

Interval Type-2 Fuzzy Modelling and Stochastic Search for Real-World Inventory Management

Simon Miller · Mario Gongora
· Jon Garibaldi · Robert John

Received: date / Accepted: date

Abstract Real-world systems present a variety of challenges to the modeller, not least of which is the problem of uncertainty inherent in their operation. In this research, an Interval Type-2 Fuzzy model is applied to a real-world problem, the goal being to discover a suitable optimisation configuration to enable a search for an inventory plan using the model. To this end, a series of Simulated Annealing configurations and the Interval Type-2 Fuzzy model were used to search for appropriate inventory plans for a large-scale real-world problem. A further set of tests were conducted in which the performance of the Interval Type-2 Fuzzy model was compared with a corresponding Type-1 Fuzzy model. In these tests the results were inconclusive, though, as will be discussed there are many ways in which Type-2 Fuzzy Logic can be exploited to demonstrate its advantages over a Type-1 approach. To conclude, in this research we have shown that a combination of Interval Type-2 Fuzzy Logic and Simulated Annealing is a logical choice for inventory management modelling and inventory plan search, and propose that the bene-

fits that a Type-2 model offers, can make it preferable to a corresponding Type-1 system.

Keywords Interval Type-2 Fuzzy Logic · Real-World Inventory Management · Simulated Annealing

1 Introduction

Optimising inventory levels within a supply chain is an area of ongoing interest for supply chain managers. Planning the allocation of resources within a supply chain has been critical to the success of manufacturers, warehouses and retailers for many years. Poorly managed resources result in two main problems:

- (a) *Stock outs*, where a node within the chain is unable to satisfy the demand placed on it.
- (b) *Surplus stock* or carry over, where stock is stored from one period to the next.

The consequence of stock outs is lost sales, and potentially lost customers. Surplus stock results in added cost of storing stock and the possibility of stock losing value as it becomes obsolete. Holding some surplus stock is advantageous however; *safety stock* (stock held as contingency) can be used in the event of an unexpected increase in demand or to cover lost productivity.

Various degrees of uncertainty are present in the different data sources used in supply chain management. This uncertainty is exacerbated in demand forecasts by applying methods of analysis which have varying degrees of inherent uncertainty within themselves. Furthermore, other data that is often used in resource planning such as transportation and other costs, customer satisfaction information, etc. is also uncertain. Therefore, Fuzzy Logic and especially Type-2 Fuzzy Logic (T2FL) are particularly appropriate for this problem. While traditional (or Type-1 (T1)) Fuzzy Logic (T1FL) has successfully been used many times for modelling supply chain operation (e.g., Petrovic et al. (2008) and Aliev et al. (2007)), T2FL has been shown to offer a better representation of uncertainty on a number of problems (e.g., Hagrass (2004) and Karnik and Mendel (1999)) as it is able to retain more information about uncertainty and, unlike T1, represent the linguistic uncertainty and multiple perceptions of real-world terms. Both of these advantages are particularly appropriate to the problem of inventory management. The authors believe that T2FL offers a better method of representing the uncertainty within a supply chain, and that its advantages can be beneficial to supply chain managers. In this research an Interval Type-2 Fuzzy Logic (IT2FL) model is used, as it benefits from some

Simon Miller
Horizon Digital Economy Research / Intelligent Modelling and Analysis, University of Nottingham, Nottingham, UK
Tel.: +44 (0)115 8232556
E-mail: s.miller@nottingham.ac.uk

Mario Gongora · Robert John
Centre for Computational Intelligence, De Montfort University, Leicester, UK
E-mail: m.gongora@dmu.ac.uk, rij@dmu.ac.uk

Jon Garibaldi
Intelligent Modelling and Analysis, University of Nottingham, Nottingham, UK
E-mail: jmg@cs.nott.ac.uk

of the advantages of T2FL, while incurring considerably less computation.

The search spaces involved in inventory management are often very large even for a relatively simple problem. As such, it is often not possible to carry out an exhaustive search to find a resource plan; a more efficient method needs to be selected. To address this, Simulated Annealing (SA) is evaluated for the task of finding appropriate resource plans in this work. Initially, Evolutionary Algorithms (EAs) and hybrid methods were also considered, however, early testing showed that these methods were not able to achieve results comparable to SA, which then this became the focus of the research. This method is able to search a large solution space, without the need to evaluate a high proportion of the possible solutions. One difficulty in using SA is the identification of an appropriate configuration, as it presents a significant configuration space. Here, a series of SA configurations were evaluated for this purpose using the IT2FL model mentioned previously as a cost function.

This paper is structured as follows: Section 2 contains background information on Fuzzy Logic, SA and related work in the problem domain. Section 3 provides an overview of the real-world test scenario that is being tackled and Section 4 gives details of how the chosen method is being employed to this end. Section 5 describes the experiments that have been conducted, along with the results. Finally, Section 6 considers the conclusions that can be drawn from the work, and what form future work might take.

2 Preliminaries

This section will provide an introduction to the techniques that have been used in this research, and a look at related existing work in the literature.

2.1 Fuzzy Logic

Traditional fuzzy logic (Zadeh, 1973), also referred to as ‘*Type-1 fuzzy logic*’, is based on fuzzy set theory (Zadeh, 1965) which extends classical (or *crisp*) set theory. In a crisp set, boundaries are sharp, an element either belongs to a set or it does not. With fuzzy sets an element may belong to one or more sets to a varying degree. In Mendel (2001) a fuzzy set F is described as:

“...a generalization of a crisp set. It is defined on a universe of discourse X and is characterized by a membership function $\mu_F(x)$ that takes on values in the interval $[0, 1]$. A membership function

provides a *measure of the degree of similarity* of an element in X to the fuzzy set.”

A common example when considering the difference between crisp sets and fuzzy sets is that of classifying people by height (e.g., Klir et al. (1997)). With crisp sets we might say that a person over 190cm is *tall*, and anyone under this height is *short*. Figure 1a shows how these crisp sets might look. If a person is 189cm, this classical set definition would classify them as *short* as they are under 190cm, a human however would probably still call them *tall* as they would be almost indistinguishable from someone who is 190cm tall. Clearly, these two crisp sets are not adequate for describing whether a person is tall or not, as the boundary creates a situation where a very small difference in height can result in a completely different classification. Fuzzy sets allow us to represent these uncertain boundaries by using varying degrees of membership, usually in the range $[0, 1]$. In this case, for example, a person who is over 190cm could belong to the set *tall* to the degree 1 and the set *short* to the degree 0. Then, if a person who is 189cm is to be classified, they might be *tall* to the degree 0.99 and *short* to the degree 0.01 (the membership grades do not have to add up to 1). As we move down the height scale, individuals would have a lower degree of membership to the set *tall* and a greater degree of membership to the set *short*. Figure 1b shows how the fuzzy sets *short* and *tall* might look.

2.1.1 Type-2 Fuzzy Logic

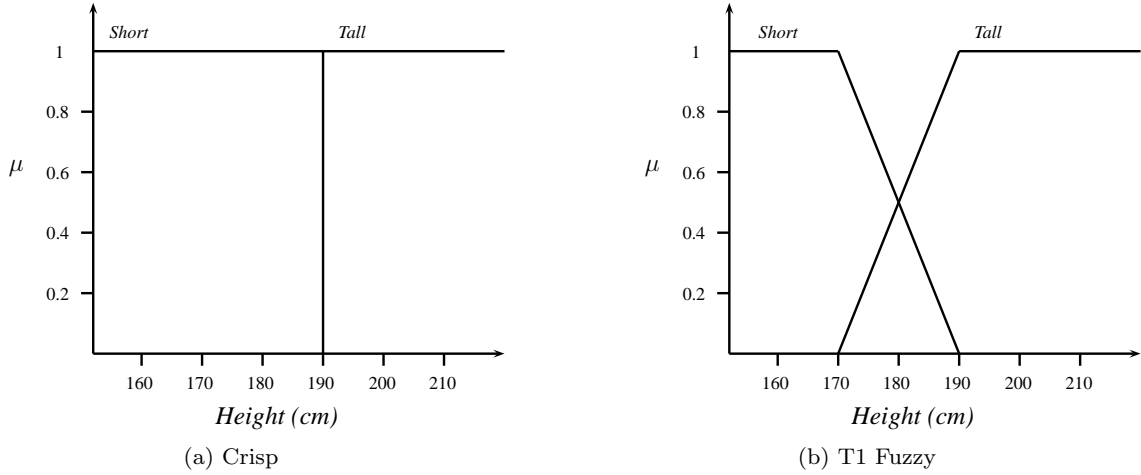
Later in this section (see Section 2.3) it will be shown that previous work using fuzzy logic to model inventory planning problems has been limited to T1FL (e.g., Petrovic et al. (2008), Aliev et al. (2007) and Wang and Shu (2005)). However T1 fuzzy sets represent the fuzziness of the particular problem using a ‘non-fuzzy’ (or crisp) representation - a number in $[0, 1]$.

As Klir and Folger (1988) point out:

“...it may seem problematical, if not paradoxical, that a representation of fuzziness is made using membership grades that are themselves precise real numbers.”

This paradox leads us to consider the role of Type-2 (T2) fuzzy sets as an alternative to the T1 paradigm. T2 fuzzy sets (Mendel and John, 2002) represent membership grades not as numbers in $[0, 1]$, but as T1 fuzzy sets.

Mendel and John (2002) state in their definition:

Fig. 1: Crisp and T1 fuzzy sets *tall* and *short*

“A *type-2 fuzzy set*, denoted \tilde{A} , is characterized by a *type-2 membership function* $\mu_{\tilde{A}}(x, u)$ where $x \in J_x \subseteq [0, 1]$.”

and later added (Mendel et al., 2006):

“When all $\mu_{\tilde{A}}(x, u) = 1$ then \tilde{A} is an interval T2 FS.”

This definition provides an extra dimension over that seen in a T1 fuzzy set, making a T2 fuzzy set 3D. In a *Generalised T2 fuzzy set* (GT2), the secondary membership functions are T1 fuzzy sets. The area in the primary domain covered by a T2 fuzzy set is called its ‘*footprint of uncertainty*’ (FOU), this is the union of all primary memberships. In an Interval Type-2 (IT2) fuzzy set, the FOU is representative of the whole set, as all secondary memberships are equal to 1. Figures 2a and 2b show how the IT2 and GT2 fuzzy sets *short* and *tall* might look.

The authors go on to list four sources of uncertainty that are present in the real-world, that are not represented by T1 fuzzy systems (Mendel and John, 2002):

1. The meanings of words that are used in the antecedents and consequents of rules can be uncertain (words mean different things to different people).
2. Consequents may have a histogram of values associated with them, especially when knowledge is extracted from a group of experts who do not all agree.
3. Measurements that activate a T2FL system may be noisy and therefore uncertain.
4. The data that are used to tune the parameters of a T1FL system may also be noisy

T2 fuzzy sets address these problems by offering added degrees of freedom that may provide a more complete model of uncertainty.

In John and Coupland (2007) the authors state that:

“The more imprecise or vague the data, then type-2 fuzzy sets should offer a significant improvement on type-1 fuzzy sets.”

They then suggest that while T1FL can be used to represent the linguistic uncertainty associated with words, the perception of words (i.e., what words mean to different people) can be better represented with T2FL.

T2 fuzzy sets have been widely used in a number of applications (see John and Coupland (2007) and Mendel (2007) for examples), and on a number of problems T2FL has been shown to outperform T1FL (e.g., Hagrass (2004) and Karnik and Mendel (1999)). T2FL is able to represent uncertainty better by retaining more information and, unlike T1, is able to represent linguistic uncertainty and varying perceptions of real-world terms. Both of these advantages have applications in the type of model presented here. Supply chain operation contains a great deal of uncertainty which can benefit from the extra dimension of fuzziness offered by T2FL. Also, the use of words can provide a more natural interface for supply chain managers allowing them to describe numerical information linguistically.

2.2 Optimisation

A wide variety of optimisation techniques have been used to search for solutions to operations management problems. This section will provide an overview of SA as this is the method we have used in this study. The goal in this work is to find an inventory plan, **not** to optimise the model itself. Some work has been done regarding the use of optimisation methods to design

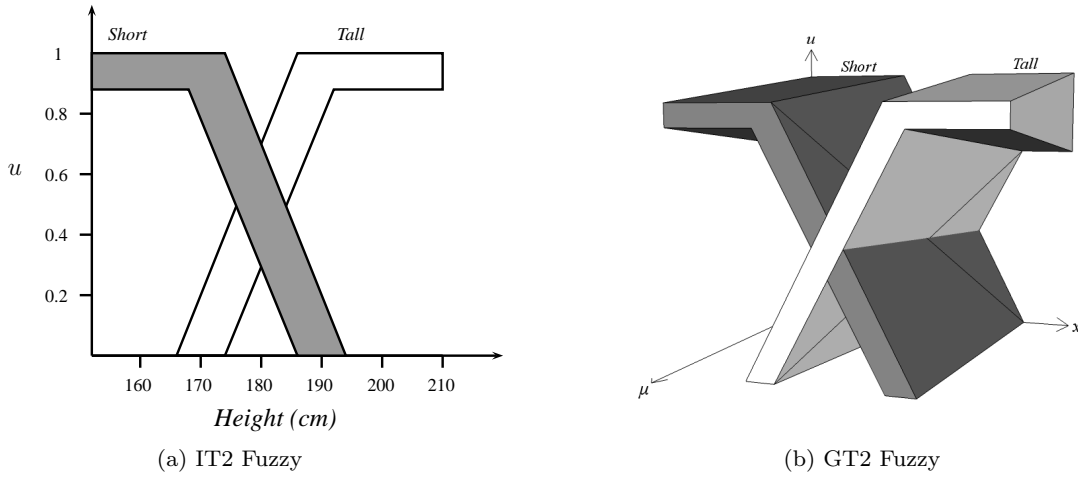


Fig. 2: Interval Type 2 and Generalised T2 sets *tall* and *short*

T2 Fuzzy sets (e.g., Wagner and Hagrass (2007), Sanz et al. (2010), Sanz et al. (2011) and Biglarbegian et al. (2011)) however, optimising the sets themselves is not appropriate for this problem, as there is no ground truth and therefore no basis for comparing one set of fuzzy sets with another. Also, the sets are constantly being created and deleted, there is not a fixed set of fuzzy sets used in the model. In this work we do not optimise the sets; we use an IT2 fuzzy model as a cost function for evaluating potential inventory plans.

2.2.1 Simulated Annealing

SA (Kirkpatrick et al., 1983) is inspired by a real-world phenomenon, in this case the heating and cooling (annealing) of metals to reduce defects. An initial solution is created, then a neighbouring solution is selected and compared with it. The probability of the algorithm accepting the neighbour as the current solution is based upon a parameter value ‘temp’ (representing the temperature) and the difference in quality between the two solutions. The higher the value of ‘temp’, and the closer it is to the current solution, the more likely it is that the algorithm will accept an inferior solution (that is, one with a lower fitness value). The process is then repeated using the selected solution as a starting point. Over the course of a run ‘temp’ is gradually reduced making it less likely that an inferior solution will be accepted. SA has been successfully applied to a number of optimisation problems. For example, in Garibaldi and Ifeakor (1999) SA was used to tune a fuzzy Umbilical Acid-Base assessment model for determining information about a newborn infant’s health, and Tang (2004) conducted a series of experiments that show SA to be a good choice for optimising a production/inventory system. Melouk

et al. (2004) and Bouleimen and Lecocq (2003) use SA to solve production scheduling problems.

The SA algorithm used in this research is detailed in Section 4.3

2.3 Resource Optimisation with Fuzzy Models

A number of research projects have explored the use of optimisation techniques and T1FL models, to search for good solutions to manufacturing and inventory problems. The nature of these problems means that there are no direct methods of calculating an appropriate solution, and the solution spaces are so large that a method that evaluates all possible solutions is prohibitive. To perform a search for ideal solutions in this situation, we can look to the stochastic search methods considered for this paper: EAs and SA. In this section we will look at how EAs and SA have been used in existing research work.

EAs are a popular method of optimising fuzzy resource models. In Aliev et al. (2007) a T1 fuzzy system was used with a GA to model a supply chain. The GA searches for a configuration that maximises profit, while meeting a target service level (percentage of customer orders satisfied) specified by the user; fuzzy sets were used to describe costs, returns, production capacities, storage capacities and forecasts. The proposed fuzzy method, and a crisp method were compared. The crisp system was unable to produce a feasible configuration if the actual demand was lower than the forecast demand. In contrast, the fuzzy model presented was robust and able to cope with fluctuation in demand and production capacity with little impact on profitability.

A similar approach was presented by Wang and Shu (2005). T1FL was used to represent customer demand,

processing time and delivery reliability; a GA found order-up-to levels which were used to decide how much stock should be ordered (what level to ‘order-up-to’) in the next time period. The system attempted to find the configuration that incurred the minimum cost. An optimism-pessimism index was set by the user and passed to the system. When optimistic, the model assumed the best case scenario for material response time. A pessimistic attitude presumed the worst. The results showed that more pessimistic strategies increased the service level, reducing the sales lost through stock outs, and incurring higher inventory holding cost as more stock was kept. More optimistic strategies resulted in a drop in service level and an increase in sales lost, though the cost of holding inventory was also reduced.

Fayad and Petrovic (2005) carried out work using a T1FL job scheduling model with a GA to find near-optimal solutions. In this case, a real world example (a printing company) was used to demonstrate the model. Fuzzy sets were used to describe due dates, processing times and the level of customer satisfaction. The experiments described show that the GA was able to find good solutions.

Earlier work by Sakawa and Mori (1999) also described a method of scheduling jobs that have a T1 fuzzy due date and processing time. A GA was used to find schedules. The GA took into account the similarity of the solutions in a given population. When the initial individuals were created they had a similarity of 0.8 or less to ensure diversity. It was shown that ensuring diversity in this way led to the GA finding an optimal solution on more occasions than a GA without a similarity measure. In testing, the GA was shown to work well, finding solutions with a large correlation between the processing time and the due date. For comparison, a SA process was used, the GA was shown to consistently outperform the SA method. Not only were the solutions better, but the results were more stable. The variance in the best solutions found with SA from test to test was much larger than that of the GA. The computation time for the GA was also shown to be faster.

3 Problem Scenario

In this section the problem that was tackled is described. The data set was provided by one of the industrial partners, Unipart Expert Practices Ltd.

The supply chain consists of three tiers, in the first is a manufacturing distribution centre based in Newquay, and the second contains distribution centres in Stockport (England), Dublin (Ireland) and Orléans (France). The final tier contains customers spread over the UK, Ireland, France and Spain.

Early testing with the model showed that it was not possible to use the whole test scenario for evaluation purposes because of the size of the problem. For these experiments, a subset of the whole problem containing 100 customers and 100 products was chosen, this reduced problem is still considerably larger than any of the fuzzy inventory models seen in the literature (see Table 2). The customers were evenly distributed among the three distribution centres in the second tier, and the products were chosen to represent those stocked at all three distribution centres. An overview of the test scenario used can be seen in Table 1.

Table 1: Real-world scenario test supply chain setup

Tiers	3
Nodes	1,2,100
Products	100
Periods	13
Batch Cost	100
Distance	100
Stock out	25
Multiplier	
Holding Cost	10% purch. price
Production Cost	1.2
Purchase Cost	1.2
Transport Cost	0.1

3.1 Cost values

The aim of the research is to find an inventory plan for this scenario using the IT2 model and SA. In order to calculate the cost of a potential plan, the following values are used.

- The cost of setting up an order is called the *batch cost*. This represents the cost of administration, setting up any machines that are required, and picking the items for dispatch. There is a flat fee for each batch (that can vary by node) which is charged once at each warehouse for the production of a particular item for a particular customer.
- Products have a *production cost*. The total production cost for each batch is calculated by multiplying the number of items by the production cost.
- Transport cost* represents the cost of carrying an item between nodes and customers. The product of the transport cost and distance gives the overall transport cost for a batch of product.
- A *holding cost* is charged if a product is kept at a node for more than one period. The cost is calcu-

lated by taking a specified percentage of the purchase price of the goods held, for items carried over from one period to the next. The purpose of the charge is to represent the cost of storing items, the depreciating value of stock and the losses incurred by tying up capital in unsold stock.

- (e) A *stock out cost* is charged for the shortfall of a product in a particular period. In this model we make the assumption that the end customer is always provided with an item. If it is not in the warehouse, it is obtained at purchase price from a competitor. The stock out cost is the sum of the value of items that had to be purchased. To discourage discovery of solutions that have a shortfall of stock, a stock out penalty is added by taking the product of the stock out cost and a multiplier provided by the user. Stock out penalty is applied to all but the final echelon, which supplies the end customer. In the final echelon, service level is used to determine how good a solution is. To administer a stock out penalty as well would be to penalise a solution twice for the same shortfall, leading to the SA algorithm being pressured to find solutions that satisfy 100% of customer demand, regardless of the service level required by the user.
- (f) To discourage solutions that do not meet service level requirements, a *service penalty* is added to the cost of poor solutions in proportion to a solution's distance from a specified target service level. Service level is calculated by taking the percentage of customer demand that is completely satisfied. To measure satisfaction the fuzzy sets for the current stock level and the forecast are compared. An agreement index is calculated by looking at where the sets intersect, or is set to 1 if the inventory level exceeds the forecast. As stated before, stock out penalties are not applied here. It may appear that a more satisfactory solution would be to simply measure service level throughout the chain and not use stock out penalties. However in practice applying service penalty throughout the model results in SA finding solutions in which the nodes within the chain place little or no orders on each other, enabling solutions to achieve a good service level without satisfying a significant amount of customer demand. Stock outs need to be charged within the chain however, else SA finds solutions in which only the final echelon before the customer supplies any product.

3.2 Additional data

It became clear early on in the research that some values were missing from the data set that had been provided,

and the missing data proved to be unavailable for our experiments. To remedy this, a number of methods were agreed with the industrial partner to generate the data, while maintaining the integrity of the data. The data did not state which distribution centre supplied each customer; as stated previously, customers were served by one fixed distribution centre. To deduce the supply relationships, the products being demanded by each customer were examined. The distribution centre that supplied the most of these was then assigned to the customer. Products that appear in the forecast but were not listed as supplied by any distribution centre were allocated to the distribution centre of the customer demanding the product and given a lead time, minimum order quantity and unit of order that were an average of all products listed.

For the Newquay and France distribution centres, there was some information missing regarding lead times; after discussing this with our industrial partner, the missing information was taken from the Stockport distribution centre. Capacities, minimum order quantities and order quantities were not explicitly given in the test data; to derive these, the forecast was examined, and the smallest order (capped at 1,000) for a product at a node was used to represent both the minimum order quantity and order quantity. For Newquay, the minimum orders and units of order were calculated by multiplying the Stockport numbers by 35, which is in line with the demand that it could reasonably expect. The capacities were set to 20 times the minimum order quantity. A flat batch cost of £100 was applied across all distribution centres.

3.3 Comparison

Table 2 shows various aspects of the problem being tackled in this work, the proposed solution, and those of other projects described in the literature. Though these are all different problems, with different models (and therefore not directly comparable), the table does illustrate the difference in size of the problem being tackled in our research.

4 Proposed System

This section will focus on the model that has been created and the implementation of the optimisation methods that are used.

Table 2: Comparison of problems and models

	Petrovic (2008)	Aliev (2007)	Wang (2005)	Proposed System
Tiers	2	3	7	3
Nodes	1	4	15	4
Customers	3	2	1	100
Products	1	3	6	100
Periods	-	2	26	13
Fuzzy Logic	T1	T1	T1	IT2
Optimisation	Iterative	GA	GA	SA

4.1 Model

The proposed model represents the interaction of nodes within a multi-tier supply chain. Figure 3 provides an example of a typical supply chain. In each tier (or *echelon*) there are one or more nodes that supply the subsequent tier with one or more products, and receive stock from the preceding tier. The first tier receives goods from an external supplier which for the purpose of these experiments is assumed to have infinite capacity, the final tier supplies the customer. Below the first tier, capacity is limited by node and product. If the external supplier were not able to supply 100% of the first tier's requirements, the external supplier would need to become part of the model, allowing us to represent the limitations of their supply.

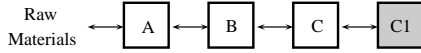


Fig. 3: A typical supply chain

Customer demand is provided by a forecast which is given to the model at run-time. This forecast represents the demand placed upon the final tier in the supply chain. Tiers above this can see their own demand by looking at the suggested inventory levels in the succeeding tier, as they will be required to supply these items.

In order to use the model, the following information must be provided:

- Number of echelons (not including the end customer)
- Number of nodes in each echelon
- Number of end customers
- Number of products
- Number of periods
- Service level required (as a percentage of orders filled completely)

- Capacities for each product at each node (amount that can be produced in one period)
- Lead time (in periods) for production/supply of each product at each node
- Minimum order and unit of order quantities for each product at each node
- Initial stock levels for each node
- Distance between nodes in successive echelons
- Forecast of customer demand
- Suggested inventory levels
- Costs including:
 - Batch cost
 - Production cost
 - Transport cost
 - Holding cost (as a percentage of purchase price)
 - Purchase price

This information is provided using a series of .csv files, which are loaded into the model before it is executed. As will be shown in Section 4.1.1 some of these values are represented using IT2 fuzzy numbers to account for the uncertainty they present. Using this information the model will calculate the cost of the given resource plan as shown in Equation 1 where $F(x)$ is the fitness of the solution x and:

- T = total number of periods to be evaluated
- I = total number of inventory locations
- b = batch cost
- p = production cost
- tr = transport cost
- h = holding cost
- s = stockout cost
- sr = service level
- tsr = target service level

$$F(x) = \left(\sum_{t=0}^T \sum_{i=0}^I (b_{i,t} + p_{i,t} + tr_{i,t} + h_{i,t} + s_{i,t}) \right) + \max((tsr - sr) 1000000, 0) \quad (1)$$

The resource plan is provided by the SA algorithm, and takes the form shown in Section 4.3.

4.1.1 Interval Type-2 Fuzzy Numbers

In this model, IT2 fuzzy numbers (represented using IT2 fuzzy sets) are used to represent some values, and fuzzy arithmetic is used to calculate costs using a method seen in Kaufmann and Gupta (1985) and Hamrawi and Coupland (2009). Fuzzy sets are represented using a series of α -cuts, each set is an array of pairs of intervals. Each pair shows the area of values in x covered at a particular value of μ , the first interval is

the left hand side of the set, and the second the right. Operations on the IT2 fuzzy sets are performed at the interval level, corresponding intervals (at the same μ) are taken from two sets, the operation performed and the result stored in a third fuzzy set. Four interval level operations were required to compute the fitness of solutions as shown in Equation 1; sum, subtract, multiply and maximum intersection.

Algorithms 1, 2, and 3 describe the addition, subtraction and product processes respectively where:

- x = first IT2 fuzzy number
- y = second IT2 fuzzy number
- z = resulting IT2 fuzzy number
- $x_i = i^{th}$ pair of intervals in x
- $y_i = i^{th}$ pair of intervals in y
- $z_i = i^{th}$ pair of intervals in z
- at = temporary interval a
- bt = temporary interval b

for each i in I :

$$z_i = [x_i^{as} + y_i^{as}, x_i^{ae} + y_i^{ae}, x_i^{bs} + y_i^{bs}, x_i^{be} + y_i^{be}]$$

Algorithm 1: IT2 fuzzy addition

for each i in I :

$$z_i = [x_i^{as} - y_i^{ae}, x_i^{ae} - y_i^{as}, x_i^{bs} - y_i^{be}, x_i^{be} - y_i^{bs}]$$

Algorithm 2: IT2 fuzzy subtraction

for each i in I :

$$at = [a_i^{as} \times b_i^{as}, a_i^{as} \times b_i^{ae}, a_i^{ae} \times b_i^{as}, a_i^{ae} \times b_i^{ae}]$$

$$z_i^{as} = \min(at)$$

$$z_i^{ae} = \max(at)$$

$$bt = [a_i^{bs} \times b_i^{bs}, a_i^{bs} \times b_i^{be}, a_i^{be} \times b_i^{bs}, a_i^{be} \times b_i^{be}]$$

$$z_i^{bs} = \min(bt)$$

$$z_i^{be} = \max(bt)$$

Algorithm 3: IT2 fuzzy product

To compute service level, an agreement index between the forecast and the current inventory level was calculated using the maximum intersection of the sets

representing inventory and forecast. The agreement index was then used to calculate service level. The pseudocode shown in Algorithm 4 describes the process in detail.

$$h = 1/I$$

$$IF a_{I-1}^{bs} \geq b_{I-1}^{as} THEN result = 1, EXIT$$

ELSE

for each i in I :

$$IF (a_i^{bs} > b_i^{as} AND a_i^{bs} < b_i^{ae})$$

$$OR (a_i^{bs} > b_i^{ae} AND a_i^{bs} < b_{i+1}^{as})$$

$$THEN result = (i + 1) \times h, EXIT$$

Algorithm 4: IT2 fuzzy maximum intersect

Figure 4 shows how the set ‘about 200’ may look with the α -cut representation used, where x is the scale of values being represented. Expressing variables in this way offers supply chain managers the means to state the uncertainty associated with a particular variable, and their confidence in their appraisal, this is not possible when using a T1 system without discarding some of the information.

In order to produce an output that can be applied to a real-world supply chain, some of the IT2 fuzzy numbers need to be defuzzified. Defuzzification is the process of taking a fuzzy set and deriving a crisp value from it. To do this, the Karnik-Mendel method (Karnik and Mendel, 2001) is used. This is a widely used method that finds an interval representing the centroid of an IT2 fuzzy set. The interval can then be used to obtain a crisp number by finding its centre.

In the literature, other than the author’s previous work (e.g., (Miller and John, 2010)), fuzzy systems that address inventory management problems have been restricted to T1FL systems that represent relatively small supply chain networks. This work proposes that such an approach limits the value of a model, and that T2FL may provide a better solution. IT2FL has been used because it is computationally cheaper than general T2FL as it restricts the additional dimension, referred to as the secondary membership function, to only take the values 0 or 1. We believe that the extra degree of freedom offered over a T1FL model will allow a better representation of the uncertain and vague nature of data used in supply chain management.

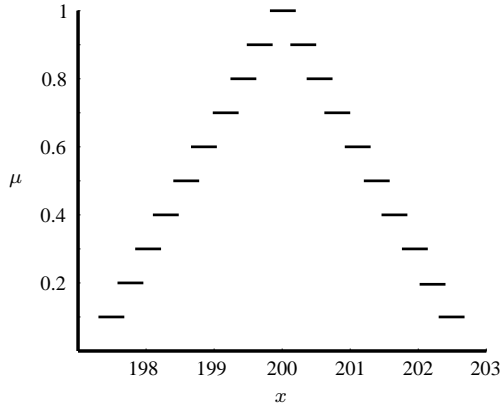


Fig. 4: Interval representation of IT2 fuzzy set ‘about 200’

All of the data in the real-world scenario is crisp, to reflect the uncertainty in the data some values were fuzzified. The following values were represented using IT2 fuzzy numbers.

- Forecast demand
- Inventory level
- Transportation distance
- Transportation cost
- Stock out level
- Stock out cost
- Carry over
- Holding cost

For each of these values we can use the linguistic term ‘about n ’, e.g., forecast demand of product a for customer b in period c may be ‘about 200’.

The forecast was fuzzified using a primary and secondary fuzzification factor, a primary fuzzification factor of 10% of the value (representing uncertainty) and a secondary fuzzification factor of 2% of the value (representing confidence) were chosen for the forecast. For these experiments the values chosen are somewhat arbitrary, though the primary fuzzification value should be larger than the secondary fuzzification value. The primary fuzzification factor creates the principle membership function (which can be thought of as a T1 set that the IT2 set is based around), and the secondary fuzzification factor is used to produce the FOU. Larger or smaller values can be chosen to reflect an expert’s opinion of the margin of error of a forecast and their confidence in their opinion.

4.2 Inventory Plan Optimisation

The purpose of these experiments is to evaluate the performance of a set of optimisation configurations.

Initially, EAs, SA and a hybrid method were chosen to conduct the search for inventory plans as these have been shown to work well in previous experiments using earlier versions of the model, and simpler problems than the one used in this research. In Miller et al. (2009) experiments were performed with 5 methods: EAs, SA, the Great Deluge Algorithm (GDA) and hybrids of an EA and SA and an EA and the GDA. The results of the experiments showed that SA was able to find the best inventory plans, but that by combining an EA with SA, comparable costs could be achieved in less than half the time. Subsequent work in Miller et al. (2010) showed how varying the configuration of an EA with variable mutation and uniform crossover could improve results gained using the method. In this paper, a large-scale real-world problem is modelled, presenting additional problems (e.g., scaling issues) during optimisation. This work focuses on discovery of an ideal optimisation method and configuration for use with such a problem.

In preliminary experiments however, it quickly became apparent that the EA and hybrid algorithm (a combination of EA and SA) were performing significantly worse than SA. For this reason, we chose to focus our work on the SA algorithm.

4.3 Simulated Annealing Implementation

In this implementation, SA consists of the following components:

- (a) The *representation* used to describe potential resource plans is a three dimensional matrix of inventory levels. The following pseudocode describes the array.

```
float PLAN[Periods][Nodes][Products]
```

Each element of the matrix contains a floating point value (between 0 and 1) representing the number of items held in a time period, by a source node of a particular product. For example if $PLAN[1][2][25]$ held a value of 0.5, this would indicate that in period 1, node 2 is holding 50% of its capacity of product 25. Using the capacity in this way limits the values that can be created, it is not possible to overload a node. In addition to this values are restricted by minimum order quantities, and order quantities. For example, there may be a minimum order quantity of 500 for a particular product, and then orders restricted to 100s above that (e.g. 500, 600, 700, 800 etc.).

- (b) The *initial solution* is randomly generated, containing only valid inventory levels between the minimum order quantity and the capacity, in units that comply with valid order quantities.

- (c) *Fitness* is evaluated using the IT2FL model described. A solution's fitness is judged by the cost and service level achieved, Equation 1 describes how this is computed. For those solutions that meet the service level requirement specified by the user, the fitness is the cost of the solution; for those that do not a penalty is added to the cost as discussed in Section 4.1. This encourages SA to find low cost solutions, that meet a target service level.
- (d) *Selection* of a new solution is based on its fitness and the current temperature. If the perturbed solution is better than the current solution it is selected, if it is not, then its distance from the current solution and the temperature are used to produce a probability of selection. The closer it is to the current solution, and the higher the temperature, the more likely it is to be selected.
- (e) *Perturbation* takes a selected solution and then replaces a variable amount of elements with other valid values to create a child. Elements are selected for replacement at random, the number of changes is decided using a fitness ranking proportionate method similar to *roulette wheel selection*, the difference being that the chance of selection is *rank proportionate* rather than *fitness proportionate*. For example, in a population of 100, the individual ranked 1st is twice as likely to be chosen as the individual ranked 50th. In this case, it means that it is more likely that a smaller number of perturbations will be chosen, though it is possible that a large number will be made.
- (f) *Temperature* is reduced after a given number of iterations with improvements have passed, or if a number of iterations without improvements (usually less) have passed. The reduction is made by subtracting the decrement value from the current temperature.
- (g) *Termination* of the algorithm occurs when the temperature reaches zero.

5 Experimental Methods

Three stages of experiments have been conducted, these are:

1. **Identification of a suitable configuration** (Section 5.1) - The size of the problem prohibits the running of in-depth tests on a range of configurations. This initial set of experiments aims to discover a configuration that can be used in further, more extensive tests, in a reasonable time frame.
2. **Performance evaluation** (Section 5.2) - Once we have a suitable configuration, we will perform a

more extensive set of sets to evaluate the performance of SA and our IT2FL model on the real-world problem that has been selected.

3. **Comparison with T1 system** (Section 5.3) - The final set of tests will use the SA configuration found in the first stage to search for inventory plans using a corresponding T1 model. The purpose of this stage is to identify differences in performance between T1 and IT2, and decide whether one model can be said to be better than the other for this problem.

5.1 Identification of a suitable configuration

The first stage of testing aimed to identify an appropriate area of the configuration space that was suitable for further experimentation. It was shown in Miller et al. (2009) that SA setup is entirely dependent upon the problem domain, so the temperature and the rate of decrement were varied. In all tests 100 iterations were allowed to pass with improvements and 50 iterations without improvement before the temperature was reduced by the decrement value. Table 3 shows the base configuration used.

Table 3: Base configurations for SA

Its. with improvement	Its. no improvement
100	50

In this first set of tests, configurations of SA were explored; the configurations can be seen in Table 4 along with the results. For these exploratory tests, time was a factor; only a single test was executed for each configuration. Early tests showed that runs took a very long time to complete with the test scenario. It was also with time in mind that the configurations have been chosen, far fewer individuals were evaluated than in previous work in order to allow these preliminary tests to be completed within a reasonable time frame. Configurations were chosen to result in a comparable number of evaluations in each, to avoid a situation where more and more iterations were performed. When a suitable configuration was discovered for this problem, longer tests were executed to take advantage of multiple runs.

5.1.1 Results

The best results came from in test 3 when a starting temperature of 25,000 was used, Table 5 shows the attributes of the plan found.

Table 4: Results of first stage of testing

No.	Temp.	Dec.	Cost
1	100,000	1,000	£608,914,048
2	50,000	500	£601,184,960
3	25,000	250	£598,187,840
4	12,500	125	£602,834,688
5	125,000	1250	£606,111,104
6	150,000	1500	£610,011,648
7	10,000	100	£619,027,008
8	15,000	150	£599,733,632
9	20,000	200	£612,790,464
10	30,000	300	£606,846,080
11	40,000	400	£604,292,352
12	60,000	600	£604,731,008
13	70,000	700	£600,298,880
14	80,000	800	£608,938,368
15	90,000	900	£619,253,760
16	21,000	210	£607,766,656
17	22,000	220	£606,997,952
18	23,000	230	£600,488,576
19	24,000	240	£601,860,608
20	26,000	260	£600,713,984
21	27,000	270	£606,076,096
22	28,000	280	£618,071,104
23	29,000	290	£609,204,032

It was promising that less than 1% of the total cost was incurred through stockouts. As should be expected, the majority of the cost was production/procurement costs. Holding cost, however, was a concern, in this plan as 29.4% of the total cost came from holding stock in warehouses. This suggests that too much stock was being allocated to the warehouses. As stated before, this can result in products losing value and means that capital is unnecessarily tied up in stock. For seasonal products, it can mean that stock is being held while it is out of season.

Table 5: Test 3 - Resource plan attributes

Total Cost	Service Level	Batch Cost
£598,187,840	99.9%	£9,067,901 (1.5%)
Production Cost	Transport Cost	
£407,605,024 (68.1%)	£899,522.19 (0.2%)	
Stockout Cost	Holding Cost	
£4,886,310.50 (0.8%)	£175,729,056 (29.4%)	

In all subsequent tests in this stage, the result produced using the configuration in test 3 was never bettered. This is the configuration that was used in the next stage of testing for a more extensive look at performance.

5.2 Performance Evaluation

With the best configuration found in the first stage, extra tests were completed. Table 6 shows the setup used, where *stalls* refers to the number of iterations that were allowed to pass without improvement. In all, 30 tests were run with differing random seeds. The purpose of these tests was to give a more complete picture of the performance of the model and optimisation with the settings found, enabling a comparison with a corresponding T1FL system in the next stage of testing. Table 7 provides a summary of the results.

5.2.1 Results

The results showed that there was a spread of £40,183,616 between the most expensive and cheapest solutions found. In real terms this represents a significant amount, though, relative to the costs themselves the best is just 6.4% cheaper than the most expensive; along with the standard deviation, this suggests that that SA is finding relatively stable solutions. In the next stage of testing, a corresponding T1 fuzzy system was compared with the IT2 system to see if there was any difference in the results, and whether one system could be said to be better than the other.

Table 6: Optimisation setup for second stage of testing

Temp.	Dec.	Iterations	Stalls
25,000	250	100	50

Table 7: Summary of results of second stage of testing

Min. Cost	Max. Cost	Mean Cost
£586,283,904	£626,467,520	£605,181,126.4
Std. Dev.		
£10,110,866.88		

5.3 Comparison with T1 System

In this stage a T1 system was created that represented how the IT2 system would be, if it were implemented using T1FL. To create a T1 system, the intervals in the IT2 system were reduced to zero width eliminating their FOU's. Both IT2 and T1 tests use the same model,

because of this, their execution times are the same as the computation is not reduced for the T1 version of the model. The tests performed in Stage 2 were run again. The purpose of these tests was to see the difference made by using IT2 over T1, and whether it could be demonstrated that one approach worked better than the other. Table 8 provides a summary of the results.

5.3.1 Results

The results appeared to be similar on first inspection. The cost of solutions found by the T1 system were marginally more expensive, and had slightly more deviation. A range of £43,756,032 existed between the cheapest and most expensive solutions found; the best was 7% cheaper than the worst. The attributes of the resource plans found in the best of the T1 and IT2 tests are provided in Table 9, in both cases 99.9% of customer demand was satisfied.

Table 8: Summary of results of third stage of testing

Min. Cost	Max. Cost	Mean Cost
£584,430,912	£628,186,944	£606,121,892.3
Std. Dev.		
£11,316,239.36		

Table 9: Best resource plan attributes found in third stage of testing

Type	Total Cost	Batch Cost
T1	£584,430,912.00	£9,067,501.00 (1.6%)
IT2	£586,283,904.00	£9,067,699.00 (1.5%)
Type	Production Cost	Transport Cost
T1	£396,506,880.00 (67.8%)	£899,482.50 (0.2%)
IT2	£400,246,080.00 (68.3%)	£899,502.25 (0.2%)
Type	Stockout Cost	Holding Cost
T1	£4,874,550.50 (0.8%)	£173,082,480.00 (29.6%)
IT2	£5,190,420.00 (0.9%)	£170,880,224.00 (29.1%)

Analysing the best solutions it is apparent that the solution found using the T1 model saved money by achieving lower production and stockout costs, but incurred higher holding costs. The best solution was found using T1 fuzzy logic, however, on average the IT2 solutions were slightly better, and more stable. We can speculate as to the reason for the difference

in results, for example, a likely explanation seems to be that the extra uncertainty afforded by the IT2 model leads to a smoother cost function, this could potentially result in less variation between results, as the ‘steps’ are smaller. However, the resulting cost of a solution is not necessarily an accurate guide to the quality of a solution. It may be possible to produce a model that results in the discovery of a cheaper solution; however, if these solutions do not produce the same results in the real-world then the quality of the solution is in doubt.

The next step was to examine the results to ascertain whether there were any significant differences between the T1 and IT2 tests. Table 10 shows the absolute value of the mean difference between tests, the standard deviation of differences, the maximum and minimum differences (where the IT2 results are subtracted from the T1 results), and the test statistic and two-tailed p values produced by performing a t-test on the two sets of results with $\alpha=0.05$. The t-test is used to test the hypothesis that there is no difference between the means of two samples of data. A test statistic and a p value are produced that can be used to determine whether the mean difference between two samples of data is statistically significant. Typically (and in this case) a p value of less than 0.05 is taken as evidence of there being a statistically significant difference between the two sets of data.

Table 10: Comparison of results in T1 and IT2 tests

Abs. Mean Diff.	Std. Dev.	Min. Diff.
£940,765.87	14,885,709.55	£-27,818,496.00
Max. Diff.	Test Statistic	Test Critical Two Tail
£31,395,136.00	0.34	2.05
Two Tailed p		
0.74		

The *Test Critical Two Tail* column shows the value that the test statistic must be greater than in order to be statistically significant. In this test it was shown that neither a p value of less than 0.05, nor a test statistic greater than 2.05 (test critical) were produced. From this, it was deduced that there was no appreciable difference between the mean results of the T1 and T2 tests. When considering reasons for this we should note that the IT2 fuzzy sets were created in a symmetrical manner, that is, by taking a T1 set and shifting the boundaries. It may be that in this case the approach taken has resulted in an IT2FL system that is roughly equivalent to the T1FL system it was

compared with, reducing the advantage IT2FL can provide. To tackle this shortcoming, an approach that aims to model the actual uncertainty in the data (which is significantly more complex), rather than applying a uniform uncertainty should be adopted. By doing this, the extra information that the model represents would be based only on actual information provided by the user, providing a more accurate description of uncertainty.

One possible view is that this is a positive result with regard to the fact that the IT2FL model has the advantages that IT2FL brings, while matching the performance of a T1 model. In Section 6, the meaning of these results is considered in more detail, and courses of action for fully exploiting the advantages of T2FL are proposed.

6 Conclusion

A large-scale real-world case study was collected from a UK supply chain design and management consultancy, and a series of experiments was conducted to discover a suitable method and configuration to determine near-optimal resource plans using an IT2 fuzzy supply chain model. The tests showed that using the model, SA was able to find realistic solutions to the real-world scenario, satisfying 99.9% of customer demand. As the algorithm was guided purely by the model, this suggests that the IT2FL model is a valid representation of the problem. If this were not the case, they may have been guided towards solutions that were unfeasible. However, all of the tests gave results with high holding costs, showing that there are some limitations that could be considered in additional research. The size of the problem tackled posed problems for the algorithm used, this led to the algorithm being restricted in order to allow them to run in a reasonable time, on the hardware available, limiting performance. Future work could look at how resource plans could be encoded to reduce the resources required to process them.

In an extra set of tests comparing the IT2 system with a corresponding T1 system it was demonstrated that there was no statistically significant difference in the results produced. On reflection, perhaps this is not surprising. Uncertainty was applied in a symmetrical, uniform manner; in reality, the uncertainties involved are much more varied and complex. To address this IT2FL (or T2FL) should be used in a way that takes advantage of its strengths. Using an IT2 approach to this problem allowed the user to specify both the uncertainty of supply chain variables, and their confidence in their assessment. This was then represented without compromising the information gathered, unlike a T1

approach. Further to this, a generalised T2 methodology could allow the representation of the uncertainty and confidence levels associated with a group of experts opinions (as seen in Wagner and Hagnas (2010)) while preserving all of the information gathered including where there was agreement, something not possible in an IT2 approach. Although some work has been done here, further exploiting these advantages could form an important part of future work with the model. These improvements would thoroughly demonstrate the advantage of the method chosen over using T1FL, as correctly modelling words and maintaining all information from a group of experts, can only be achieved at this level using T2FL. This makes T2FL a logical choice when modelling systems that contain a great deal of uncertainty, as we can produce a more complete model of the uncertainty present.

Acknowledgements The research reported here has been funded by the Technology Strategy Board (Grant No. H0254E).

References

- R.A. Aliev, B. Fazlollahi, B.G. Guirimov, and R. R. Aliev. Fuzzy-genetic approach to aggregate production-distribution planning in supply chain management. *Information Sciences*, 177:4241–4255, 2007.
- M. Biglarbegan, W. Melek, and J. Mendel. On the robustness of type-1 and interval type-2 fuzzy logic systems in modeling. *Information Sciences: an International Journal*, 181(7):1325–1347, 2011.
- K. Bouleimen and H. Lecocq. A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *European Journal of Operational Research*, 149(2):268–281, 2003.
- C. Fayad and S. Petrovic. A fuzzy genetic algorithm for real-world job shop scheduling. In *Proceedings of the 18th international conference on Innovations in Applied Artificial Intelligence*, pages 524–533, Bari, Italy, 2005.
- J.M. Garibaldi and E.C. Ifeachor. Application of simulated annealing fuzzy model tuning to umbilical cord acid-base interpretation. *Fuzzy Systems, IEEE Transactions on*, 7(1):72–84, 1999. ISSN 1063-6706.
- H. Hagnas. A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. *IEEE Transactions on Fuzzy Systems*, 12(4):524–539, August 2004.
- H. Hamrawi and S. Coupland. Type-2 fuzzy arithmetic using Alpha-planes. In *Proceedings of the Joint 2009 International Fuzzy Systems Association World*

- Congress and 2009 European Society of Fuzzy Logic and Technology conference*, pages 606–611, 2009.
- R. John and S. Coupland. Type-2 fuzzy logic a historical view. *IEEE Computational Intelligence Magazine*, 2(1):57–62, February 2007.
- N. Karnik and J. Mendel. Applications of type-2 fuzzy logic systems to forecasting of time-series. *Information Sciences*, 120:89–111, 1999.
- N. Karnik and J. Mendel. Centroid of a type-2 fuzzy set. *Information Sciences*, 132:195–220, 2001.
- A. Kaufmann and M.M. Gupta. *Introduction to fuzzy arithmetic: theory and applications*. Van Nostrand Reinhold New York, 1985. ISBN 0442230079.
- S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220 (4598):671–680, May 1983.
- G.J. Klir and T.A. Folger. *Fuzzy Sets, Uncertainty and Information*. Prentice Hall, 1988.
- G.J. Klir, Z. Wang, and D. Harmanec. Constructing fuzzy measures in expert systems. *Fuzzy Sets and Systems*, 92:251–264, 1997.
- S. Melouk, P. Damodaran, and P.Y. Chang. Minimizing makespan for single machine batch processing with non-identical job sizes using simulated annealing. *International Journal of Production Economics*, 87 (2):141–147, 2004.
- J. Mendel. *Uncertain Rule-Based Fuzzy Logic Systems - Introduction and New Directions*. Prentice Hall, Upper Saddle River, NJ 07458. USA., 2001.
- J. Mendel. Advances in type-2 fuzzy sets and systems. *Information Sciences*, 177(1):84–110, January 2007.
- J. Mendel and R. John. Type-2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems*, 10(2):117–127, April 2002.
- J.M. Mendel, R.I. John, and F. Liu. Interval type-2 fuzzy logic systems made simple. *IEEE Transactions on Fuzzy Systems*, 14(6):808–821, December 2006.
- S. Miller and R. John. An interval type-2 fuzzy multiple echelon supply chain model. *Knowledge-Based Systems*, 23(4):363–368, May 2010.
- S. Miller, M. Gongora, and V. Popova. A comparison of methods for optimising resource plans. In *Proceedings of the 9th Annual Workshop on Computational Intelligence (UKCI 09)*, pages 37–42, Nottingham, UK, 7th-9th September 2009.
- S. Miller, M. Gongora, and R. John. Inventory optimisation with an interval type-2 fuzzy model. In *Proceedings of 2010 IEEE World Congress on Computational Intelligence (WCCI 2010)*, pages 2077–2083, Barcelona, Spain., 18th-23rd July 2010. IEEE.
- D. Petrovic, Y. Xie, K. Burnham, and R. Petrovic. Coordinated control of distribution supply chains in the presence of fuzzy customer demand. *European Journal of Operational Research*, 185:146–158, 2008.
- M. Sakawa and T. Mori. An efficient genetic algorithm for job-shop scheduling problems with fuzzy processing time and fuzzy due date. *Computers & Industrial Engineering*, 36:325–341, 1999.
- J. Sanz, A. Fernández, H. Bustince, and F. Herrera. A genetic tuning to improve the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets: Degree of ignorance and lateral position. *International Journal of Approximate Reasoning*, 52 (6):751 – 766, 2011.
- J.A. Sanz, A. Fernández, H. Bustince, and F. Herrera. Improving the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets and genetic amplitude tuning. *Information Sciences*, 180 (19):3674–3685, 2010.
- O. Tang. Simulated annealing in lot sizing problems. *International Journal of Production Economics*, 88 (2):173–181, 2004.
- C. Wagner and H. Hagaras. A genetic algorithm based architecture for evolving type-2 fuzzy logic controllers for real world autonomous mobile robots. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, pages 193–198, London, UK., July 2007.
- C. Wagner and H. Hagaras. An approach for the generation and adaptation of z-slices based general type-2 fuzzy sets from interval type-2 fuzzy sets to model agreement with application to intelligent environments. In *Proceedings of 2010 IEEE World Congress on Computational Intelligence (WCCI 2010)*, pages 650–657, Barcelona, Spain, 18th-23rd July 2010. IEEE.
- J. Wang and Y-F. Shu. Fuzzy decision modelling for supply chain management. *Fuzzy Sets and Systems*, 150(1):107–127, 2005.
- L.A. Zadeh. Fuzzy sets*. *Information and control*, 8 (3):338–353, 1965.
- L.A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-3 (1):28 –44, January 1973.