) is evaluated by using the fill rate

$$FR = \frac{\sum\limits_{o \in O : o_k = CUST} u_o}{\sum\limits_{o \in O : o_k = CUST} 1}, \tag{9}$$

which is defined as the fraction of satisfied customer orders and thus is aimed to be high as it directly translates to overall customer satisfaction.

Provided the total SC costs, the overall performance of the FRBS is primarily evaluated by the three metrics: (1) total SC cost, (2) SC fill rate, and (3) SC holding cost. Metrics (1) and (2) are given by Equations (3) respectively (4), whereas the SC fill rate is computed as in (9).

4.4 Fuzzy parameters

A crucial aspect in the development of cost-effective multi-echelon inventory control policies is the way uncertainty is taken into account. The proposed general multi-echelon FRBS combines and processes multiple uncertain input parameters in order to determine optimal order quantities for each installation point in each order period. The following four fuzzy input parameters are considered in the proposed models: (1) demand, (2) current inventory position, (3) expected lead time, and (4) market price, to finally obtain one output value — (5) order quantity — which again is represented by fuzzy sets.

Demand is based on incoming orders from end-customers being placed at echelon 1. These orders, in turn, cause internal demand along the SC's remaining echelons in a sequential order. However, only demand arising at distribution centers, warehouses, and plants is of relevance to the fuzzy system since the inventories of material suppliers are external.

Lead time is defined as the time between the placement of an order and its delivery (Eiselt & Sandblom, 2012). At the production level, it is the time between the start and completion of a production process. Lead time inputs are based on forecasts that are derived from past data. For instance, in the case study of this thesis, the lead time is estimated by the sample average of all lead times that have occurred up to the date of interest.

In order to evaluate the availability of goods, it is crucial to consider the inventory position instead of the on-hand inventory. This is due to the existence of lead times between all echelons. In particular, considering the on-hand inventory, an order may be set more than once if a previously placed order has not arrived yet.

In general, the market prices of raw materials are not constant. As Moheb-Alizadeh & Handfield (2018) argue, in fact, they can be highly volatile due to a variety of uncontrollable external macroeconomic factors such as inflation, interest rates, or industrial production. However, even small changes in prices can have a considerable effect on a company's success provided the price of end-products is highly dependent on the prices of raw material (Moheb-Alizadeh & Handfield, 2018). Hence, the market price of raw materials is seen to be a crucial element in the process of finding a globally optimal control policy.

5 Methodology

In this chapter, a method for the derivation of a fuzzy multi-echelon inventory control model with automatic knowledge base generation is suggested. First, an outline of the method's core element — the 3-step KB generation procedure — is presented. Following this, details about each of its three steps are provided. Finally, it is shown how the procedure is applied in the context of global multi-echelon inventory control by providing a newly introduced formulation of a global multi-echelon FRBS, including a heuristic approach.

5.1 Outline of 3-step KB generation procedure

The procedure of defining an optimal KB for a particular inventory consists of three consecutive steps. The main goal is to establish fuzzy rules and membership functions (MFs) that minimise the total operating inventory costs while a reasonable fill rate is maintained.

As shown in Figure 5, the proposed method requires data that includes information about all fuzzy input and output variables for each given review period of the observed time horizon. In general, the procedure can be applied to any set of input variables as long as order quantity remains the only output variable. For the purpose of testing, it is recommended to initially divide the data into a training and testing part. Hence, the proposed 3-step procedure is first applied to a training data set and then tested and compared to other inventory policies using the remaining test data.

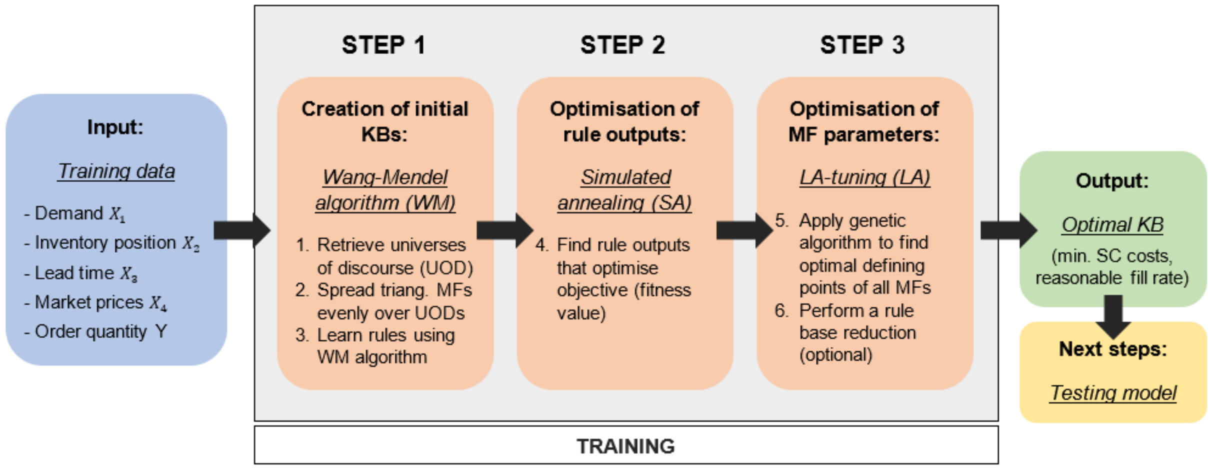

STEP 1
STEP 2
STEP 3
Input:
Training data
- Demand X_1
- Inventory position X_2
- Lead time X_3
- Market prices X_4
- Order quantity Y
Creation of initial KBs:
Wang-Mendel algorithm (WM)
1. Retrieve universes of discourse (UOD)
2. Spread triang. MFs evenly over UODs
3. Learn rules using WM algorithm
Optimisation of rule outputs:
Simulated annealing (SA)
4. Find rule outputs that optimise objective (fitness value)
Optimisation of MF parameters:
LA-tuning (LA)
5. Apply genetic algorithm to find optimal defining points of all MFs
6. Perform a rule base reduction (optional)
Output:
Optimal KB
(min. SC costs, reasonable fill rate)
Next steps:
Testing model
TRAINING

Figure 5: 3-step KB generation procedure

Studies have shown that the information contained in the DB, that is, the definition of the

MFs and the number of values per fuzzy variable, has a much larger impact on a fuzzy system's performance than the structure of the RB itself (Cordón et al., 2001; Bonissone et al., 1996; Cordón et al., 2000; Zheng, 1992). Therefore, the main focus of the procedure is on step 3 (see Figure 5), that is, the tuning of the definition points of triangular MFs by applying a genetic algorithm called LA-tuning. This method was invented by Alcalá et al. (2007) and makes use of the linguistic 3-tuples representation of triangular MFs. As opposed to the classical tuning methods of triangular MFs, which use all three definition points (left, top, right), the LA-tuning algorithm limits the search space considerably by including only two points: the lateral displacement and the amplitude variation of a label's support. A complete description of the LA-tuning is provided in Section 5.4.

Like most of the genetic tuning methods, the LA-tuning method also requires an initial KB as a starting point for the optimisation. As mentioned before, an initial KB is usually defined by expert knowledge. However, as will be shown in later sections, a global multi-echelon FRBS requires a large number of fuzzy variables and thus leads to high system complexity. Therefore, in order to ensure the applicability of the model to practical use cases, an initial KB is created automatically by analysing past data. As the LA-tuning examines an immensely high search space in this application, the step of defining an initial KB is crucial for the quality of the model. That is because the better the initial solution is, the faster and more productive the genetic algorithm will search through the search space. Hence, the quality of the initial KB directly influences the computation time as well as the objective value.

The design of a good initial KB is accomplished in two consecutive steps. First, a starting KB is defined by directly learning from examples, that is, from existing information on inputs and outputs. For this purpose, a simple and widely known approach — the Wang and Mendel method (WM) — is applied. This method, which was originally developed by L.-X. Wang & Mendel (1992), is an ad-hoc data covering RB generation process. Concisely, the method aims to extract unique and highly contributing rules such that all input-output pairs from the past are covered. A thorough description of the WM algorithm can be found in Section 5.2.

One shortcoming of automatic KB generation methods for inventory applications is that, in general, they assume that the "correct" (optimal) output is among the observed values from the past. However, in practice, it is not guaranteed that a previously applied order policy is optimal or even near-optimal, especially considering the complexity that is faced in multi-echelon systems. Therefore, we introduce an an additional second step — the optimisation of the rule outputs by using simulated annealing (SA), an algorithm invented by Kirkpatrick et al. (1983). The main idea is to find RBs that lead to the best objective value by iteratively exploring the neighbourhood of "good" solutions. In this context, "good" solutions stand for rule outputs that lead to better or not significantly worse objective values. The method of SA is considered suitable since it can handle arbitrary systems and cost functions (Ledesma et al., 2008), e.g., functions that are not continuous, as faced in fuzzy systems. Moreover, we assume that even though companies may not follow optimal order policies in practice, such policies still lead to reasonable results, i.e., are not arbitrarily worse than optimal policies. Hence, the optimal RB is expected to lie in the close neighbourhoods of the initial RB that is derived from the given order policy. In such a case, SA also offers benefits because of its ability to explore neighbourhoods quickly without easily getting stuck in local optima. A detailed description of the SA algorithm can be found in Section 5.3.

5.2 Creation of initial KB: Wang and Mendel method (WM)

The main idea of the WM method (L.-X. Wang & Mendel, 1992) is to generate initial fuzzy rules of a RB by considering input-output pairs that have been observed in the past. The crucial point is that the newly created rules must cover the entire data that is known. This can be achieved by a process that consists of three steps, which are further explained by using Figure 6. In Figure 6, a fuzzy partitioning of the input and output space of a simplified system consisting of two fuzzy inputs (x_1, x_2) and one fuzzy output y (order quantity) is shown. The blue triangular functions represent the MFs describing both fuzzy input variables, whereas the blue dots signify the observed data. There is an output value linked to each data point, which is expressed by a number (numerical value) and its corresponding fuzzy (linguistic) value. Finally, the dashed lines demonstrate the fuzzification of the input data.

Step 1: Fuzzy partitioning of the input variable space

Let us define the variable input space as the area that is covered by the universes of discourse of the fuzzy input variables. A fuzzy partitioning of this area is obtained by dividing the entire input variable space into smaller subspaces (partitions) that cover the intersection area of n (number of input variables) particular fuzzy sets within the universe of discourse. This results in a grid with $\prod_{i=1}^{n} m_i$ fields, where m_i is the number of linguistic labels related to variable i. An initial definition of the DB can be retrieved by experts or by a so-called normalisation process (Cordón et al., 2001). A process of this type divides a universe of discourse into equal or unequal parts and assigns a specific type of MF to each of these parts (Cordón et al., 2001; L.-X. Wang & Mendel, 1992).

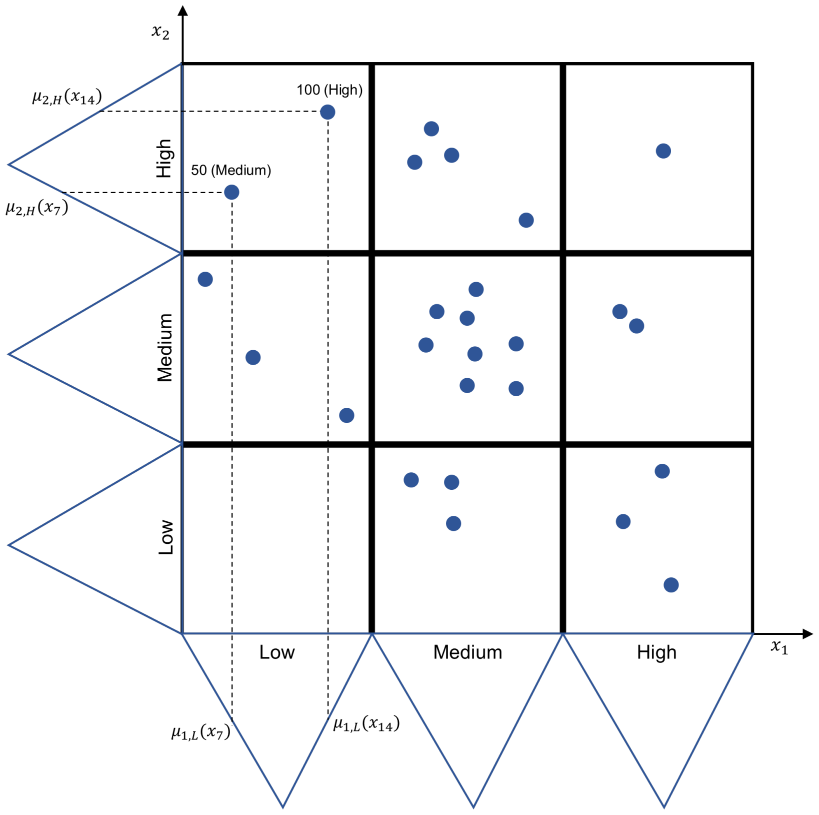


Figure 6: Fuzzy partitioning of input variable space

Step 2: Generation of preliminary linguistic rule set

In this step, an initial RB is created by assigning a rule to each input-data pair of the input-output data set. The best covering approach is used to translate an $n+1$ dimensional real array (n input, 1 output values) into an $n+1$ dimensional array containing linguistic labels for each data point (Cordón et al., 2001). More specifically, for each variable, a linguistic label is selected that best covers the real input value of the data point (Cordón et al., 2001). For instance, the data point that is labelled "100 (High)" in Figure 6 would translate to the rule:

IF x_1 IS low AND x_2 IS high THEN y is high

Step 3: Selection of unique rules

The RB generated in the previous step is likely to run into an essential problem that can arise from the fact that each data point gets a separate rule assigned. Thus, if multiple data points lie in the same partition, the method creates multiple rules with the same antecedents (IF-part) but possibly a different consequence (THEN-part) (L.-X. Wang & Mendel, 1992). To prevent such conflicting rules, it must be ensured that each fuzzy partition has exactly one rule assigned. For this purpose, L.-X. Wang & Mendel (1992) suggest assigning a degree of importance to each rule of the preliminary RB, which is computed as

$$G(R_l) = \mu_{A_1}(x_1^l)...\mu_{A_n}(x_n^l)\mu_B(y^l), \tag{10}$$

where R_l is the l-th rule in the RB, μ_{A_i} (μ_B) is the membership function of the i-th input respectively output variable and $(x_1^l, ..., x_n^l, y^l)$ is the sample related to rule l. The final RB is obtained by taking the rule with the highest importance in each partition.

5.3 Rule base optimisation: Simulated annealing (SA)

In the previous step, rules were extracted that are seen to describe the ordering behaviour of the past in the most relevant way. In other words, the current system tries to mimic the decision patterns from the past and similarly apply them in the future. This would require that the current order policies are already optimal, but in practice, that is usually not the case for multi-echelon inventory systems. Therefore, we propose an additional step namely the determination of rule outputs (consequences) that minimise the total SC costs while still providing a reasonable fill rate. Hence, for each rule in the given RB, the model must select exactly one linguistic label for the output such that, in the end, the resulting RB minimises the objective value.

Let L be the number of rules in the given RB and m the number of fuzzy values representing a fuzzy variable. Then, there exist m^L different combinations of rule outputs in the RB. For example, if a RB contains 20 rules and a variable is represented by three fuzzy sets (MFs), there are $3^{20} = 3,486,784,401$ different rule output configurations. Hence, finding optimal rule outputs in a fuzzy KB is of exponential complexity, namely exponential in the number of fuzzy rules.

A technique that is suitable for finding the best solution among a finite but immensely high number of possible solutions is the simulated annealing (SA) algorithm. The main idea is to start with an initial solution and to iteratively replace this solution with a new "better" solution from a predefined neighbourhood of the old solution until a stopping criterion is met. The crucial point is that not only solutions with strictly better solution values are accepted, but likewise worse solutions with a certain probability that is negatively proportional to the difference between the old and new solution. In other words, solutions that are close to being better are more likely to be accepted than ones that are clearly worse. This way, the algorithm should be prevented from getting stuck in local optima. The individual steps of the SA algorithm, including definitions of the used parameters, are explained in the following. Figure 7 offers a summary of all the steps.

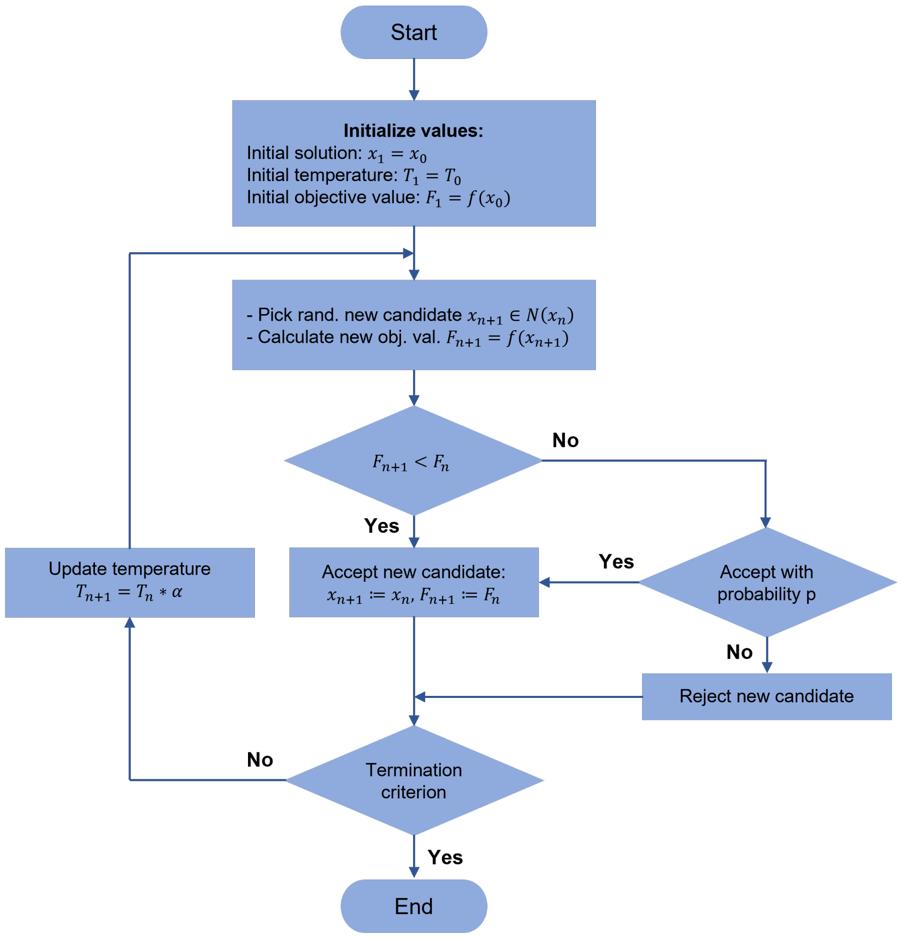

Start
Initialize values:
Initial solution: $x_1 = x_0$
Initial temperature: $T_1 = T_0$
Initial objective value: $F_1 = f(x_0)$
- Pick rand. new candidate $x_{n+1} \in N(x_n)$
- Calculate new obj. val. $F_{n+1} = f(x_{n+1})$
$F_{n+1} < F_n$
No
Yes
Update temperature $T_{n+1} = T_n * \alpha$
Accept new candidate: $x_{n+1} := x_n, F_{n+1} := F_n$
Yes
Accept with probability p
No
Reject new candidate
No
Termination criterion
Yes
End

Figure 7: Flow chart of SA algorithm

5.3.1 Required parameters and initialisation

As shown in Figure 8, a solution x_n of the SA algorithm is represented by an array $x \in Q^L$ (set of linguistic labels), where $x(l) \in Q$ is the linguistic label describing the output of rule l. The algorithm is initialised with x_0, the solution array related to the RB obtained from the WM method.

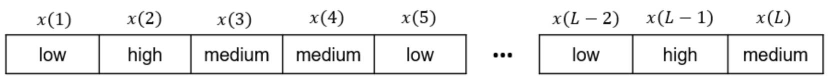


Figure 8: Representation of solution array x in SA algorithm

In addition to the initial RB, the algorithm requires the specification of a fitness function $f(x)$, an initial temperature T_0, a cool-down factor α, and a neighborhood $N(x)$.

Fitness function

In this application, the fitness function of a solution x is defined as in Ignaciuk & Wieczorek (2020):

$$f(x) = (1 - \frac{C}{C_{max}})^{\gamma} FR^{\phi}. \tag{11}$$

As in this section, the method is presented for the case of a single inventory (an application to a global system follows in 5.5), C represents the total inventory costs computed in (3), but only taking into account the portion related to the particular inventory of interest. The total costs C are retrieved by means of a simulation. In fact, they are the costs that result from applying the current fuzzy KB to the inventory over the whole training time horizon. Furthermore, FR is the fill rate as shown in (9) and C_{max} is a large constant that must be defined in advance such that it is larger than any total inventory costs that might be faced during the simulation. This constant must be selected by trial and error, as due to the unlimited storage capacities of the inventories, there is no theoretical boundary for the SC costs. In the case study of this thesis, for example, C_{max} is set to be 5 times the total SC costs obtained by the model of comparison (EOQ). Moreover, γ and ϕ are tuning coefficients that are used for prioritising SC costs vs. customer satisfaction. In particular, setting a higher γ puts more weight on reducing the total costs, whereas increasing ϕ favours higher customer satisfaction. Note that the costs are normalised by introducing the term C/C_{max} in order to prevent scaling issues. As a result, it holds that $C/C_{max} \in [0, 1]$ and $FR \in [0, 1]$, implying that $f(x) \in [0, 1]$. In particular, the higher the fitness value (closer to 1), the better the SC performs. A fitness value $f(x) = 1$ would imply zero SC costs and a perfect fill rate.

Initial temperature and cool-down factor

The initial temperature T_0 is of great importance for a successful SA algorithm. A too low temperature may lead to insufficient exploration of the search space and thus promote getting stuck in local optima, whereas an exceedingly high initial temperature may result in a random search, i.e., the algorithm wildly jumps around by accepting bad solutions (Ledesma et al., 2008). The level of exploration is additionally controlled by the cool-down factor α. Throughout the algorithm, the temperature is linearly reduced to ensure the refinement of qualitative solutions (local search) found in the beginning. In practice, α is typically chosen to be between 0.8 and 0.99 (Weicker, 2015). More practical considerations on the initial temperature and the factor α can be found in Ledesma et al. (2008) and Ben-Ameur (2004).

Neighbourhood of solution

Let x_n be the current solution, and further, let $Q = \{\text{"low"}, \text{"medium"}, \text{"high"}\}$ be the set of linguistic labels describing the variable "order quantity". Then, a solution x_{n+1} (in the neighbourhood of x_n) can be obtained by picking K randomly selected indices between 1 and R (number of rules) and applying the following three rules for each of the indices $k = 1, ..., K$:

1. If $x_n(k)$ = "low", then $x_{n+1}(k)$ = "medium".
2. If $x_n(k)$ = "high", then $x_{n+1}(k)$ ="medium".
3. If $x_n(k)$ = "medium", then $x_{n+1}(k)$ = "low".

The neighbourhood $N(x)$ is the entirety of all the arrays that can be created in this fashion.

5.3.2 The algorithm

In each iteration, a candidate solution $x_{n+1} \in N(x_n)$ is picked at random. If the fitness value of the candidate solution is better than or equal to the fitness value of the current solution, it is immediately accepted as the new current solution for the next iteration. Otherwise, the candidate solution is only accepted with a probability of

$$\exp(-\frac{\Delta f}{T}), \tag{12}$$

where $\Delta f = f(x_n) - f(x_{n+1})$ and T is the current temperature. If the termination criterion is met, the algorithm stops. Otherwise, the current temperature is updated and the algorithm continues with the next iteration (Weyland, 2008). In this application, the termination criterion is reached if no improvement in the solution value can be observed for a certain number of iterations.

5.4 MF parameter tuning: LA-tuning

In this step, the KB that is obtained from the previous two steps is tuned with the aim of further increasing the fitness value (11) by means of a genetic algorithm called LA-tuning. This method, which was invented by Alcalá et al. (2007), originally consists of two main components that can be performed simultaneously: a reduction of the RB and a tuning of the definition points of triangular

MFs. However, for the application of a multi-echelon FRBS, only a tuning of the MF parameters is considered, as a RB reduction would strongly increase the search space of the genetic algorithm and, therefore, may be counterproductive in finding an optimal solution in a reasonable time.

As suggested by Alcalá et al. (2007), the genetic algorithm is based on the evolutionary model CHC (Eshelman, 1991). This model is selected because of its good balance between exploration and exploitation and thus its applicability to problems with highly complex search spaces (Alcalá et al., 2007). In Figure 9, the main steps of CHC are displayed. The scheme is applied until a stopping criterion is met. For instance, in this thesis, the algorithm is terminated after a certain number of crossover evaluations. Alternatively, the algorithm may be stopped after a certain number of restart cycles. In the following, the components of CHC are explained in detail.

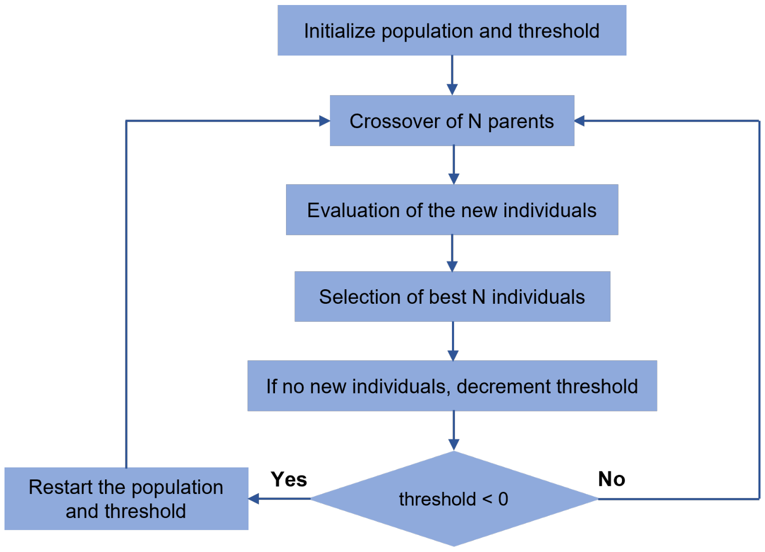

Initialize population and threshold
Crossover of N parents
Evaluation of the new individuals
Selection of best N individuals
If no new individuals, decrement threshold
threshold < 0
Yes
No
Restart the population and threshold

Figure 9: Evolutionary model of CHC. Adapted from Alcalá et al. (2007).

5.4.1 Chromosome representation

The method of Alcalá et al. (2007) makes use of a linguistic 3-tuples representation (s, α, β) of triangular MFs. This means that by starting with a fuzzy set s (represented by a triangular MF with three definition points), a new fuzzy set can solely be represented by two points, namely its lateral displacement α and amplitude variation β towards the initial fuzzy set.

As shown in Figure 10, the real value α expresses to what extent the whole domain of a label is shifted to its preceding or succeeding label (depending on the sign). Concretely, it is given as a fraction of the distance between the peaks of both labels and is limited to the range $[-0.5, 0.5)$, as defined by Alcalá et al. (2007). Hence, when moving the peak of a label, it cannot exceed half the distance to its preceding or succeeding label. The value β represents a change in the width (amplitude) of a MF and is given as a fraction of its original size. Similarly, $\beta \in [-0.5, 0.5)$, which means that the width of a label can be increased or reduced by at most 50% of its original width.

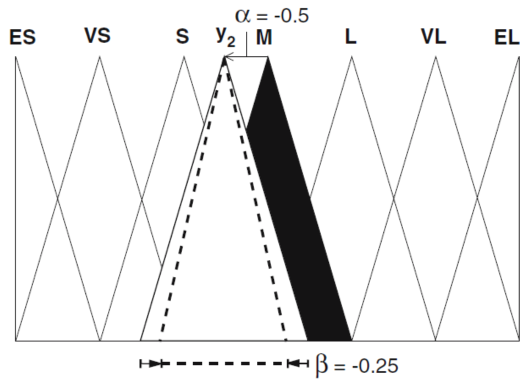


Figure 10: Lateral displacement and amplitude variation of a linguistic variable (Alcalá et al., 2007)

A chromosome encodes the lateral displacement and amplitude variation of all the MFs describing the system. Figure 11 (bottom) visualises the configuration of such a chromosome. As illustrated, a chromosome is divided into the two parts C^L (lateral displacement) and C^A (amplitude variation). Let n represent the number of fuzzy variables in the system ($n-1$ input variables and 1 output variable) and m represent the number of labels per fuzzy variable. Then, considering a global tuning of the semantics, both components C^L and C^A are formally described as:

$$C^L = (c^L_{11}, ..., c^L_{1m}, ..., c^L_{n1}, ..., c^L_{nm})$$
$$C^A = (c^A_{11}, ..., c^A_{1m}, ..., c^A_{n1}, ..., c^A_{(nm)})$$

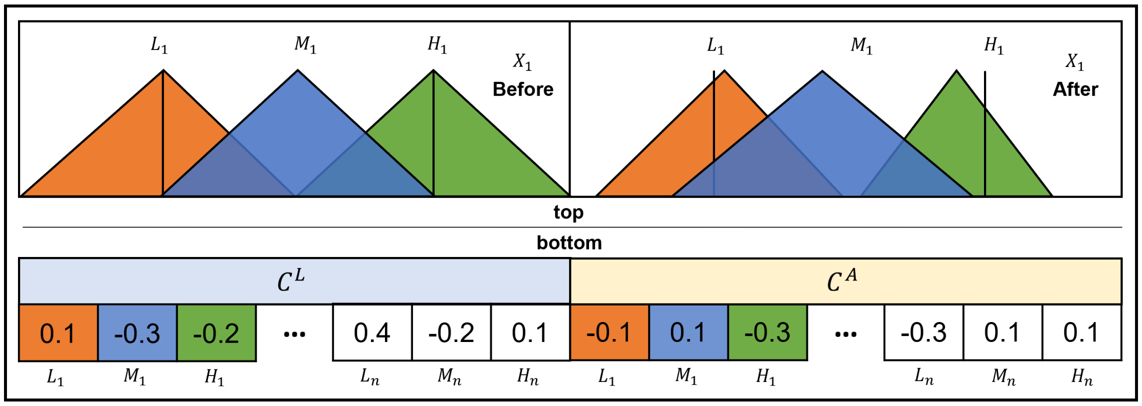


Figure 11: Example of encoded chromosome (bottom) and corresponding MF changes of variable X_1 (top)

Figure 11 (top) illustrates how the shape of a variable (e.g., X_1) may alter after an iteration has been performed. A global tuning of the semantics means that the same tuned label of a fuzzy variable is applied to all rules where it occurs, i.e., the tuning is applied to the level of linguistic partition (Alcalá et al., 2007). For instance, a newly defined fuzzy partition "Demand is (High, 0.1, -0.3)" would similarly be applied to all rules in which "Demand is high" is one of the antecedents. This type of tuning is preferred since it leads to models with higher interpretability.

5.4.2 Initialisation

For initialising the LA-tuning algorithm, an initial population of N individuals is created. This population is based on the individual that corresponds to the currently optimal KB, that is, the KB obtained from step 2 (SA). Following the procedure outlined in Alcalá et al. (2007), first, an individual having all genes equal to 0 (no lateral displacement, no amplitude variation) is added to the population. The remaining $(N - 1)$ individuals are created by randomly assigning real values in the range $[-0.5, 0.5)$ to all genes.

5.4.3 Chromosome evaluation

As opposed to the MSE that is used by Alcalá et al. (2007) for evaluating the quality of an individual, in the proposed FRBS, the fitness function as presented in (11) is used. Since, in general, the observed order quantities of a multi-echelon inventory system are not the "right" (i.e., optimal) quantities, an accuracy measure is groundless. Therefore, the quality of a solution is measured by the fitness value that results from applying the KB to the training data set.

5.4.4 Selection procedure

As used by Alcalá et al. (2007), a "population-based selection" is applied for updating the population after crossover has been applied. In particular, the N individuals of the current population are combined with the new offspring to finally choose the N fittest individuals for the new population. Following the model of CHC (Eshelman, 1991), in order to diversify the population throughout the algorithm, an incest prevention algorithm combined with a restarting approach is applied instead of mutation. This means that only sufficiently different parents are crossed with each other. The difference between two parents is thereby measured by their Hamming distance. The Hamming distance between two equal-length strings of symbols is the number of bit positions in which they differ (Macleod, 1993). Since a chromosome is represented by real values, each of its genes must be transformed into a gray code with a fixed number of genes (BITSGENE) first (Alcalá et al., 2007). This way, a real value is represented by a fixed number of 0's and 1's. Finally, two parents can only be crossed if their Hamming distance divided by 2 (the Hamming distance of the expected children from their parents) is over a difference threshold (Eshelman, 1991). This threshold is initially set to half the expected Hamming distance between two randomly generated strings (Eshelman, 1991), that is,

$$L_T = (\#Genes * BITSGENE)/4, \tag{13}$$

where $\#Genes$ is the number of genes in a chromosome. The threshold is decremented by 1 every time no offspring entered a new population. Hence, less diversity is required as the population naturally converges (Eshelman, 1991). In addition, L_T is reduced by $\theta\%$ of its current value in each iteration, with θ typically set to 10%. This way, the algorithm becomes independent from #Genes and BITSGENE (Alcalá et al., 2007).

Following the original CHC scheme, in order to avoid a premature convergence, the algorithm uses a restart approach every time the threshold L_T falls below 0. As suggested by Alcalá et al. (2007), a restart is accomplished by retaining the best individual and generating $N - 1$ new individuals by adding a random number in the range $[-0.125, 0.125)$ to each gene of the retained individual. Note that a newly generated value cannot exceed the predefined range of $[-0.5, 0.5)$. In such a case, it is substituted by the closest point within the allowed range, namely -0.5 or 0.5.

5.4.5 Crossover operator

As suggested by Alcalá et al. (2007), the PCBLX (parent centric BLX) operator is used for all crossover operations in this application (Herrera et al., 2003). The main essence of this method is to create offspring that lies within a close distance to both parents. In this way, the convergence of the algorithm is forced via the offspring, which proves to be advantageous in evolutionary models such as CHC (Alcalá et al., 2007). Concretely, each offspring lies within the radius of a particular parent, i.e., one child is like the male parent, whereas the other is similar to the female parent. As explained by Alcalá et al. (2007), let $X = (x_1, ..., x_g)$ and $Y = (y_1, ..., y_g)$, where $x_i, y_i \in [a_i, b_i], i = 1, ..., g$, are both parents to be crossed. Then, an offspring $Z = (z_1, ..., z_g)$ is created by randomly choosing each z_i from the range $[l_i, u_i]$, where $l_i = max\{a_i, x_i - I\}$, $u_i = min\{b_i, x_i + I\}$, and $I = |x_i - y_i|$. Finally, by taking one parent as a male parent and another as a female parent, and vice versa, two offspring are created.

5.5 Global multi-echelon FRBS

In this section, we propose a fuzzy rule-based method for finding a global order policy for multi-echelon inventory systems. The method is specifically designed for SCs in which inventory information is shared among all echelons throughout the entire SC. The main goal is to derive fuzzy KBs for each inventory that ultimately lead to the lowest total SC costs and a reasonable fill rate when simultaneously applied in future decisions.

5.5.1 Fuzzy assumptions

In the proposed model, it is assumed that each echelon can have different fuzzy variables. For instance, a fuzzy variable "market price" of raw materials would only make sense at the production level, since this echelon is responsible for the replenishment of raw materials. Furthermore, a lead time may be constant or not given at all in some cases; for example, in order to respond quickly to unexpected peaks in customer demand, distribution centers may be adjusted to replenish stock from warehouses on the same day. Another assumption is that the number of linguistic labels describing a fuzzy variable is constant throughout the whole model. A variable number of labels would significantly increase the difficulty of the model's implementation. Moreover, due to the natural complexity that is given in multi-echelon inventory systems, only a minimal number of three labels per variable would lead to reasonable computation times for the SA algorithm and the LA-tuning.

5.5.2 Formulation of optimisation problem

Based on the work of Xiong & Helo (2006), let us first formulate a fuzzy rule-based multi-echelon model (MEF) that takes a predefined set of KBs as an input. Since a global inventory control policy must consider the inventory information of each installation in the SC, a MEF is formulated as

$$\begin{aligned} \text{MEF} &= \text{Multi-echelon fuzzy model}(F_1, F_2, ..., F_n) \\ &= f\Big\{[F(RB_{-1}, MF_{-1})], [F(RB_{-2}, MF_{-2})], ..., [F(RB_{-n}, MF_{-n})]\Big\}. \end{aligned} \tag{14}$$

F_i thereby represents the fuzzy model, that is, the KB that describes all control actions at inventory i. A fuzzy model F_i consists of a rule base RB_{-i} and an array MF_{-i} that contains the three definition points of all fuzzy MFs in the DB of inventory i. Hence, the parameters of the fuzzy variables can differ across inventories. Furthermore, let $C(T, I_T, MEF)$ be the total SC costs resulting from applying the KBs of the MEF to a SC for a set of time periods T, taking into account the input data I_T for these time periods, and let $FR(T, I_T, MEF)$ be the corresponding fill rate. Then, the objective is to find a MEF that maximises

$$\max_{MEF} \left(1 - \frac{C(T, I_T, MEF)}{C_{max}}\right)^{\gamma} FR(T, I_T, MEF)^{\phi}. \tag{15}$$

As discussed in Section 5.3.1, C_{max} is a large constant that must be determined in advance so that it is greater than any total SC costs that may occur during the simulation process. The tuning parameters γ and ϕ are defined as in (11).

In order to solve the optimisation problem (15), the 3-step procedure as presented in Figure 5 and introduced in sections 5.2-5.4 is applied. In the next section, it is shown how the individual components are integrated in the context of a global optimisation policy.

5.5.3 Application

The extension to a global multi-echelon FRBS requires a few adjustments in the application of the SA algorithm and the LA-tuning algorithm. But first, the procedure starts with the generation of the initial KBs that serve as inputs for the MEF in (14). This is accomplished by applying the WM method introduced in 5.2 to each inventory separately.

Next, a rule base optimisation using the SA algorithm presented in 5.3 is performed. Principally, the algorithm follows the same steps as shown. However, it must be noted that in the case of a global MEF, the input RB that is optimised comprises all RBs, that is, a RB for each inventory, combined together. More specifically, by defining the number of rules in the RB related to inventory i as L_i, the total number of rules in the combined RB becomes $\sum_{i=1}^{n} L_i$. Furthermore, let x_i be the solution array representing the RB of inventory i. Then, the expanded solution array $\boldsymbol{x}$ is expressed as

$$\boldsymbol{x} = \Big[\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_n}\Big].$$

Finally, the LA-tuning as introduced in 5.4 is applied. However, since for a centralised global inventory control model, information along the whole SC must be considered, the representation of a chromosome must be expanded. Consider the representation of a chromosome as shown in Figure 11. Moreover, let the initially presented array $C^L + C^A$ describe the definition points of triangular MFs for a particular inventory. Then, the new combined encoding of a chromosome is structured as displayed in Figure 12.

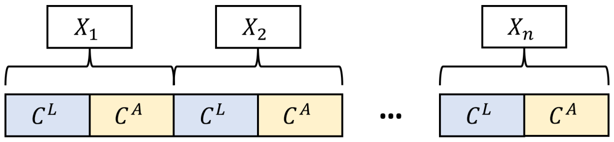


Figure 12: Expanded encoded chromosome for global multi-echelon FRBS

5.5.4 Applicability in real-world problems

In order to evaluate the complexity and thus the applicability of the proposed global MEF to real-world problems, it is essential to take a closer look at the number of fuzzy variables in the system first. There are n inventories in the system. Furthermore, assuming that each inventory i is represented by b_i fuzzy variables, this leads to a total number of - $\sum_{i=1}^{n} b_i$ - fuzzy variables in the system. This number is important since the number of rules in a fuzzy system increases exponentially with the number of fuzzy input variables, and further, a higher number of variables directly enlarges the search space of the SA and LA-tuning algorithms. Since the WM algorithm is proposed to be applied separately to each inventory, the increase in rules can be brought from exponential to linear. However, the larger search space of the evolutionary algorithms poses a problem. In general, the number of iterations that are necessary to approach a global optimum considerably increases with the size of the solution array $\boldsymbol{x}$ in the SA algorithm and similarly with the size of a chromosome in the genetic algorithm. In contrast, recall that the objective value (fitness function) of a solution requires a complete simulation of the SC over a predefined (training) time horizon. Hence, the computation time per simulation run increases linearly with the number of KBs in the MEF. In other words, the number of iterations as well as the time spent per iteration increases when the number of fuzzy variables is increased.

Taking all that into consideration, it is concluded that a pure global MEF that considers all available inventory information is (mostly) not applicable to real-world problems as they usually deal with SCs that consist of a high number of inventories. However, as will be shown in Section 6, for small applications, it may still be a useful tool for a global inventory control.

5.6 Global multi-echelon FRBS - Heuristic

As concluded in the previous section, a global multi-echelon FRBS leads to exceedingly high computation times in SCs with many inventories. To tackle the issue of time complexity, we propose a heuristic that is significantly faster and yet considers information along the SC to some extent.

The main idea is to apply the introduced 3-step procedure (Figure 5) to each inventory separately but still in a particular order such that the derived order policies of succeeding inventories are considered in the fuzzy input of lower echelon inventories. Algorithm 1 clarifies the individual steps of the heuristic.

Algorithm 1 Multi-echelon FRBS heuristic

```
1:  Input: dates = $\{1, 2, ..., T\}$, E = {DC, WH, P} (ordered set), I = $\{i_1, i_2, ..., i_n\}$,
2:         dataSets = $\{\boldsymbol{A}^{(i_1)}, \boldsymbol{A}^{(i_2)}, ..., \boldsymbol{A}^{(i_n)}\}$
3:  knowledgeBases = $\emptyset$
4:  for each echelon e in E: do
5:      p ⟵ precedingEchelon(e)
6:      for each inventory k in echelon p: do
7:          Initialise a vector $\boldsymbol{q}^{(k)}$ of |dates| zeros
8:      for each inventory i in echelon e: do
9:          initKB ⟵ applyWM($\boldsymbol{A}^{(i)}$)
10:         initKB ⟵ applySA(initKB)
11:         bestKB ⟵ applyLA(initKB)
12:         knowledgeBases := knowledgeBases ∪ {bestKB}
13:         newQntVector ⟵ simulateInventory(dates, bestKB, $\boldsymbol{A}^{(i)}$)
14:         b ⟵ precedingInventory(i)
15:         $\boldsymbol{q}^{(b)} = \boldsymbol{q}^{(b)}$ + newQntVector
16:     for each inventory k in echelon p: do
17:         Replace demand column of $\boldsymbol{A}^{(k)}$ by $\boldsymbol{q}^{(k)}$
18: Output: knowledgeBases
```

The heuristic requires a set of training dates (*dates*), a definition of the echelons (E) and inventories (I), and a set of matrices $\boldsymbol{A}^{(i)}$ containing the fuzzy input data of each inventory for each date in *dates*. In particular, the set E includes the echelon types "DC" (distribution center), "WH" (warehouse), and "P" (plant) in this particular order. This order is essential, since the heuristic is required to process the echelons from low to high in a sequential order. The data matrix $\boldsymbol{A}^{(i)}$ corresponds to a $T \times m^i$ matrix, where T is the number of training dates and m^i is the number of fuzzy variables describing inventory i. Hence, an element $\boldsymbol{A}^{(k)}(i, j)$ of a data matrix $\boldsymbol{A^{(k)}}$ is defined as the crisp input value of a fuzzy variable j on date i that is related to inventory k. For each echelon (from low to high), the algorithm applies the 3-step procedure for determining an optimal KB to each inventory (within this echelon) separately. In particular, the method *applyWM()* applies the WM method (Section 5.2) to a given input data set, whereas *applySA()* and *applyLA()* perform the SA algorithm (Section 5.3) and the LA-tuning algorithm (Section 5.4).

Once an optimal order policy has been determined for an inventory, it is directly applied to the given training data (*simulateInventory()*) to produce a vector of T order quantities (*newQntVector*). Hence, *newQntVector* contains an order quantity for each day of the training horizon. This vector is essential since it serves as new demand data for the predecessor of an inventory. In particular, for each inventory i in the preceding echelon, the column of $\boldsymbol{A^{(i)}}$ that corresponds to the fuzzy variable "demand" is replaced by a column of order quantities that result from applying the optimal KBs to the successors of inventory i. However, since an inventory can have multiple successors in an arborescent system, the order quantities of all successors must be summed up for each given day.

This is accomplished for a preceding inventory k by using an auxiliary vector $\boldsymbol{q^{(k)}}$ $(T \times 1)$ that keeps track of the total demand per day that results from the fuzzy order policies of all its succeeding inventories. Finally, the output of Algorithm 1 is a set of optimal KBs — one KB per inventory.

6 Results

In this section, the validity of the two proposed models (exact FRBS and heuristic FRBS) is tested in an extensive case study. Both models are compared to a simple local policy (Section 6.1.2), that is, a policy that takes inventory decisions for each echelon and inventory independently. The case study consists of two main parts: 1) a simulation/sensitivity analysis to evaluate the robustness of the models at different levels of uncertainty; and 2) an application of the heuristic to a large instance using real-world data.

6.1 Case study - Description

This thesis's case study is based on an artificial data set that describes the inventory of a German muesli producer. The data set, which is freely accessible on the platform www.data.world, originates from an ERP simulation game called "Manufacturing game", developed by the ERPsim Lab, HEC Montreal. This game was designed with the purpose of teaching students the underlying concepts of integrated ERP systems by using an innovative "learning-by-doing" approach (Léger, 2006). As this simulation mimics the complexity of a real-life environment (Léger, 2006), the data set is seen to be legitimate for applying the concepts suggested in this thesis. Further, the use of simulation games for evaluating a method's performance is not uncommon in research. For instance, Geevers (2020) used the Beer game, a simulation game teaching the consequences of the bullwhip effect in serial multi-echelon systems, in order to analyse a deep reinforcement learning approach for inventory problems.

For this case study, the following artificial scenario is assumed: Over the past year, the producer of cereals has applied a simple local inventory policy (Section 6.1.2) to each of its inventories, which as a result has not led to the expected outcomes. Even though the company recorded reasonably low total SC costs, the anticipated fill rate could not always be satisfied. After a thorough analysis, the company came to the conclusion that their current model is not well designed to handle any level of uncertainty in the demand, lead time, and market prices of raw materials. Given this information, the company wants to establish a new inventory control model that is more robust and equally prioritises customer satisfaction and total SC costs. In particular, given the current local order policy, the main goal is to derive an improved order policy that minimises the total SC costs along the whole chain while still maintaining a fill rate of at least 95 %. Details related to the company's SC and their currently applied local order policy are provided in the following sections.

6.1.1 The supply chain

The suggested data set describes the make-to-stock manufacturing supply chain of a German cereal producer. As shown in Figure 13, the SC is organised into five different stages: procurement, production/assembly, warehousing, distribution, and retail. The company fabricates twelve different end-products (cereal boxes) which comprise six different flavours (Original, Nut, Blueberry, Strawberry, Raisin, and Mixed-Fruit) in two different sizes (0.5 kg and 1 kg). The different sorts are composed of at most six different basic ingredients (nuts, blueberries, strawberries, raisins, wheat, oats) and are packed in both plastic bags and cardboard boxes (packaging material) (Léger, 2006). All in all, the company's inventory system consists of 58 separate inventories — 6 raw material stocks (P), 4 packaging material stocks (P), 12 end-product stocks (WH), and finally 12 end-product stocks (DC) at each of the three distribution centers.

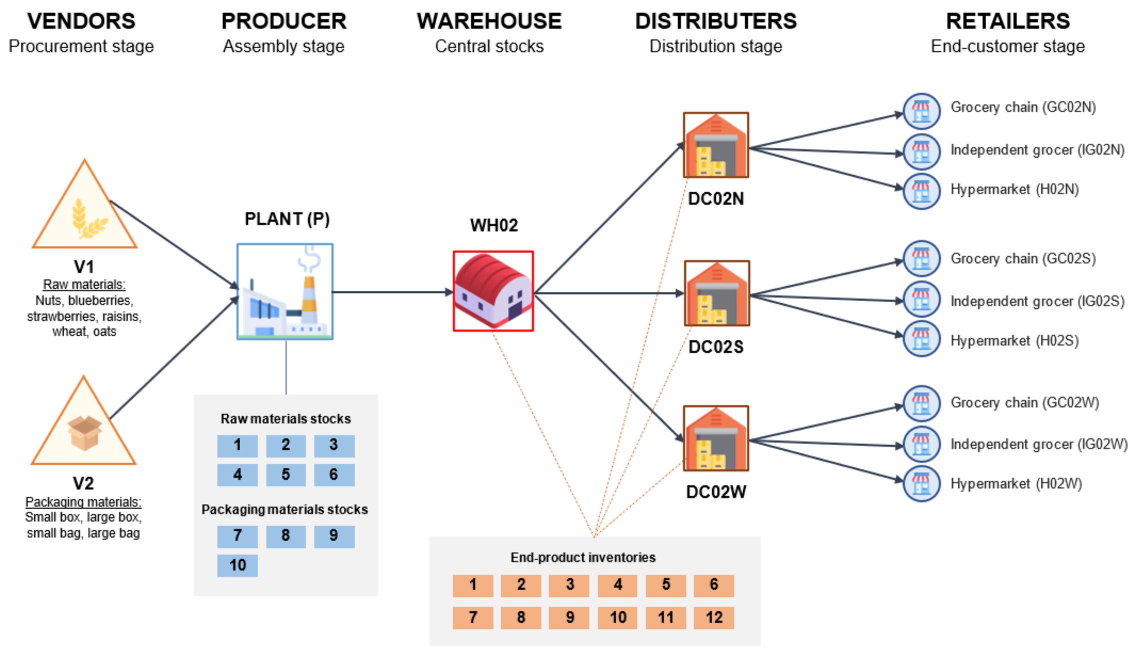


Figure 13: Supply chain diagram of a German muesli producer

The specifics of the company's SC fully align with the description of Section 4. Hence, the model assumptions, the underlying cost structure, and the objectives entirely apply to the SC. Note that all the data describing the SC is provided in a separate GitHub repository accessible via:

`www.github.com/piperovilia/Fuzzy-ME-inventory-model-with-automatic-KB-generation`

A detailed description of the available data sets can be found in Appendix B. The complete SC, as presented in this section, is used in the real-world case study in Section 6.3.

6.1.2 Current local order policy

Thus far, the company controls its inventories by applying a periodic review (s,S)-policy that is based on the EOQ (economic order quantity) respectively the EPQ (economic production quantity) at production level to each inventory separately. To be more specific, assuming normal demand and a constant lead time, the optimal batch quantity (Q) is determined by using the EOQ (EPQ) formula, which minimises the sum of total holding and order (setup) costs. The applied EOQ formula is given by

$$Q = \sqrt{\frac{2Ad}{h}}, \tag{16}$$

where A are the fixed costs per order, d is the sample average of the daily demand, and h are the holding costs per day and unit. The decision on whether an order is placed is determined by the reorder level

$$s = SS + \mu^{'}, \tag{17}$$

which aims to cover the demand $\mu^{'}$ that occurs within the lead time of the new order (lead time demand). Note that even though the company assumes a constant lead time, in reality, there is a certain degree of uncertainty linked to it. In order to prevent a potential stock-out due to such uncertainty, the company additionally reserves a safety stock (SS), which is determined by

$$SS = k\sigma^{'}, \tag{18}$$

where $\sigma^{'}$ is the standard deviation of the lead time demand and k is a safety factor that aims to control the required level of safety stock according to the stock-out probability that is required by the company. Hence, when an inventory's inventory position falls to or below the reorder level s during a given review period, the inventory orders up to the level $S = s + Q$.

6.2 Model comparison (simulation experiment)

In this section, the two proposed models (exact FRBS and heuristic FRBS) are compared to the EOQ-based local policy in a controlled simulation setting. In particular, the models' performances are evaluated and compared for different scenarios of uncertainty. Subsequently, the results are used to assess the robustness of the proposed models by conducting a sensitivity analysis.

6.2.1 Simulation setting

For a fair comparison between all three models (EOQ, exact FRBS, and heuristic FRBS), the SC as presented in 6.1.1 was modified. First, the network was reduced to one single product (0.5kg Original Mix) that solely consists of two ingredients (60% wheat and 40% oat). This resulted in a system of 6 inventories (instead of 58) which guaranteed a reasonable runtime of the exact FRBS. Moreover, instead of using real SC data, the variables demand, lead time, and market price were simulated. This ensures that the EOQ-based local order policy being used as a comparison is applied in perfect conditions. In this way, it should be guaranteed that differences in the results between all three models can be solely attributed to the working principles of the methods rather than to wrong model decisions or uncleanness in the data. Furthermore, a controlled simulation of the fuzzy variables enables the generation of appropriate uncertainty scenarios, which are used to assess the robustness of the proposed methods. In particular, the reduced problem instance was exposed to eight different scenarios of uncertainty, which result from applying two levels of uncertainty (low and high) to each of the three fuzzy variables. Table 2 displays the data generation processes of the variables.

Variable	Low uncertainty	High uncertainty
D	MP$\sim$N(400, 36)	MP$\sim$N(370, 142)
LT	LT$\sim$Gamma*(1.5, 1)	LT$\sim$Gamma*(3, 1)
MP	Wheat: MP$\sim$N(0.99, 0.099)	Wheat: MP$\sim$N(0.99, 0.297)
	Oat: MP$\sim$N(0.86, 0.086)	Oat: MP$\sim$N(0.86, 0.258)
*Gamma(k,θ) parametrisation with shape parameter k and scale parameter θ (in days) used		

Table 2: Data generation processes of (fuzzy) uncertain variables

For simplicity, the provided demand distribution parameters relate to one particular customer (retailer), but are similarly applied to all nine customers. Moreover, lead time uncertainty occurs between the warehouse and the plant (= production time) and likewise between the plant and the vendors of raw material. For simplicity, the lead time between the DCs and the warehouse is fixed at one day. Furthermore, since it is assumed that only plants can place orders for raw materials, market prices are only considered at the production level. Finally, to avoid negative demand and market prices, low coefficients of variation of 0.38 (0.3) for the demand respectively market price were chosen. In this way, the probabilities of negative outcomes are kept at 0.00459 and 0.00043.

The following eight scenarios were considered in the simulation: (1) L/L/L, (2) L/L/H, (3) L/H/L,

(4) L/H/H, (5) H/L/L, (6) H/L/H, (7) H/H/L, and (8) H/H/H. A scenario L/L/H, for instance, corresponds to the scenario of low (L) demand uncertainty, low (L) lead time uncertainty, and high (H) market price uncertainty. The scenarios were ultimately simulated over a period of 360 days — 252 days (70%) for training and 108 days (30%) for testing. All fuzzy variables were represented by 3 labels (”Low”, ”Medium”, and ”High”) for both the exact model and the heuristic. Tables 3 and 4 summarise the parameters being applied in the SA and LA-tuning algorithms.

Parameter	Global (exact)	Local (heuristic)
Cool-down factor (α)	0.95	0.95
Initial temperature (T_0)	0.25	0.4
Number of switched outputs (K)	60% of outputs	60% of outputs
SC cost prioritisation parameter (γ)	1.0	1.0
Fill rate prioritisation parameter (ϕ)	1.0	1.0
Stopping criterion	200 iter. without progress	200 iter. without progress

Table 3: Parameter setting for simulated annealing

Parameter	Global (exact)	Local (heuristic)
Population size	30	20
Bits per gene (BITSGENE)	12	12
SC cost prioritisation parameter (γ)	1.0	1.0
Fill rate prioritisation parameter (ϕ)	1.0	1.0
Stopping criterion	after 100 iterations	after 100 iterations

Table 4: Parameter setting for LA-tuning (genetic algorithm)

First, it is important to note that due to the high computational times of both the SA and the LA-tuning algorithm, a complete grid search-based tuning of all parameters was not achievable. Instead, the selection of the model parameters was based on practical considerations.

For the SA algorithm, a cool-down factor α of 0.95 — a value between 0.8 and 0.99, as recommended by Weicker (2015) — was selected for both methods. Moreover, by following Ledesma et al. (2008), the initial temperatures of 0.25 (exact FRBS) and 0.4 (heuristic FRBS) were derived based on the premise that the starting value for the probability of acceptance (12) should be approximately 0.8. In this way, the algorithm can sufficiently explore diverse neighbourhoods in its beginning. Furthermore, by trial and error, the number of switched outputs (K) per iteration was set to 60% of the total number of rule outputs. In general, a too low number of switches results in exceedingly high computation times as the algorithm requires a higher number of iterations to explore the entire search space. A too high number of switches, on the other hand, may lead to sub-optimal results since the algorithm may not exploit "good" solution paths sufficiently. We recommend a K in the range between 40% and 80% of the total number of rule outputs. Further, by setting the parameters γ and ϕ to 1.0, the required equal prioritisation of SC costs and customer satisfaction is guaranteed. Finally, terminating the algorithm after 200 iterations without improvement ensures that the algorithm is not terminated prematurely.

As for the choice of the relatively small population sizes of 30 (exact FRBS) and 20 (heuristic FRBS) in the LA-tuning algorithm, it is crucial to study the working principles of the CHC scheme. In particular, the algorithm uses a remarkably aggressive search strategy (elitist selection) that is offset by applying a highly disruptive uniform crossover (Whitley & Sutton, 2012; Eshelman, 1991). Due to this combination, the CHC scheme can be applied to rather small population sizes. The algorithm is proven to be very effective for a population size of 50 (Whitley & Sutton, 2012; Eshelman, 1991). It should be noted that a population size of 50 was not possible due to the long computation times of the LA-tuning algorithm in the context of multi-echelon systems. The increase in computation time was deemed more unfavorable than the loss of performance due to the smaller population size. Aside from population size, the number of bits required to convert a gene to gray code is set at 12. Furthermore, similar to the SA algorithm, an equal weighting of costs and customer satisfaction is considered. Finally, the LA-tuning algorithm is terminated after a fixed number of 100 iterations. As will be demonstrated in subsequent sections, even with such a small number of iterations, the algorithm can achieve a significant increase in the fitness value. This is due to the algorithm's ability to rapidly advance through the search space by combining an elitist selection with a highly disruptive crossover.

6.2.2 Results - Training

In this part, the effectivity of the proposed 3-step KB generation procedure is evaluated by analysing the results of training the three methods (EOQ, exact FRBS, and heuristic FRBS).

Table 5 displays the final training results of the full factorial experiment. The column "EOQ" constitutes the status quo, that is, the SC cost (C) and fill rate (FR) resulting from the current local order policy, whereas the columns "Exact" and "Heuristic" contain the results after training the two proposed fuzzy control policies. The fitness value $f(x)$ is chosen to be the main metric of comparison as it contains information about both the fill rate and the total SC costs. This is crucial since the company aims to increase customer satisfaction while keeping low SC costs.

	EOQ			Exact			Heuristic		
Scenario	f(x)	FR	C	f(x)	FR	C	f(x)	FR	C
L/L/L	0.740	0.926	1,912	0.786	1.000	2,042	0.807	0.999	1,832
L/L/H	0.745	0.931	2,025	0.777	1.000	2,254	0.759	0.983	2,308
L/H/L	0.748	0.935	2,228	0.806	1.000	2,161	0.784	0.992	2,333
L/H/H	0.742	0.928	2,105	0.787	1.000	2,245	0.779	0.997	2,301
H/L/L	0.748	0.935	1,868	0.782	0.999	2,022	0.813	0.997	1,722
H/L/H	0.750	0.938	1,754	0.760	1.000	2,108	0.800	0.998	1,738
H/H/L	0.722	0.903	2,271	0.813	1.000	2,119	0.806	0.999	2,200
H/H/H	0.728	0.910	2,141	0.787	0.997	2,253	0.721	0.987	2,887

Table 5: Final results (fitness values, fill rates, SC costs) of training EOQ, exact model and heuristic

Considering the training period, the exact multi-echelon FRBS could significantly increase the fitness value in each given scenario. The heuristic achieved an increase in 7 out of 8 scenarios (except H/H/H). More importantly, applying the proposed global 3-step procedure (WM, SA, and LA-tuning) led to close-to-perfect fill rates while maintaining SC costs that are not significantly worse than for the EOQ model (except for scenario H/H/H for the heuristic).

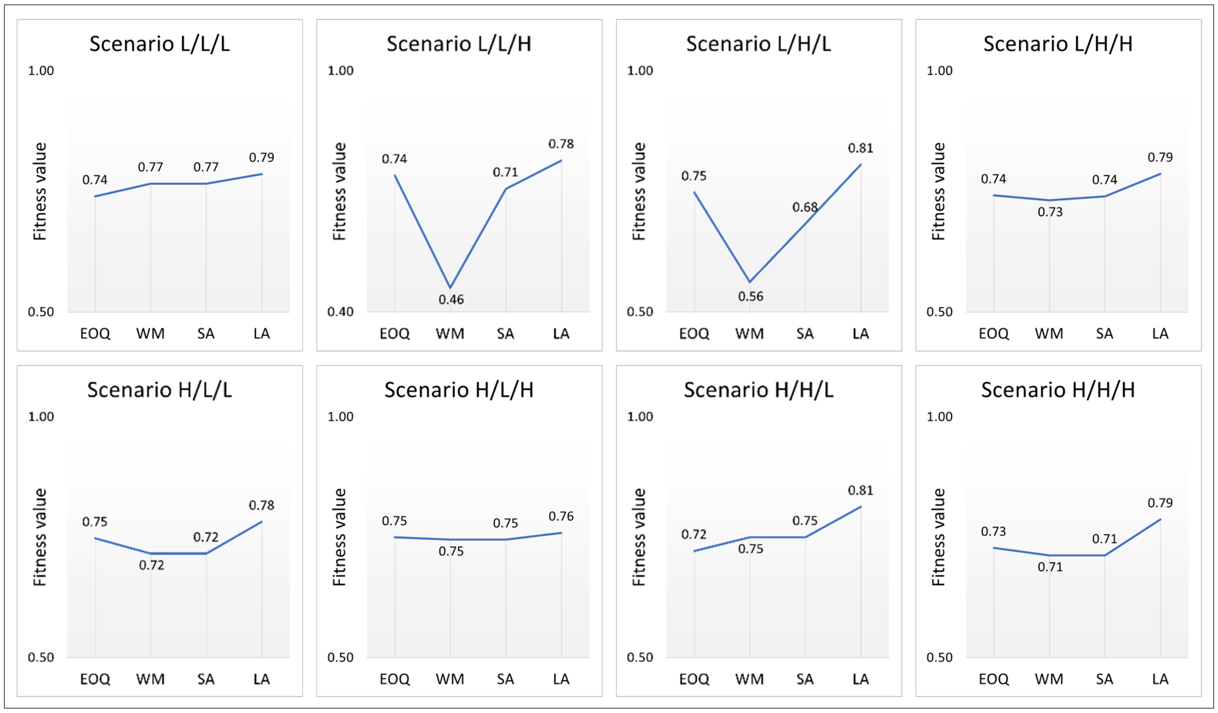


Figure 14: Interim fitness values per scenario after training exact FRBS

Figure 14 provides a more detailed insight into the behaviour of the 3-step procedure. As can be seen, the procedure improves the fitness values in each given scenario. The improvement lies between 1.28% (H/L/H) and 12.67% (H/H/L). While the LA-tuning algorithm (LA) significantly contributes to the improvements, the simulated annealing (SA) is not effective in every case. In the scenarios L/L/L, H/L/L, H/L/H, H/H/L, and H/H/H it has not led to any improvements. This may be because the algorithm cannot effectively search through the immensely high search space before hitting the termination criterion. This data suggests that the SA algorithm may be omitted from the 3-step KB generation procedure. However, as Figure 14 shows, the SA algorithm still proves to be helpful in cases where the Wang-Mendel method (WM) represents the data poorly (L/L/H and L/H/L), as it clearly brings the fitness values (0.71, 0.68) close to the initial levels (0.74, 0.75). Because the SA algorithm produces better initial solutions for the LA-tuning algorithm in such scenarios, and thus may contribute to a better final solution, it is recommended to include the SA algorithm in the 3-step procedure.

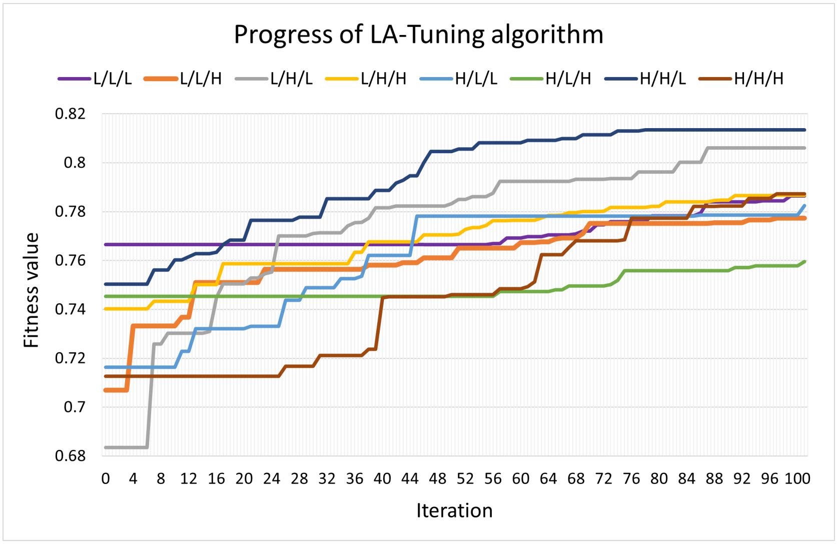


Figure 15: Progress of fitness value when applying LA-tuning in the exact FRBS

Another interesting insight related to the LA-tuning algorithm is provided in Figure 15, which depicts the progress of the fitness value over 100 iterations of the global LA-tuning algorithm for each given scenario. Concretely, the lines of scenarios L/L/L (purple), H/L/L (light blue) and H/L/H (green) indicate that even though the fitness values seem to stagnate for a long period, the algorithm can still lead to an improvement. Scenario H/L/L (light blue) demonstrates this in the most coherent way. The algorithm seems to converge starting from iteration 46. However, interestingly, the algorithm shows improvement again at iteration 98. This implies that the policies derived from the exact FRBS could further be improved by increasing the number of iterations. On the other hand, the number of iterations (for the same computation time) can be increased by reducing the computation time per iteration. This can be achieved by using parallel computing in the evaluation of the new offspring — the computationally most expensive part of the algorithm. Hence, using n splits can make the genetic algorithm up to n times faster.

6.2.3 Results - Test

In the previous section, it was shown that the proposed automatic learning method for fuzzy systems performs as anticipated. It generates fuzzy KBs whose application leads to higher fill rates and lower SC costs depending on the prioritisation that is set. However, the specific results obtained from training do not provide information about the quality of the models in future applications. For evaluating the generalisation ability of the proposed fuzzy rule-based control systems, a test experiment was performed. In particular, for each uncertainty scenario, the models were applied to 100 newly generated test data sets that originated from the same probability distributions as used in training. Tables 6-8 display the average results obtained by this procedure.

As shown in Tables 7 and 8, the application of fuzzy sets and fuzzy logic leads to highly reliable inventory policies. Both the exact model and the heuristic consistently lead to close-to-perfect fill rates regardless of the level of uncertainty faced. The lowest average fill rates resulting from the exact model and the heuristic are 0.985 (H/H/H) respectively 0.988 (L/H/L). The reliability of fuzzy control becomes even clearer when considering the spectrum of fill rates ([MIN,MAX] in Tables 6–8) faced during the simulation. In 100 simulation runs per scenario, the worst fill rates produced by the exact model and the heuristic are 0.924 (L/H/H) respectively 0.828 (H/L/L). The EOQ model, in contrast, can perform poorly in specific cases. For all scenarios, the model happens to create fill rates far below 0.4 (0.221 at its lowest).

Tables 6–8 also show, that in some cases, high reliability comes at the expense of higher total SC costs. Specifically, in the presence of high lead time uncertainty (L/H/L, L/H/H, H/H/L, and H/H/H), fuzzy control leads to higher total SC costs as compared to the EOQ model. In fact, the differences are sufficiently big to result in (slightly) higher fitness values. Note that the data does not provide a clear explanation for this discovery, as for each scenario and model, different cost components (holding costs, penalty costs, or order costs) are found to be responsible for the increased costs. Nevertheless, high lead time uncertainty evidently has an impact on the SC costs when fuzzy control is applied. Another clear result is that the heuristic FRBS performs poorly in the presence of overall high uncertainty (H/H/H). A significant increase of the holding and order costs can be noticed, which indicates that the heuristic counteracts high overall uncertainty by placing orders more frequently and thus maintaining higher stock levels.

Fuzzy control has also an effect on the individual SC cost components. The most significant changes can be observed in the order and setup costs. On average, the exact model and the heuristic led to 3 (2.3) times higher order costs and 2.6 (2.6) higher setup costs as compared to the EOQ model. These increases can be explained by the type of control that is used in the proposed models. As opposed to the EOQ model, which only places an order if the inventory position is below a certain level, the proposed FRBSs decide on each given day whether an order needs to be set. Hence, the systems tend to place smaller (production) orders more frequently.

Scenario	f(x)	FR [MIN,MAX]	C [MIN,MAX]	C_H	C_L	C_O	C_S	C_P	C_U	C_F
L/L/L	0.743	0.928 [0.342,0.989]	847 [704,2191]	319	228	57	10	124	93	54
L/L/H	0.747	0.934 [0.352,0.989]	843 [713,2202]	328	204	57	9	125	94	55
L/H/L	0.771	0.964 [0.295,0.988]	839 [753,2374]	391	104	59	10	134	103	57
L/H/H	0.771	0.964 [0.264,0.988]	829 [761,2438]	379	113	59	11	134	103	57
H/L/L	0.667	0.834 [0.221,0.0.984]	1,018 [664,2483]	288	534	47	8	97	73	43
H/L/H	0.768	0.960 [0.272,0.982]	770 [706,2375]	353	106	55	9	120	90	52
H/H/L	0.762	0.952 [0.332,0.983]	807 [725,2258]	370	133	55	9	119	91	51
H/H/H	0.757	0.947 [0.357,0.982]	819 [737,2205]	367	152	55	9	119	91	51

Table 6: Final results of testing EOQ model (average values after 100 runs) | Costs in KEUR

Scenario	f(x)	FR [MIN,MAX]	C [MIN,MAX]	C_H	C_L	C_O	C_S	C_P	C_U	C_F
L/L/L	0.785	0.999 [0.986,1.000]	908 [874,978]	423	64	140	28	116	87	52
L/L/H	0.762	0.999 [0.992,1.000]	1,001 [957,1042]	526	56	125	27	129	99	57
L/H/L	0.755	0.996 [0.960,1.000]	1,013 [955,1152]	437	106	183	28	120	90	53
L/H/H	0.732	0.990 [0.924,1.000]	1,078 [949,1324]	524	160	136	24	130	102	57
H/L/L	0.815	0.992 [0.955,1.000]	911 [860,1003]	377	164	188	20	117	87	52
H/L/H	0.754	0.997 [0.968,1.000]	939 [879,1019]	453	188	156	24	114	85	51
H/H/L	0.758	0.992 [0.942,1.000]	954 [907,1025]	373	66	240	27	123	94	54
H/H/H	0.740	0.985 [0.942,1.000]	1,019 [925,1123]	422	292	158	20	101	76	45

Table 7: Final results of testing exact FRBS (average values after 100 runs) | Costs in KEUR

Scenario	f(x)	FR [MIN,MAX]	C [MIN,MAX]	C_H	C_L	C_O	C_S	C_P	C_U	C_F
L/L/L	0.805	0.993 [0.966,1.000]	802 [759,881]	321	53	134	23	133	100	59
L/L/H	0.750	0.990 [0.924,1.000]	1,021 [934,1211]	562	80	105	26	134	79	59
L/H/L	0.737	0.988 [0.927,1.000]	1,065 [1001,1189]	544	53	155	25	136	111	59
L/H/H	0.747	0.993 [0.939,1.000]	1,025 [974,1232]	564	131	89	21	143	86	62
H/L/L	0.841	0.989 [0.828,1.000]	763 [713,1167]	328	71	106	21	121	87	53
H/L/H	0.795	0.995 [0.975,1.000]	773 [740,842]	343	45	132	25	123	76	54
H/H/L	0.751	0.993 [0.961,1.000]	985 [942,1055]	515	37	139	25	126	104	54
H/H/H	0.695	0.995 [0.878,1.000]	1,234 [1,191,1466]	715	100	153	25	135	93	59

Table 8: Final results of testing heuristic FRBS (average values after 100 runs) | Costs in KEUR

Another clear difference is that, in general, the fuzzy systems lead to lower penalty costs. This was to be expected as both the exact model and the heuristic lead to considerably higher fill rates, as previously shown. Fuzzy control has an impact on the holding costs as well. On average, the holding costs increased by 27.2% when applying the exact FRBS. The heuristic caused an average surge of 38.1%. Hence, the proposed FRBSs maintain higher inventory levels. This is also understandable since higher fill rates usually go along with higher inventory levels. Finally, no significant changes in production, purchase, and transport costs could be observed. Hence, including the market prices of raw materials as a fuzzy input variable seems to have a minor effect on the overall costs.

At last, the data in Tables 7 and 8 does not suggest that one of the fuzzy models (exact or heuristic) performs better than the other. Considering the fitness value, the heuristic performs better in 4 out of 8 scenarios (L/L/L, L/H/H, H/L/L and H/L/H). However, since 100 iterations most likely do not lead to a convergence of the algorithms, a comparison between both models may be misleading.

6.2.4 Sensitivity analysis

In this section, the sensitivity of the proposed models (exact FRBS and heuristic FRBS) to different levels of uncertainty is evaluated. Figure 16 depicts the change (in percent) of the fill rate and the total SC costs when applying the exact model and the heuristic to the eight scenarios of uncertainty introduced in the previous section. For both models, the scenario of overall low uncertainty (L/L/L) serves as the base case.

Firstly, recall from Tables 7 and 8 that the differences in the fill rate are remarkably small for both models. Hence, a thorough analysis of the sensitivity based on the fill rate may not be adequate. Nevertheless, for the exact model, first patterns can be observed: In general, increasing the uncertainty of any variable(s) leads to a lower fill rate. Moreover, as expected, the fill rate drops most (-1.39%) when the uncertainty in all variables increases (H/H/H). Further, increased lead time uncertainty seems to cause more stock outs. This is based on the observation that both scenarios H/H/L and L/H/H (both high lead time uncertainty) result in relatively high decreases (-0.70% and -0.98%), but the scenario H/L/H (low lead time uncertainty) does not (-0.28%).

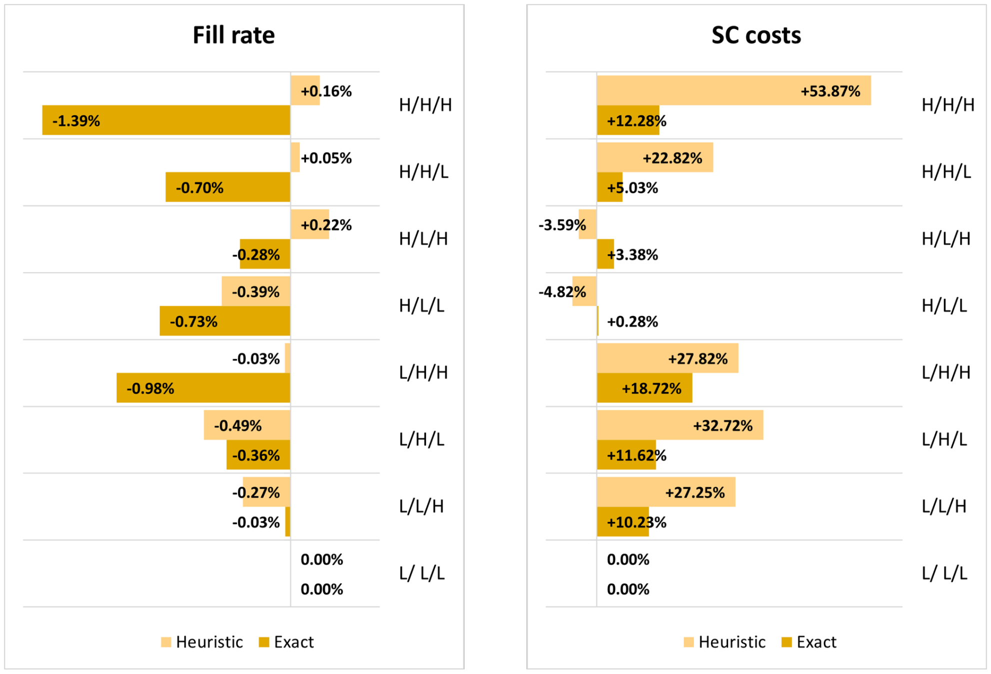


Figure 16: Sensitivity of SC fill rate (left) and costs (right) for different scenarios of uncertainty

Changing the level of uncertainty has mainly an impact on the total SC costs. Let us first consider the cases in which the uncertainty of one single variable is increased: Only dealing with higher demand uncertainty (H/L/L) appears to have the least negative effect on costs. The SC costs increase by 3.38% for the exact model and even decrease by 3.59% in case of the heuristic. In contrast, increased lead time uncertainty (L/H/L) results in the greatest cost increase (+32.72% for the exact model; +11.62% for the heuristic). A higher market price uncertainty (L/L/H) has a similar but slightly lower effect. Focusing on the increase of two variables, a similar pattern to that of the fill rate can be observed: In the scenarios where lead time is one of the increased variables, clearly higher costs can be detected. This especially holds for the heuristic, which leads to cost increases of 27.82% and 22.82% in the scenarios L/H/H and H/H/L, respectively. This confirms the assumption that high lead uncertainty may be the most decisive factor for the success of a fuzzy rule-based inventory control model. Finally, note that an overall high uncertainty (H/H/H) leads to a high cost surge in both models. For the heuristic, it even causes an increase of more than 50%.

6.3 Results - Real-world case study

It is important to realise that, thus far, the performance of the proposed FRBSs has been analysed in perfect conditions. In this part, it is tested if fuzzy control is still applicable to large networks in which data may not be perfectly distributed as expected. For this purpose, a fuzzy control system for managing the supply chain of the German cereal producer, as introduced in Section 6.1, was created. Due to the high complexity of the network (58 inventories), the exact approach as shown in Section 5.5 is not applicable. Instead, the heuristic approach is applied by using the same parameter setting as defined in 6.2.1 except for the following two modifications: (1) a 75/25 training/test split ratio is used, and (2) the LA-tuning algorithm is terminated after 250 iterations. Moreover, the real inventory data of the company is used as input. For each input variable, the data includes 220 data points collected during a complete year (only working days). Table 9 displays the final results of the training and testing stage.

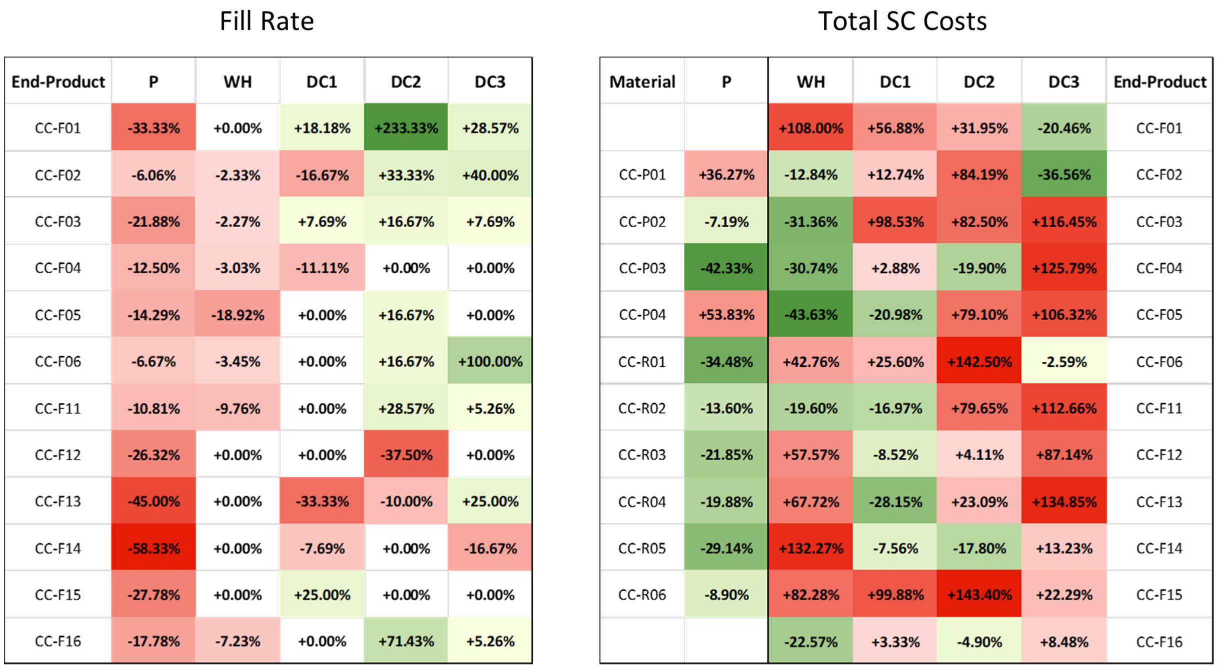
Method (stage)	f(x)	FR	C	C_H	C_L	C_O	C_S	C_F	C_P	C_U
EOQ (TRAIN)	0.688	0.860	32,616	23,319	1,855	885	264	1,334	4,106	1,022
Heuristic (TRAIN)	0.738	0.949	36,243	26,032	2,610	1,671	464	1,530	4,655	1,158
EOQ (TEST)	0.683	0.853	11,291	7,793	688	260	78	504	1,566	401
Heuristic (TEST)	0.710	0.925	13,150	9,146	1,220	638	179	617	1,707	473

Table 9: Final results of training and testing EOQ model and heuristic (costs in KEUR)

Table 9 confirms the deficiencies of the currently used local order policy (EOQ). The company experiences a fairly low fill rate of 0.86 (0.853) during the training respectively test phase. On the other hand, fuzzy rule-based control seems to manage the inventory more reliably given this vague starting position. As can be seen in Table 9, the derived FRBS achieves a test fill rate of 0.925, which is 8.5% higher as compared to the EOQ method. However, as previously found, this happens at the cost of higher overall SC costs. With fuzzy control, the company has to pay around 16.5% more (13.150M euros) than using the EOQ method (11.291M euros). Nevertheless, given an equal prioritisation of customer satisfaction and SC costs, the fuzzy heuristic performs better than the EOQ as it scores a fitness value of 0.738 (0.710) in comparison to 0.688 (0.683) for the EOQ model.

However, it is important to recognise that the results of the test stage are based on one single test data set and should therefore be interpreted carefully. Nonetheless, the simulation experiment of the last chapter (Table 8) showed that the range of observed fill rates and SC costs is fairly small. Hence, the test results in Table 9 serve as a good reference point. The more important result of this case study is that the proposed 3-step procedure in combination with the heuristic FRBS is still applicable to larger networks in which uncertainty and vagueness are faced.

Another interesting insight related to the working principles of the proposed FRBS is offered in Figure 17. It displays the differences (in percent) in the test fill rates (left) and the SC costs (right) that result from applying the FRBS to each inventory separately, whereby a higher color intensity stands for a higher improvement (green) or decline (red). Note that the corresponding absolute values as well as the individual cost components per inventory are provided in Appendix C. Also, note that in connection with a raw material inventory, a fill rate would not make sense, as a plant inventory solely delivers finished goods to its warehouses. Therefore, for plants, fill rates refer to end-products (e.g., P, CC-F01), whereas the total SC costs are related to the actual raw material inventories (e.g., P, CC-P01).


Fill Rate

End-Product	P	WH	DC1	DC2	DC3
CC-F01	-33.33%	+0.00%	+18.18%	+233.33%	+28.57%
CC-F02	-6.06%	-2.33%	-16.67%	+33.33%	+40.00%
CC-F03	-21.88%	-2.27%	+7.69%	+16.67%	+7.69%
CC-F04	-12.50%	-3.03%	-11.11%	+0.00%	+0.00%
CC-F05	-14.29%	-18.92%	+0.00%	+16.67%	+0.00%
CC-F06	-6.67%	-3.45%	+0.00%	+16.67%	+100.00%
CC-F11	-10.81%	-9.76%	+0.00%	+28.57%	+5.26%
CC-F12	-26.32%	+0.00%	+0.00%	-37.50%	+0.00%
CC-F13	-45.00%	+0.00%	-33.33%	-10.00%	+25.00%
CC-F14	-58.33%	+0.00%	-7.69%	+0.00%	-16.67%
CC-F15	-27.78%	+0.00%	+25.00%	+0.00%	+0.00%
CC-F16	-17.78%	-7.23%	+0.00%	+71.43%	+5.26%

Total SC Costs

Material	P	WH	DC1	DC2	DC3	End-Product
		+108.00%	+56.88%	+31.95%	-20.46%	CC-F01
CC-P01	+36.27%	-12.84%	+12.74%	+84.19%	-36.56%	CC-F02
CC-P02	-7.19%	-31.36%	+98.53%	+82.50%	+116.45%	CC-F03
CC-P03	-42.33%	-30.74%	+2.88%	-19.90%	+125.79%	CC-F04
CC-P04	+53.83%	-43.63%	-20.98%	+79.10%	+106.32%	CC-F05
CC-R01	-34.48%	+42.76%	+25.60%	+142.50%	-2.59%	CC-F06
CC-R02	-13.60%	-19.60%	-16.97%	+79.65%	+112.66%	CC-F11
CC-R03	-21.85%	+57.57%	-8.52%	+4.11%	+87.14%	CC-F12
CC-R04	-19.88%	+67.72%	-28.15%	+23.09%	+134.85%	CC-F13
CC-R05	-29.14%	+132.27%	-7.56%	-17.80%	+13.23%	CC-F14
CC-R06	-8.90%	+82.28%	+99.88%	+143.40%	+22.29%	CC-F15
		-22.57%	+3.33%	-4.90%	+8.48%	CC-F16

Figure 17: Test fill rates and total SC costs per inventory

Figure 17 shows that fuzzy multi-echelon control increases the fill rate of direct customer orders while reducing the fill rate of orders at higher echelons (especially plants) at the same time. Hence, the automatic KB generation method learns that it is not always necessary to maintain high stock levels at high echelons in order to score high customer fill rates. This is valuable since practitioners usually tend to overestimate the required inventory levels at higher echelon inventories (e.g., central warehouses) (Axsäter, 2015).

All in all, the application of the heuristic approach to a real-world example confirmed the findings of the simulation experiment. Fuzzy control led to higher customer satisfaction at the expense of slightly higher but tolerable total costs.

6.4 Runtimes

This section presents the runtimes of both proposed methods (exact FRBS and heuristic FRBS) being applied to the simulation experiment and the real-world application. Table 10 shows the runtimes (in s) of both the exact FRBS and the heuristic FRBS for each scenario of uncertainty in the simulation experiment. The application of the heuristic FRBS to the real-world problem took 40,096 seconds in total, which is approximately 11 hours.

Method	L/L/L	L/L/H	L/H/L	L/H/H	H/L/L	H/L/H	H/H/L	H/H/H
Exact	4664	4388	4494	4437	4364	4439	4411	4491
Heuristic	2823	2688	2637	2671	2632	2710	2719	2766

Table 10: Runtimes (in s) of exact FRBS and heuristic for each uncertainty scenario of the simulation

The presented runtimes (Table 10) were obtained by a machine with a CPU of 1.8 GHz and 4 cores. Moreover, the code for conducting the simulation experiment and solving the real-world problem was written in Java.

7 Discussion and Conclusions

This thesis addresses the question of whether fuzzy rule-based control is globally applicable to general multi-echelon inventory systems as usually found in practice. One main problem that is faced in practice is that the high complexity of such multi-echelon systems makes it immensely difficult for human experts to define suitable fuzzy rule-based systems (FRBSs).

As a solution to this, we present a new general procedure for automatically generating fuzzy rule-based systems that meet our specified requirements for a future application to real-world problems: (1) the fuzzy control model should be cost-effective while meeting customer satisfaction requirements, regardless of the level of uncertainty; (2) it should be applicable to a wide range of network sizes and offer reasonable computation times. To realise this, our model includes a new formulation of a global (exact) multi-echelon FRBS (MEF) and a newly developed heuristic that relies on a limited exchange of information but is considerably faster. We use a modified version of Wang and Mendel's method (WM) to create initial fuzzy knowledge bases (KBs), a simulated annealing algorithm (SA) to optimise the fuzzy rule outputs, and a genetic algorithm (LA-tuning) to tune the membership functions of the fuzzy data base (DB).

To test whether our models meet the requirements for applicability, we applied both proposed models to eight different scenarios of uncertainty in a controlled simulation experiment. The results were compared to a simple local order policy that is based on the classical economic order quantity (EOQ). The main metrics for comparison were chosen to be the total SC costs, the SC fill rate, and most importantly, a fitness value that combines both the fill rate and the SC costs.

The evaluation has shown that even for a small number of 100 iterations, the procedure can derive highly reliable control systems. Both the exact FRBS and the heuristic led to close-to-perfect fill rates (>=0.985), regardless of the level of uncertainty that was faced. However, the high reliability came at the expense of higher total SC costs. The proposed models led to higher total SC costs in 4 out of 8 uncertainty scenarios. A sensitivity analysis of the costs revealed that higher costs solely occurred in the presence of high lead time uncertainty. Hence, compared to a local order policy, the application of fuzzy rule-based inventory control systems can lead to lower total inventory costs while clearly surpassing customer satisfaction requirements, provided that the lead time uncertainty is not too high. However, the analysis was performed with an equal prioritisation of costs and customer satisfaction. By increasing the weight of the SC costs, the models might also be suitable for systems with high lead time uncertainty.

Regarding the question, if the proposed models can be applied to networks of any size while being computationally efficient at the same time, clear conclusions can be drawn: Since the exact model considers information along the whole supply chain in each given review period, the search spaces of the SA algorithm and the LA-tuning algorithm grow immensely with the number of inventories. This makes the exact model not applicable to networks with a high number of inventories. In particular, the LA-tuning algorithm acts as a bottleneck for the computation time. This is mainly because the number of iterations as well as the time spent per iteration increases when the number of fuzzy variables is increased. The number of fuzzy variables, in turn, is directly related to the number of inventories in the system. However, the data suggests that the LA-tuning algorithm works reasonably well, even for a small number of iterations. Therefore, in practice, the total computation time of the exact approach can easily be estimated by determining the time spent per iteration of the LA-tuning algorithm. An application for up to 10 inventories is seen as realistic.

The heuristic, however, can be applied to significantly larger supply chains. As the procedure for generating a FRBS is applied to each inventory separately, the computation time only increases linearly with the number of inventories. We tested the heuristic on an artificial data set that describes the manufacturing supply chain of a German cereal producer (58 inventories). In short, the heuristic increased the fill rate from 0.853 (EOQ) to 0.925 at the expense of higher (+16.5%) but tolerable SC costs and thus slightly outperformed the current local order policy. Hence, fuzzy control was successfully applied to a network of real-world dimensions. As a result, we can conclude that fuzzy rule-based inventory control systems are applicable to real-world problems involving SCs that rely on simple local order policies, such as the EOQ method discussed in this thesis. Fuzzy control is especially useful for systems that operate in a rapidly changing environment or have scarce, vague, imperfectly measured, or incomplete data and therefore rely on expert knowledge. However, it is important to note that for stationary systems, there may be more advanced multi-echelon order policies that are superior to fuzzy rule-based inventory systems.

This study has some limitations that should be taken into account in future applications: First, since optimal order policies for multi-echelon inventory systems are usually not known in advance, the current models do not consider the accuracy of a policy during the optimisation process. Instead, fuzzy knowledge bases are derived based on a single realisation of training data, which might limit a model's generalisation ability. Second, in the current models, the universes of discourses of the fuzzy variables are solely based on the minimal and maximal values that are observed in past data. This might pose a problem if a variable (e.g., market price) has a clear trend (positive or negative), as future values might fall out of the range. This issue can be resolved by applying the entire generation procedure more frequently or by readjusting the boundaries of the universes of discourse based on forecasts. Third, note that the fuzzy rules for a particular inventory are solely based on observations from the past. Hence, if the training data is not sufficient, this might lead to missing rules, which in turn might lower a model's performance. Fourth, due to the high computation times, no grid-based parameter tuning was performed. Instead, an acceptable combination of parameters was derived by trial and error and practical considerations. Also, note that the same tuning parameters have been applied to all inventories. By tuning parameters for each inventory separately, the performance of the models might be improved further. Finally, in the heuristic, for each inventory, it is assumed that its predecessor has sufficient stock for an order. This means that fuzzy rules are optimised for these perfect scenarios. As a result, in the presence of high uncertainty and thereby arising stock outs, fuzzy rules may no longer be optimal.

All in all, fuzzy logic proves to be a promising tool for reliable and cost-effective inventory control of multi-echelon inventory systems and, therefore, should be further developed in the future. Future research should be devoted to the extension of FRBSs from arborescent to general multi-echelon inventory systems. This would require the development of multi-input multi-output FRBSs, as inventories of general inventory systems can have multiple predecessors and thus can set multiple orders. Another possible extension is a multi-echelon FRBS in which the number of fuzzy labels can vary for different fuzzy variables and inventories. This would be a major advancement as literature suggests that the number of fuzzy labels is decisive for the success of a fuzzy model. Last but not least, a possible extension might be finding a method for extracting the fuzzy input variables that matter most (e.g., by principal component analysis (PCA) or Lasso regression). By reducing the number of fuzzy input variables, global fuzzy rule-based inventory control models could be applicable to networks of higher complexity. A successful implementation of the aforementioned extensions may eventually answer the fundamental question of whether fuzzy rule-based systems are capable of outperforming other multi-echelon inventory policies in any given environment, thereby providing an all-around solution for the industry.

References

Aengchuan, P., & Phruksaphanrat, B. (2018). Comparison of fuzzy inference system (FIS), FIS with artificial neural networks (FIS + ANN) and FIS with adaptive neuro-fuzzy inference system (FIS + ANFIS) for inventory control. *Journal of Intelligent Manufacturing*, *29*(4), 905–923.

Alcalá, R., Alcalá-Fdez, J., Gacto, M. J., & Herrera, F. (2007). Rule base reduction and genetic tuning of fuzzy systems based on the linguistic 3-tuples representation. *Soft Computing*, *11*(5), 401–419.

Axsäter, S. (2015). *Inventory control* (Vol. 225). Springer.

Ben-Ameur, W. (2004). Computing the initial temperature of simulated annealing. *Computational Optimization and Applications*, *29*(3), 369–385.

Bessler, S. A., & Veinott Jr, A. F. (1966). Optimal policy for a dynamic multi-echelon inventory model. *Naval Research Logistics Quarterly*, *13*(4), 355–389.

Bezdek, J. C. (1994). The thirsty traveler visits Gamont: A rejoinder to "Comments on fuzzy sets — What are they and why?". *IEEE Transactions on Fuzzy Systems*, *2*(1), 43–45.

Bonissone, P. P., Khedkar, P. S., & Chen, Y. (1996). Genetic algorithms for automated tuning of fuzzy controllers: A transportation application. In *Proceedings of IEEE 5th International Fuzzy Systems* (Vol. 1, pp. 674–680).

Chapman, S., Gatewood, A. K., Arnold, T. K., & Clive, L. (2016). *Introduction to materials management, eBook*. Pearson Higher Ed.

Chen, G., Pham, T. T., & Boustany, N. (2001). Introduction to fuzzy sets, fuzzy logic, and fuzzy control systems. *Applied Mechanics Reviews*, *54*(6), B102–B103.

Christopher, M., & Peck, H. (2004). Building the resilient supply chain. *The International Journal of Logistics Management*, *15*, 1-14.

Clark, A. J., & Scarf, H. (1960). Optimal policies for a multi-echelon inventory problem. *Management Science*, *6*(4), 475–490.

Cordón, O. (2011). A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems. *International Journal of Approximate Reasoning*, *52*(6), 894–913.

Cordón, O., Herrera, F., & Villar, P. (2000). Analysis and guidelines to obtain a good uniform fuzzy partition granularity for fuzzy rule-based systems using simulated annealing. *International Journal of Approximate Reasoning*, *25*(3), 187–215.

Cordón, O., Herrera, F., & Villar, P. (2001). Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base. *IEEE Transactions on Fuzzy Systems*, *9*(4), 667–674.

Dubois, D., & Prade, H. (1996). What are fuzzy rules and how to use them. *Fuzzy Sets and Systems*, *84*(2), 169–185.

Dubois, D., Prade, H., & Sandri, S. (1993). On possibility/probability transformations. In *Fuzzy Logic* (pp. 103–112). Springer.

Eiselt, H. A., & Sandblom, C.-L. (2012). *Operations research: A model-based approach.* Springer Science & Business Media.

Erlenkotter, D. (1990). Ford Whitman Harris and the economic order quantity model. *Operations Research*, *38*, 937-946.

Eshelman, L. J. (1991). The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In *Foundations of Genetic Algorithms* (Vol. 1, pp. 265–283). Elsevier.

Geevers, K. (2020). *Deep reinforcement learning in inventory management* (Unpublished master's thesis). University of Twente.

Giannoccaro, I., Pontrandolfo, P., & Scozzi, B. (2003). A fuzzy echelon approach for inventory management in supply chains. *European Journal of Operational Research*, *149*(1), 185–196.

Gümüs, A. T., & Güneri, A. F. (2007). Multi-echelon inventory management in supply chains with uncertain demand and lead times: Literature review from an operational research perspective. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *221*(10), 1553–1570.

Herrera, F., Lozano, M., & Sánchez, A. M. (2003). A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. *International Journal of Intelligent Systems*, *18*(3), 309–338.

Ignaciuk, P., & Wieczorek, Ł. (2020). Continuous genetic algorithms in the optimization of logistic networks: Applicability assessment and tuning. *Applied Sciences*, *10*(21), 7851.

Kayacan, E., & Khanesar, M. A. (2015). *Fuzzy neural networks for real time control applications: Concepts, modeling and algorithms for fast learning.* Butterworth-Heinemann.

Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4598), 671–680.

Ledesma, S., Aviña, G., & Sanchez, R. (2008). Practical considerations for simulated annealing implementation. *Simulated Annealing*, *20*, 401–420.

Léger, P.-M. (2006). *Using a simulation game approach to teach ERP concepts.* HEC Montréal, Groupe de Recherche en Systèmes D'information.

Macleod, M. D. (1993). 14 - Coding. In F. Mazda (Ed.), *Telecommunications Engineer's Reference Book* (p. 14-1-14-13). Butterworth-Heinemann.

Mamdani, E. H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, *7*(1), 1–13.

Moheb-Alizadeh, H., & Handfield, R. (2018). The impact of raw materials price volatility on cost of goods sold (cogs) for product manufacturing. *IEEE Transactions on Engineering Management*, *65*(3), 460–473.

Park, K. S. (1987). Fuzzy-set theoretic interpretation of economic order quantity. *IEEE Transactions on Systems, Man, and Cybernetics*, *17*(6), 1082–1084.

Petrovic, D. (2001). Simulation of supply chain behaviour and performance in an uncertain environment. *International Journal of Production Economics*, *71*(1-3), 429–438.

Petrovic, D., Roy, R., & Petrovic, R. (1999). Supply chain modelling using fuzzy sets. *International Journal of Production Economics*, *59*(1-3), 443–453.

Petrovic, D., & Sweeney, E. (1994). Fuzzy knowledge-based approach to treating uncertainty in inventory control. *Computer Integrated Manufacturing Systems*, *7*(3), 147–152.

Riza, L. S., Bergmeir, C., Herrera, F., & Benítez, J. M. (2015). FRBS: Fuzzy rule-based systems for classification and regression in R. *Journal of Statistical Software*, *65*(6), 1–30.

Rotshtein, A., & Rakityanskaya, A. (2006). Inventory control as an identification problem based on fuzzy logic. *Cybernetics and Systems Analysis*, *42*(3), 411–419.

Sadollah, A. (2018). *Introductory chapter: Which membership function is appropriate in fuzzy system?* IntechOpen.

Salgado-Plasencia, E., Carrillo-Serrano, R. V., & Toledano-Ayala, M. (2020). Development of a DSP microcontroller-based fuzzy logic controller for heliostat orientation control. *Applied Sciences*, *10*(5), 1598.

Salleh, M. N. M., Talpur, N., & Hussain, K. (2017). Adaptive neuro-fuzzy inference system: Overview, strengths, limitations, and solutions. In *International Conference on Data Mining and Big Data* (pp. 527–535).

Samanta, B., & Al-Araimi, S. A. (2003). Application of an adaptive neuro-fuzzy inference system in inventory control. *International Journal of Smart Engineering System Design*, *5*(4), 547–553.

Sherbrooke, C. C. (1968). METRIC: A multi-echelon technique for recoverable item control. *Operations Research*, *16*(1), 122–141.

Tanthatemee, T., & Phruksaphanrat, B. (2012). Fuzzy inventory control system for uncertain demand and supply. In *Proceedings of the International Multiconference of Engineers and Computer Scientists* (pp. 1224–1229).

Wang, K. (2001). Computational intelligence in agile manufacturing engineering. *Agile Manufacturing: The 21st Century Competitive Strategy, Oxford, UK: Elsevier Science Ltd*, 297–315.

Wang, L.-X. (1999). *A course in fuzzy systems.* Prentice-Hall press, USA.

Wang, L.-X., & Mendel, J. M. (1992). Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics*, *22*(6), 1414–1427.

Weicker, K. (2015). *Evolutionäre Algorithmen.* Springer-Verlag.

Weyland, D. (2008). Simulated annealing, its parameter settings and the longest common subsequence problem. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation* (pp. 803–810).

Whitley, D., & Sutton, A. M. (2012). Genetic algorithms - A survey of models and methods. In *Handbook of Natural Computing* (pp. 637–671). Springer Berlin Heidelberg.

Wiecek, P. (2016). Intelligent approach to inventory control in logistics under uncertainty conditions. *Transportation Research Procedia*, *18*, 164–171.

Xiong, G., & Helo, P. (2006). An application of cost-effective fuzzy inventory controller to counteract demand fluctuation caused by bullwhip effect. *International Journal of Production Research*, *44*(24), 5261–5277.

Zadeh. (1965). Fuzzy sets. *Information and Control*, *8*(3), 338-353.

Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*(1), 28–44.

Zadeh, L. A. (1975). Fuzzy logic and approximate reasoning. *Synthese*, *30*(3), 407–428.

Zadeh, L. A. (1988). Fuzzy logic. *Computer*, *21*(4), 83–93.

Zafari, A. (2014). *Developing a fuzzy inference system by using genetic algorithms and expert knowledge: With a case study for landslides in Iran* (Unpublished master's thesis). University of Twente.

Zalnezhad, E., & Sarhan, A. (2015). Fuzzy modeling to predict the adhesion strength of TiN ceramic thin film coating on aerospace AL7075-T6 alloy. In *Recent Advances in Structural Integrity Analysis-Proceedings of the International Congress (APCF/SIF-2014):(APCFS/SIF 2014)* (p. 239).

Zheng, L. (1992). A practical guide to tune of proportional and integral (PI) like fuzzy controllers. In *[1992 Proceedings] IEEE International Conference on Fuzzy Systems* (pp. 633–640).

Appendices

This chapter summarises the additional information related to the case study of this thesis. Due to the great volume of available information, the actual data sets are publicly uploaded on GitHub and can be accessed via:

`www.github.com/piperovilia/Fuzzy-ME-inventory-model-with-automatic-KB-generation`

The data includes a detailed description of the supply chain network (locations, materials, etc.), the input data (costs, initial inventory levels, orders, etc.), and a collection of relevant results related to the simulation experiment and the real-world application conducted in this thesis.

A End-products, raw materials and SC nodes

This section lists the locations, end-products, and raw (packaging) materials that are part of the SC of a German cereal manufacturer used in the case study of this thesis. Note that a detailed description of the individual table fields as well as additional input data related to the case study can be found in Appendix B.

id	stage_type	description
P	PLANT	Plant
02	WH	Warehouse
02W	DIST	Distributor west
02N	DIST	Distributor north
02S	DIST	Distributor south
V1	VD	Vendor 1 (Raw materials)
V2	VD	Vendor 2 (Packaging materials)
GC02N	CUST	Grocery chain (belonging to DIST 02N)
IG02N	CUST	Independent grocer (belonging to DIST 02N)
H02N	CUST	Hypermarket (belonging to DIST 02N)
GC02W	CUST	Grocery chain (belonging to DIST 02W)
IG02W	CUST	Independent grocer (belonging to DIST 02W)
H02W	CUST	Hypermarket (belonging to DIST 02W)
GC02S	CUST	Grocery chain (belonging to DIST 02S)
IG02S	CUST	Independent grocer (belonging to DIST 02S)
H02S	CUST	Hypermarket (belonging to DIST 02S)

Table 11: Specification of the SC nodes of a German cereal producer

material_code	type_code	unit_size	description
CC-F01	FERT	0.5	500g Nut Muesli
CC-F02	FERT	0.5	500g Blueberry Muesli
CC-F03	FERT	0.5	500g Strawberry Muesli
CC-F04	FERT	0.5	500g Raisin Muesli
CC-F05	FERT	0.5	500g Original Muesli
CC-F06	FERT	0.5	500g Mixed Fruit Muesli
CC-F11	FERT	1	1kg Nut Muesli
CC-F12	FERT	1	1kg Blueberry Muesli
CC-F13	FERT	1	1kg Strawberry Muesli
CC-F14	FERT	1	1kg Raisin Muesli
CC-F15	FERT	1	1kg Original Muesli
CC-F16	FERT	1	1kg Mixed Fruit Muesli
CC-P01	ROH	1	Large box (1kg)
CC-P02	ROH	1	Large bag (1kg)
CC-P03	ROH	1	Small box (500g)
CC-P04	ROH	1	Small bag (500g)
CC-R01	ROH	1	Nuts
CC-R02	ROH	1	Blueberries
CC-R03	ROH	1	Strawberries
CC-R04	ROH	1	Raisins
CC-R05	ROH	1	Wheat
CC-R06	ROH	1	Oats

Table 12: Specification of the end-products and raw (packaging) materials of a German cereal producer

B Input data

This section explains the available input information related to the case study of this thesis. Table 13 contains a technical description of each given text file. The corresponding files can be found in the root directory of the repository uploaded on GitHub.

Table name	Rows	Fields	Table description
Arcs	15	**from:** Origin of arc **to:** Destination of arc	Contains the arcs (direct connection of two locations) of the supply chain network. The nodes (locations) are defined by their inventory IDs.
BOM (BOM_Sim)	12 (1)	**rows:** End-products **columns:** Raw (packaging) materials	Contains the bills of materials of all end-products for the real-world application (BOM) respectively the simulation experiment (BOM_Sim). For packaging materials (CC-PXX), the value 1 indicates that a material is a component and 0 otherwise. For raw materials (CC-RXX), the provided quantity ($[0, 1]$) is the share of the total product weight - e.g., for product CC-F01 (0.5kg), 0.3 means that 0.15kg (0.5*0.3) of the raw material are needed.
Costs (Costs_Sim)	188 (18)	**inventory_id:** ID of location **material_code:** Product/material ID **cost_type** = $\{$h (holding cost), b (penalty cost), c_o (order cost), c_s (setup cost), c_p (production cost)$\}$ **cost:** Amount in EUR **unit:** per unit, per order, etc.	Contains information about the individual cost components for the real-world application (Costs) respectively the simulation experiment (Costs_Sim).
Dates	225	**date:** Working day	Contains all the dates relevant for the real-world application.
Init_Inv_Levels (Init_Inv_Levels_Sim)	58 (58)	**inventory:** Identifies an inventory - an inventory is defined by an *inventory_id* and a *material_code*, e.g., 02,CC-F01 **quantity:** Inventory level	Contains the initial inventory levels of each inventory for the real-world application (Init_Inv_Levels) respectively the simulation experiment (Init_Inv_Levels_Sim).

(Continued on next page)

Table name	Rows	Fields	Table description
Materials (Materials_Sim)	22 (3)	**material_code:** Product/material ID **type_code** = {FERT (end-product), ROH (raw material)} **unit_size:** Share of corresponding unit (e.g., 0.5kg nut muesli, 1 bag, 1kg nuts) **description:** Definition of material	Contains information about the end-products and raw and packaging materials for the real-world application (Materials) respectively the simulation experiment (Materials_Sim).
Nodes	16	**id:** Location ID **stage_type** = {VD (vendor), PLANT (production site), WH (warehouse), DIST (distribution center), CUST (end-customer)} **description:** Text describing node	Contains the locations of the supply chain network.
Orders	9763	**order_number:** Order ID **stage_type:** Inventory type of order **from:** Location placing order **to:** Location receiving order **material_code:** Material of order **quantity:** Ordered amount **start_date:** Order date **end_date:** Date of fulfilment **lead_time:** Order lead time in days **price:** Price of material on order date	Contains all orders being placed between inventories (only for real-world application).
Purchases	217	**material_code:** Product/material ID **date:** Date of purchase **price:** Price per unit	Contains the prices to which raw and packaging materials have been ordered (only for real-world application).
Transport_Costs (Transport_Costs_Sim)	166 (15)	**from:** Origin of transport **to:** Destination of transport **material_code:** Product/material ID **cost:** Amount in EUR **unit:** per unit	Contains the transport costs between all pairs of inventories in the SC for the real-world application (Transport_Costs) respectively the simulation experiment (Transport_Costs_Sim).

Table 13: Documentation of input data

C End results per inventory (real-world application)

SC costs - EOQ (in KEUR)

Material	P	02	02N	02W	02S	End-Product
		188	84	118	106	CC-F01
CC-P01	329	230	89	107	111	CC-F02
CC-P02	173	327	82	96	80	CC-F03
CC-P03	388	238	83	74	70	CC-F04
CC-P04	183	246	75	109	103	CC-F05
CC-R01	305	264	101	111	113	CC-F06
CC-R02	660	336	89	134	104	CC-F11
CC-R03	645	343	135	172	152	CC-F12
CC-R04	288	281	133	101	110	CC-F13
CC-R05	427	296	85	85	136	CC-F14
CC-R06	531	238	101	106	104	CC-F15
		434	145	174	165	CC-F16

SC costs - Heuristic FRBS (in KEUR)

Material	P	02	02N	02W	02S	End-Product
		392	132	155	84	CC-F01
CC-P01	448	200	101	197	71	CC-F02
CC-P02	160	225	163	175	174	CC-F03
CC-P03	224	165	85	60	158	CC-F04
CC-P04	281	138	59	196	212	CC-F05
CC-R01	200	376	126	270	110	CC-F06
CC-R02	570	270	74	240	222	CC-F11
CC-R03	504	540	124	179	284	CC-F12
CC-R04	231	472	95	124	257	CC-F13
CC-R05	303	687	79	70	154	CC-F14
CC-R06	484	433	203	258	127	CC-F15
		336	150	166	179	CC-F16

Fill rate – EOQ

End-Product	P	02	02N	02W	02S
CC-F01	1.00	1.00	0.85	0.30	0.64
CC-F02	1.00	1.00	0.86	0.75	0.67
CC-F03	1.00	1.00	0.93	0.86	0.93
CC-F04	1.00	1.00	1.00	1.00	1.00
CC-F05	1.00	1.00	1.00	0.86	1.00
CC-F06	1.00	1.00	1.00	0.86	0.50
CC-F11	1.00	1.00	1.00	0.78	0.90
CC-F12	1.00	1.00	0.89	0.80	1.00
CC-F13	1.00	1.00	0.75	1.00	0.80
CC-F14	1.00	1.00	1.00	1.00	0.80
CC-F15	1.00	1.00	0.80	1.00	1.00
CC-F16	1.00	1.00	0.88	0.54	0.86

Fill rate - Heuristic FRBS

End-Product	P	02	02N	02W	02S
CC-F01	0.67	1.00	1.00	1.00	0.82
CC-F02	0.94	0.98	0.71	1.00	0.93
CC-F03	0.78	0.98	1.00	1.00	1.00
CC-F04	0.88	0.97	0.89	1.00	1.00
CC-F05	0.86	0.81	1.00	1.00	1.00
CC-F06	0.93	0.97	1.00	1.00	1.00
CC-F11	0.89	0.90	1.00	1.00	0.95
CC-F12	0.74	1.00	0.89	0.50	1.00
CC-F13	0.55	1.00	0.50	0.90	1.00
CC-F14	0.42	1.00	0.92	1.00	0.67
CC-F15	0.72	1.00	1.00	1.00	1.00
CC-F16	0.82	0.93	0.88	0.92	0.91

Holding costs - EOQ (in KEUR)

Material	P	02	02N	02W	02S	End-Product
		146	59	42	61	CC-F01
CC-P01	171	154	63	72	54	CC-F02
CC-P02	98	230	66	69	65	CC-F03
CC-P03	248	158	79	70	67	CC-F04
CC-P04	104	187	71	93	99	CC-F05
CC-R01	221	211	97	98	71	CC-F06
CC-R02	356	211	83	104	68	CC-F11
CC-R03	386	236	115	133	146	CC-F12
CC-R04	201	218	116	96	93	CC-F13
CC-R05	219	189	80	81	91	CC-F14
CC-R06	239	206	85	103	102	CC-F15
		289	113	94	117	CC-F16

Holding costs - Heuristic FRBS (in KEUR)

Material	P	02	02N	02W	02S	End-Product
		298	123	148	55	CC-F01
CC-P01	246	96	52	191	46	CC-F02
CC-P02	69	102	153	166	165	CC-F03
CC-P03	60	80	67	50	151	CC-F04
CC-P04	168	58	44	189	208	CC-F05
CC-R01	97	288	106	266	100	CC-F06
CC-R02	206	124	62	232	198	CC-F11
CC-R03	167	435	99	90	272	CC-F12
CC-R04	153	396	62	101	250	CC-F13
CC-R05	93	593	59	59	76	CC-F14
CC-R06	142	359	198	252	118	CC-F15
		135	110	126	134	CC-F16

Penalty costs - EOQ (in KEUR)

Material	P	02	02N	02W	02S	End-Product
		0	21	75	43	CC-F01
CC-P01	0	0	21	32	53	CC-F02
CC-P02	0	0	11	23	11	CC-F03
CC-P03	0	0	0	0	0	CC-F04
CC-P04	0	0	0	14	0	CC-F05
CC-R01	0	0	0	10	41	CC-F06
CC-R02	0	0	0	28	28	CC-F11
CC-R03	0	0	17	33	0	CC-F12
CC-R04	0	0	15	0	15	CC-F13
CC-R05	0	0	0	0	40	CC-F14
CC-R06	0	0	14	0	0	CC-F15
		0	26	78	39	CC-F16

Penalty costs - Heuristic FRBS (in KEUR)

Material	P	02	02N	02W	02S	End-Product
		0	0	0	21	CC-F01
CC-P01	0	0	43	0	11	CC-F02
CC-P02	0	0	0	0	0	CC-F03
CC-P03	0	0	12	0	0	CC-F04
CC-P04	0	0	0	0	0	CC-F05
CC-R01	0	0	0	0	0	CC-F06
CC-R02	0	0	0	0	14	CC-F11
CC-R03	0	0	17	83	0	CC-F12
CC-R04	0	0	30	15	0	CC-F13
CC-R05	0	0	13	0	66	CC-F14
CC-R06	0	0	0	0	0	CC-F15
		0	26	13	26	CC-F16

Figure 18: Comparison of EOQ method and heuristic FRBS per inventory (Part 1)

Order costs - EOQ (in EUR)

Material	P	02	02N	02W	02S	End-Product
		0	3200	640	1920	CC-F01
CC-P01	8000	0	4480	3200	3840	CC-F02
CC-P02	6400	0	4620	4620	3960	CC-F03
CC-P03	7200	0	4200	4200	3500	CC-F04
CC-P04	5600	0	3360	2520	3360	CC-F05
CC-R01	9000	0	3600	3000	1200	CC-F06
CC-R02	17000	0	6560	2460	8200	CC-F11
CC-R03	16000	0	4000	5000	6000	CC-F12
CC-R04	9000	0	1800	5400	1800	CC-F13
CC-R05	21000	0	5460	4680	5460	CC-F14
CC-R06	21000	0	1720	2580	1720	CC-F15
		0	6240	2340	9360	CC-F16

Order costs - Heuristic FRBS (in EUR)

Material	P	02	02N	02W	02S	End-Product
		0	8320	7040	7680	CC-F01
CC-P01	13200	0	6400	6400	14080	CC-F02
CC-P02	10400	0	9900	9240	9240	CC-F03
CC-P03	21600	0	6300	9100	7000	CC-F04
CC-P04	10000	0	15120	6720	3360	CC-F05
CC-R01	23000	0	19800	3600	10200	CC-F06
CC-R02	43000	0	12300	8200	9840	CC-F11
CC-R03	54000	0	8000	6000	12000	CC-F12
CC-R04	13000	0	2700	8100	7200	CC-F13
CC-R05	51000	0	7020	10920	12480	CC-F14
CC-R06	56000	0	5160	5160	8600	CC-F15
		0	14040	26520	19500	CC-F16

Setup costs - EOQ (in EUR)

Material	P	02	02N	02W	02S	End-Product
		3600	0	0	0	CC-F01
CC-P01	0	7200	0	0	0	CC-F02
CC-P02	0	6000	0	0	0	CC-F03
CC-P03	0	6000	0	0	0	CC-F04
CC-P04	0	4200	0	0	0	CC-F05
CC-R01	0	4800	0	0	0	CC-F06
CC-R02	0	10400	0	0	0	CC-F11
CC-R03	0	8000	0	0	0	CC-F12
CC-R04	0	6400	0	0	0	CC-F13
CC-R05	0	8000	0	0	0	CC-F14
CC-R06	0	2400	0	0	0	CC-F15
		11200	0	0	0	CC-F16

Setup costs - Heuristic FRBS (in EUR)

Material	P	02	02N	02W	02S	End-Product
		7200	0	0	0	CC-F01
CC-P01	0	18600	0	0	0	CC-F02
CC-P02	0	15000	0	0	0	CC-F03
CC-P03	0	16800	0	0	0	CC-F04
CC-P04	0	14400	0	0	0	CC-F05
CC-R01	0	16800	0	0	0	CC-F06
CC-R02	0	26400	0	0	0	CC-F11
CC-R03	0	11200	0	0	0	CC-F12
CC-R04	0	8800	0	0	0	CC-F13
CC-R05	0	4000	0	0	0	CC-F14
CC-R06	0	10400	0	0	0	CC-F15
		29600	0	0	0	CC-F16

Production costs - EOQ (in EUR)

Material	P	02	02N	02W	02S	End-Product
		20076	0	0	0	CC-F01
CC-P01	0	37192	0	0	0	CC-F02
CC-P02	0	50165	0	0	0	CC-F03
CC-P03	0	37234	0	0	0	CC-F04
CC-P04	0	25843	0	0	0	CC-F05
CC-R01	0	30721	0	0	0	CC-F06
CC-R02	0	60316	0	0	0	CC-F11
CC-R03	0	55949	0	0	0	CC-F12
CC-R04	0	33073	0	0	0	CC-F13
CC-R05	0	51567	0	0	0	CC-F14
CC-R06	0	15274	0	0	0	CC-F15
		86962	0	0	0	CC-F16

Production costs - Heuristic FRBS (in EUR)

Material	P	02	02N	02W	02S	End-Product
		48495	0	0	0	CC-F01
CC-P01	0	45371	0	0	0	CC-F02
CC-P02	0	57172	0	0	0	CC-F03
CC-P03	0	33065	0	0	0	CC-F04
CC-P04	0	30118	0	0	0	CC-F05
CC-R01	0	46226	0	0	0	CC-F06
CC-R02	0	62861	0	0	0	CC-F11
CC-R03	0	56551	0	0	0	CC-F12
CC-R04	0	41052	0	0	0	CC-F13
CC-R05	0	49517	0	0	0	CC-F14
CC-R06	0	37286	0	0	0	CC-F15
		109765	0	0	0	CC-F16

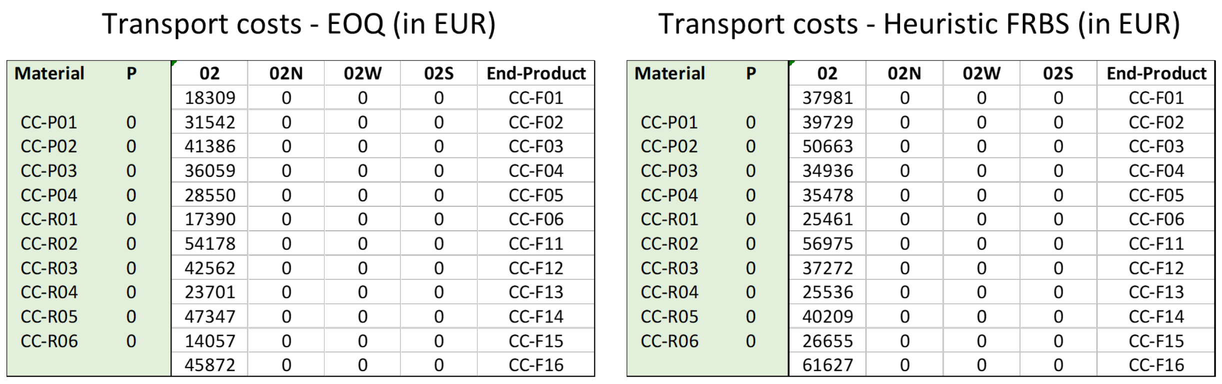
Purchase costs - EOQ (in EUR)

Material	P	02	02N	02W	02S	End-Product
		0	0	0	0	CC-F01
CC-P01	149057	0	0	0	0	CC-F02
CC-P02	68503	0	0	0	0	CC-F03
CC-P03	133281	0	0	0	0	CC-F04
CC-P04	72799	0	0	0	0	CC-F05
CC-R01	75779	0	0	0	0	CC-F06
CC-R02	287053	0	0	0	0	CC-F11
CC-R03	242967	0	0	0	0	CC-F12
CC-R04	78066	0	0	0	0	CC-F13
CC-R05	187594	0	0	0	0	CC-F14
CC-R06	271194	0	0	0	0	CC-F15
		0	0	0	0	CC-F16

Purchase costs - Heuristic FRBS (in EUR)

Material	P	02	02N	02W	02S	End-Product
		0	0	0	0	CC-F01
CC-P01	188494	0	0	0	0	CC-F02
CC-P02	80313	0	0	0	0	CC-F03
CC-P03	142106	0	0	0	0	CC-F04
CC-P04	103257	0	0	0	0	CC-F05
CC-R01	79718	0	0	0	0	CC-F06
CC-R02	321143	0	0	0	0	CC-F11
CC-R03	283631	0	0	0	0	CC-F12
CC-R04	64530	0	0	0	0	CC-F13
CC-R05	159032	0	0	0	0	CC-F14
CC-R06	285204	0	0	0	0	CC-F15
		0	0	0	0	CC-F16

Figure 19: Comparison of EOQ method and heuristic FRBS per inventory (Part 2)



Transport costs - EOQ (in EUR)

Material	P	02	02N	02W	02S	End-Product
		18309	0	0	0	CC-F01
CC-P01	0	31542	0	0	0	CC-F02
CC-P02	0	41386	0	0	0	CC-F03
CC-P03	0	36059	0	0	0	CC-F04
CC-P04	0	28550	0	0	0	CC-F05
CC-R01	0	17390	0	0	0	CC-F06
CC-R02	0	54178	0	0	0	CC-F11
CC-R03	0	42562	0	0	0	CC-F12
CC-R04	0	23701	0	0	0	CC-F13
CC-R05	0	47347	0	0	0	CC-F14
CC-R06	0	14057	0	0	0	CC-F15
		45872	0	0	0	CC-F16

Transport costs - Heuristic FRBS (in EUR)

Material	P	02	02N	02W	02S	End-Product
		37981	0	0	0	CC-F01
CC-P01	0	39729	0	0	0	CC-F02
CC-P02	0	50663	0	0	0	CC-F03
CC-P03	0	34936	0	0	0	CC-F04
CC-P04	0	35478	0	0	0	CC-F05
CC-R01	0	25461	0	0	0	CC-F06
CC-R02	0	56975	0	0	0	CC-F11
CC-R03	0	37272	0	0	0	CC-F12
CC-R04	0	25536	0	0	0	CC-F13
CC-R05	0	40209	0	0	0	CC-F14
CC-R06	0	26655	0	0	0	CC-F15
		61627	0	0	0	CC-F16

Figure 20: Comparison of EOQ method and heuristic FRBS per inventory (Part 3)

The end results per inventory related to the simulation experiment of this thesis are publicly uploaded on GitHub. A detailed explanation of the corresponding files (folder name "Final Results") is given in Appendix D.

D Additional results

This section provides an overview of all results available on GitHub, which are accessible via:

`www.github.com/piperovilia/Fuzzy-ME-inventory-model-with-automatic-KB-generation`

Figure 21 depicts how the information is organised into folders, whereas Table 15 provides a detailed explanation of the separate files contained therein.

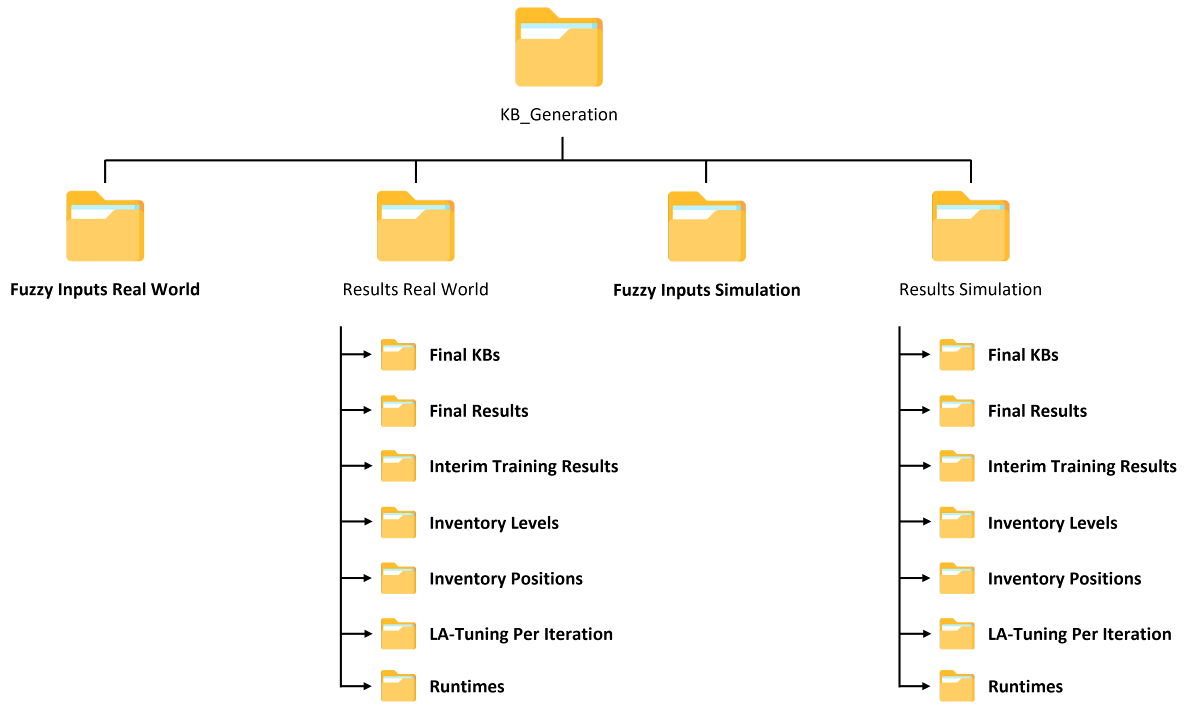


Figure 21: Folder structure of additional results

Moreover, Table 14 introduces a set of abbreviations that are used for identifying the files being presented in this additional results section. A thorough specification of the separate files follows in Table 15. Note that bold text in the file names represents dynamic text parts, i.e., parts that change from file to file, whereas regular text is static.

Abbreviation	Description
Inv	Inventory defined by "inventory_id,material_code"
Model	Type of method ("EOQ", "Exact" or "Heuristic")
S	Number of scenario (numbering corresponds to ordering presented in Section 6.2.1)
Stage	Time stage ("Training" or "Test")

Table 14: Abbreviations used in files of results section

Folder name	File name(s)	Description
Fuzzy Inputs Real World (Fuzzy Inputs Simulation)	inputData**Inv**,100.txt (inputData**Inv**,**S**.txt)	These files contain the actual input data for training the fuzzy models. Hence, they contain a value for each training date and fuzzy variable.
Final KBs	(1) dataBase**InvModel**Scenario**S**.txt (2) ruleBase**InvModel**Scenario**S**.txt	Data base files (1) contain all definition points of triangular MFs representing an inventory, whereas rule base files (2) contain the corresponding antecedents and outputs of the fuzzy rules. The definition points and rule antecedents correspond to the following order of fuzzy variables: demand, inventory position, lead time, market price, order quantity. Moreover, for a particular fuzzy variable, the definition points (left, middle, right) are provided in the order (low, medium, high). Irrelevant fuzzy variables are not included in the files for a particular inventory.
Final Results	Folder *Overall*: finalResults**ModelStage**Scenario**S**.txt Folder *Per inventory*: finalResultsPerInventory**ModelStage**Scenario**S**.txt	The folder *Overall* contains the final results as presented in Section 6. A file of the folder *Per inventory* lists the fill rate and the individual cost components for each inventory of the system.
Interim Training Results	(1) InterimTrainingResultsScenario**S**.txt (2) InterimTrainingResults**Inv**Scenario**S**.txt	These files contain the fitness value, total SC costs and fill rate after each step of the 3-step KB generation procedure (WM, SA, LA-tuning). A file of syntax (1) is generated by the exact FRBS, whereas a file of syntax (2) origins from the heuristic FRBS. Moreover, the results of the real-world application are marked with a scenario number of 100.
Inventory levels	In folder *Results Real World*: inventoryLevels**ModelStage**.txt In folder *Results Simulation*: inventoryLevels**ModelStage**,**S**.txt	These files contain the inventory levels (one per date) generated by a particular model for each inventory of the system.

(Continued on next page)

Folder name	File name(s)	Description
Inventory positions	In folder *Results Real World*: inventoryPositions**ModelStage**.txt In folder *Results Simulation*: inventoryPositions**ModelStage**,**S**.txt	These files contain the inventory positions (one per date) generated by a particular model for each inventory of the system.
LA-tuning Per Iteration	(1) ValuesPerIterationLAScenario**S**.txt (2) ValuesPerIterationLA**Inv**Scenario**S**.txt	These files contain the fitness values after each iteration of the LA-tuning algorithm. A file of syntax (1) is generated by the exact FRBS, whereas a file of syntax (2) origins from the heuristic FRBS. Moreover, the values per iteration of the real-world application are marked with a scenario number of 100.
Runtime	runtime**Model**Scenario**S**.txt	These files contain the runtimes of the different models. A scenario number 100 is linked to the real-world application. Note that the runtimes are explicitly presented in Table 10.

Table 15: Documentation of files in additional result section