2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom)
and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)

Manual Task Completion Time Estimation for Job Shop Scheduling Using A Fuzzy Inference System

Longzhi Yang, Jie Li, Phil Hackney
Department of Computer and Information Sciences,
Northumbria University
Newcastle upon Tyne, United Kingdom NE1 8XT
Email: longzhi.yang@northumbria.ac.uk

Fei Chao
Cognitive Science Department
Xiamen University
Xiamen
PR. China

Mark Flanagan
BTL Group Ltd
Shipley, West Yorkshire
BD17 7DB
United Kingdom

***Abstract*—Manual collating and packing is still the most cost-effective way of dispatching goods in many applications, despite of the rapid development of assembly robots. One such application, is the manufacturers of Point of Sale (POS) and Point of Purchase (POP) in the design and print industry; they produce and dispatch display objects in various quantities, shapes and sizes. The display objects, typically posters and 3D displays, are designed for different commercial promotion events in supermarkets, shopping malls and other high street shops. It is difficult to assemble and pack the objects using assembly robots due to the potential complexity and infinite variety of the tasks. The collate and pack department must manually pick, collate, assemble and pack items, often carried out in multiple lines based on the nature of the jobs, as the last stage of the manufacturing process. The jobs themselves are often unique bespoke arrangements defying a generic solution, flat-packed to minimise portage costs. The design of the lines and the schedule of the lines are determined by the area manager based on their expertise and historic knowledge, which seriously limits the effectiveness of the widely available automatic global scheduling system for these POP and POS print manufacturers. This paper proposes a job completion time estimation system which estimates the completion times for different tasks under different conditions such that the intelligent scheduling system can make a schedule globally by artificially treating the assembly lines as virtual machines. The system is implemented using a particular fuzzy inference system, fuzzy interpolation, and an illustrative example demonstrates the working and potential of the proposed solution.**

Keywords*—*job shop scheduling; fuzzy inference system; fuzzy rule interpolation;

I. Introduction

In the bespoke print industry, serving the Point of Sale (POS) and Point of Purchase (POP), a manufacturing job may require tens of operations, utilising several different machines coupled with manual tasks in different operation centres, and many of these jobs may need to be processed in parallel at any single time. However, many small and medium-sized enterprises (SMEs) and large manufacturers still manually implement planning and scheduling with the support of some form of information management systems, often implemented utilising tools, such as an Excel spreadsheet, codifying crude heuristics and when implemented to determine a ball-park guide to estimated manufacturing times. Consequently, the optimal or a sub-optimal scheduling of all such operations for all the current jobs is hard to be achieved. Due to its complexity, manual scheduling in this situation is often practically implemented as a simple first in and first out queue or a queue driven by dispatching deadlines. This not only restricts the efficiency of manufacturing process, but also often leads to unnoticed late jobs until the times close to the dispatch deadlines.

The late jobs may also result from too many jobs that have been bid and accepted by the sales team exceeding the maximum capacity of the manufacture facility. This happens because the sales team is not able to access to the forward view of the available capacity of the manufacturer and thus the quotations are made available to customers simply based on the nature of the job rather than a reflection of demand-supply relationship in the manufacturer. A side effect is also artificially pulling down the effective ‘management planning horizon’ to a period as short as a week, instead of maintaining a strategic month or quarter perspective. Then, costly last minute solutions or even breaches of contract occurs. This common issue exists currently in most manufacturers, which has seriously limited the delivery performance (requiring unnecessary and more expensive same-day delivery services to be invoked) and manufacturing efficiency.

Manufacturing efficiency can be significantly improved by employing an intelligent job shop scheduling (JSS) system. Various JSS solutions have been developed by the community of Operation Research (OR) and Artificial Intelligence (AI), to deal with combinatorial optimisations aiming to reduce the cost, idle time or optimise other important key performance indicators (KPIs) across multiple operation centres each with multiple possible processing pathways [1], [2], [3]. Most of the JSS problems (JSSPs) are NP-hard, with a small number of exceptions. That is the computational requirements for obtaining an optimal solution grow exponentially as the size of the problem increases. Exact approaches based on explicit or implicit enumerations, such as branch and bound, dynamic programming and integer linear programming are therefore only applicable to problems of small sizes [4]. For large scheduling problems, inexact solutions are usually of greater appeal as a trade-off between efficiency and exactness. Typical inexact solutions uses heuristics, meta heuristics and other AI approximate solutions, such as genetic algorithms, Tabu search, simulated annealing, constraint programming, artificial neural network and fuzzy logic [5].

The success of intelligent JSS systems is based on an accurate estimation of the processing time of each individual operation on each processing way at each operation centre, regardless of the approaches used for the implementation of

978-1-5386-3066-2/17 $31.00 © 2017 IEEE
DOI 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.26

the systems. The processing time of an operation through a machine can be readily estimated based on the nature of the job and the characteristics of the machine. However, it is difficult to estimate the completion time for the manual collate and pack area in the manufacturers of POS and POP in print industry due to the complexity of operational arrangement and the infinite variety of the POS and POP displays. This has often paralysed the intelligent job shop scheduling system and increased the actual time spent on collation to the estimate by between 50% and 200 % based on the evaluation form domain experts, particularly noted on large promotions. This can even destroy the profitability of taking on the commercial order.

This paper proposes a solution in such a context aiming to accurately estimate the completion time of each operation under different conditions in the manual collate and pack area to enable the implementation and effective actions of intelligent job shop scheduling system. In particular, the system is implemented by a fuzzy inference system initialised by expertise knowledge, knowing that there is usually no data available or readily obtainable for the collate and pack area [6], [7], [8], [9]. Then, the initialised system is further developed dynamically and adaptively upon its deployment whilst performing inferences. By acknowledging the sensitivity of real-world commercial data, two illustrative examples were used in the paper to demonstrate the work of the proposed system [10].

The rest of the paper is organised as follows. Section II briefs the theory underpinning of the proposed system. Section III details the proposed system. Section IV shows the effectiveness of the proposed approach through illustrative examples. Section V concludes the paper with future work pointed out.

II. Background

A. Fuzzy Inference and Interpolation

Fuzzy sets and systems represent and reason on vague information that arises due to the lack of sharp boundaries. The most widely applied fuzzy systems are fuzzy inference systems, such as the Mamdani inference [11] and the TSK inference [12]. In particular, these inference systems are able to represent nonlinear and high dimensional decision making problems as fuzzy rule bases. Rule bases are either translated from expert knowledge, or extracted from data sets [11], [13]. Given an input, a fuzzy inference system produces a system output by referring to those rules in the rule base whose antecedents overlap with the given input. However, a fuzzy inference system will fail if the given input is not covered by any rule in the rule base. Fuzzy rule interpolation (FRI) was proposed to address this limitation [6], [9], [14].

FRI is essentially a fuzzy extension of piece-wise linear or polynomial interpolation or extrapolation [15]. Accordingly, FRI enjoys the advantages of both fuzzy logic in terms of uncertainty management and piece-wise linear or polynomial interpolation or extrapolation by means of knowledge generalisation. Indeed, except being used as a supplementary to conventional fuzzy inference systems to work with sparse rule bases, FRI can also be used for system simplification for complex fuzzy models. Given an input which does not overlap with any rule antecedents in the rule base, the two closest rules can be identified for a given fuzzy distance metric, which is an aggregation (usually weighted average) of the distances between the antecedent items and their counterparts in the observation [9]. As the proposed system in this work is developed upon the scale and move transformation-based fuzzy rule interpolation (T-based FRI) approach, the underpinnings of this approach is outlined in the steps below [9]:

1) Calculate the relative placement factor based on the relative location of the observation regarding the two antecedents of the two rules by representing all the involved fuzzy sets as representative real numbers (named as representative values);
2) Interpolate the indeterminate rule whose antecedent vector shares the same representative value vector with the given input using the relative placement factor;
3) Compute the consequence of the interpolated rule by firing the intermediate rule using analogy-based reasoning such that the differentiation between the consequences of the intermediate rule and the interpolated rule is equivalent to that between the antecedents of the intermediate rule and the interpolated one.

The technical details of the above approach can be found in [9], [16], which are omitted here due to space limitation. FRI approaches have been further developed from different perspectives. For instance, adaptive fuzzy interpolation was proposed to guarantee the interpolated results are consistent throughout the inference processes [17], [18], [19], [20], [21]; backward fuzzy interpolation was proposed to support backward inference and allow flexible interpolation when certain antecedents are missing from the observation [22]; and rough-fuzzy rule interpolation was proposed for both representing the knowledge involving higher order uncertainty and facilitating rule interpolation with such knowledge [23]. FRI approaches have also been extended to support Type-2 fuzzy sets, such as [24]. Besides the extensions on the inference mechanism, FRIs have been applied to a number of real world applications, such as [25], [26], [27].

B. Job Shop Scheduling

Job shop scheduling problem (JSSP) has been a classical operation research problem or combinational optimisation problem in the communities of Operation Research and Artificial Intelligence, respectively [5], [28]. JSSP is NP-hard and thus the optimal solution cannot be guaranteed even for small JSSP instances. In particular, JSSP involves a finite set of jobs to be processed on a finite set of machines. Each job comprises a set of tasks that must be performed on a machine within a certain set of capable machines with various constraints and costs, in a given job-dependent order. Therefore, JSSP is essentially a machine scheduling problem where jobs represent activities and machines represent resources and each machine can process at most one job at a time. A typical objective of this process is to minimise the total completion time required for all jobs (i.e., the make span) or the economic cost, although other key performance indicators (KPIs) may also individually or jointly be used in the objective function.

A great number of approaches have been proposed for JSSPs in the literature [1], [2], [3], which can be classified into three categories: 1) hard search, 2) hard search with heuristics, and 3) soft search. Hard search algorithms are able to produce optimal solutions of the problems, but they require exponential computational time. Typical hard search algorithms include branch and bound, integer linear programming and dynamic programming, amongst others. Various heuristic have been developed to speed up hard search, with or without sacrificing the optimal solutions. In contrast, soft search does not guarantee optimal solutions, but it targets near optimal solutions in reasonable computational time. Soft search approaches are developed mainly based on the advances in soft computing. Common soft search approaches include genetic algorithms, simulated annealing, ant colony optimisation and fuzzy logic, amongst others.

III. The Proposed System

Manufacturing efficiency can be significantly improved by employing the advance of recent artificial intelligence development, and this is also the case for the manufacturing of POS and POP which still involves manual operations. A POS or POP is usually manufactured in multiple stages, such as printing, cutting, collating and packing. There are typically multiple ways to process the tasks in each stage with different costs regarding different KPIs, such as the cost of resources and make span. Given an objective or multiple objectives which are expressed as an objective function or multiple objective functions of the costs, an optimal allocation of jobs to machines can be made using a job shop scheduling system.

Suppose that a simple POS job requires three operations, including printing, cutting and collating, and that there are five different types of printers, four types of cutters and three possible ways to implement the collating. Then there will be 5*4*3=60 possible ways to implement the job. Assume the objective is to minimize the make span. The task of intelligent JSS system is then to find the best way guided by the objective function to implement the job under the current context of the manufacturer regarding constraints from other jobs, machines, and the shop environment amongst other factors. This can be implemented using the techniques discussed in Section II-B, if accurate time estimation of job completion can be provided. The time costs for those stages which are processed by machines can be readily estimated based on the nature of jobs and the specification of machines. However, it is rather difficult to accurately estimate the time of completion for collation tasks given that multiple collation lanes in different sizes are usually manually dynamically planned by the collate and pack manager subject to the compliance of various resource constraints, which is addressed in this work.

A. Problem Description

The collate and pack area deals with all the collating, packing and other hand finishing processes. This area is traditionally treated as a single cost centre in manufacturing information management system, although practically multiple production lines (or lanes) are often applied in the collate and pack area for multiple jobs in parallel. The failure of capturing and representing this multiple parallel processing lanes has led to some consequences:

- Late jobs can only be discovered shortly before the deadline, which cause the use of expensive last minute resolution, such as premium shipping and expensive third party outsourcing.
- The collate and pack area work load cannot be properly (ofen means evenly) distributed (which leads to high operational cost) due to the lack of knowledge or vision of the near future job scheduling.
- The overall job scheduling fails, thanks to this broken link in the production chain.

The lanes run at the collate and pack area are planned and scheduled by the area manager using their expertise knowledge. For simplicity and rule of thumb, the manager usually only uses three types of lanes: i) small lanes each implemented by one worker, ii) medium lanes each implemented by 4 workers, and iii) large lanes each implemented by 8 workers. Of course, other types of lanes may also be implemented for exceptional situations in order to achieve a deadline. The manager usually pursues of both operational and managerial efficiency when set up lanes. For instance, if other constraints are not considered, a collate and pack task may be better finished on a medium lane with 4 workers within just about one shift, which otherwise needs to be finished by more than one shifts on a small lane with less workers (with extra management cost), or needs less than a shift on a large lane with more staff members (with less operational efficiency as the other half shift may be wasted). The expert knowledge is used in this work to initialise a fuzzy rule base for the estimation of the completion times of collate and pack tasks as discussed in the next subsection.

B. Rule Base Initialisation

According to the expert knowledge, the time for completion regarding a given manual task on a particular lane setup depends on a number of factors, typically including the object size, the task complexity, the number of parts, the total glue length, the number of applications of glue, the number of stables, and the number of folds. The unskilled nature of the assembly staff role implies a relatively short work life on this particular function. The worker may either move on to more skilled operations, be seasonal and stay with the company in a matter of months, weeks, or even days, or threshold to an accepted level. The skill of the worker is therefore not a factor or parameter in the scheduling. This domain knowledge, which may vary from manufacturer to manufacturer, is used in this work to generate an initial rule base. The process of converting expert knowledge, usually expressed as linguistic rules, to fuzzy rules is beyond the scope of this paper, and a classical example of such process can be found in [11]. Of course, if there are sufficient data available, data-driven approaches can also be used for rule base generation [13], but this is usually not the case for most of the manufacturers due to the lack of data. As each piece of knowledge may be of different quality, a weight is also assigned to each rule indicating the quality of the rule. The variables used in the rule base and their domains are detailed in Table I; and each rule is of one of the three following formats:

$$\begin{aligned}
R_1^j &: \textbf{IF } x_1 \text{ is } A_1^j \text{ and } x_2 \text{ is } A_2^j \text{ and } \cdots \text{ and } x_7 \text{ is } A_7^j, \\
&\quad \textbf{THEN } z_1 \text{ is } B_1^j \ (w_1^j) \\
R_4^j &: \textbf{IF } x_1 \text{ is } A_1^j \text{ and } x_2 \text{ is } A_2^j \text{ and } \cdots \text{ and } x_7 \text{ is } A_7^j, \\
&\quad \textbf{THEN } z_4 \text{ is } B_4^j \ (w_4^j) \\
R_8^j &: \textbf{IF } x_1 \text{ is } A_1^j \text{ and } x_2 \text{ is } A_2^j \text{ and } \cdots \text{ and } x_7 \text{ is } A_7^j, \\
&\quad \textbf{THEN } z_8 \text{ is } B_8^j \ (w_8^j),
\end{aligned} \tag{1}$$

where A and B are fuzzy sets; w represents the weight of the rule expressing the confidence of the corresponding domain knowledge; R_1^j, R_4^j and R_8^j represents the j^{th} rule regarding a small lane, a medium lane and a large lane, respectively.

TABLE I. VARIABLES AND THEIR DOMAINS

Var	Meaning	Domain
x_1	Object size	{very small, small, medium, large, very large}
x_2	Task complexity	{very easy, easy, medium, complex, very complex}
x_3	No. of parts	A fuzzy integer number in a certin range
x_4	Total glue length	A fuzzy real number in a certain range
x_5	No. of glue applications	A fuzzy integer number in certain range
x_6	No. of staples	A fuzzy integer number in a certain range
x_7	No. of folds	A fuzzy integer number in a certain range
z_1	Time on a small lane	A fuzzy real number in a certain range
z_4	Time on a medium lane	A fuzzy real number in a certain range
z_8	Time on a large lane	A fuzzy real number in a certain range

Note that the rules above represent the rules of thumb in practice during the off-peak seasons. During the peak time, the simple application of these rules can lead to failure to meet the dispatch deadlines. Therefore, more ad hoc lanes (lanes with any number of workers) are often arranged during peak times in an effort to meet the dispatch deadlines. Indeed, the accurate estimation of the completion time for ad hoc lanes are of crucial importance in the implementation of solutions for JSSP for manufacturing efficiency, which is required to enable global optimisation of the manufacturing process. In other words, the times for completion need to be accurately estimated to cover all the combinations of inputs from the variable input domain with regard to various lanes of different sizes, which can be collectively represented as rules with multiple antecedents and multiple consequences as:

$$\begin{aligned}
R^j &: \textbf{IF } x_1 \text{ is } A_1^j \text{ and } x_2 \text{ is } A_2^j \text{ and } \cdots \text{ and } x_7 \text{ is } A_7^j \\
&\quad \textbf{THEN } z_1 \text{ is } B_1^j \ (w_1^j) \text{ and } z_2 \text{ is } B_2^j \ (w_2^j) \text{ and } \cdots \\
&\quad \text{and } z_m \text{ is } B_m^j \ (w_m^j).
\end{aligned} \tag{2}$$

The above rule with m consequences can be re-expressed as m rules with single consequence as:

$$\begin{aligned}
R_k^j &: \textbf{IF } x_1 \text{ is } A_1^j \text{ and } x_2 \text{ is } A_2^j \text{ and } \cdots \text{ and } x_7 \text{ is } A_7^j \\
&\quad \textbf{THEN } z_k \text{ is } B_k^j \ (w_k^j),
\end{aligned} \tag{3}$$

where $k = \{1, 2, \cdots, m\}$. The complete rule base is impossible to be implemented based on the domain knowledge of the collate and pack area manager due to the complexity of the problem. Fortunately, this can be solved by adapting the recently proposed experience-based rule base generation and adaptation approach [10] as detailed in the next subsection.

C. Rule Base Adaptation

The framework of the adapted rule base generation and adaptation system is outlined in Fig. 1. The system mainly comprises of three parts: a rule base, a FRI subsystem (particularly a transformation-based approach used in this work) and a rule base revision mechanism. In particular, for a newly quoted job, the values for variables listed in Table I are estimated by the in-house job estimating team and are used as the inputs for the transformation-based FRI subsystem. The decisions led by FRI are forwarded to the intelligent job scheduling system for planning and scheduling. The interpolated results from FRI are then reviewed upon the completion of the job and the review result represented as a performance index is used to support the rule base revision. The rule base initialisation has been discussed in Section III-B, and the FRI subsystem and rule base revision are discussed below.

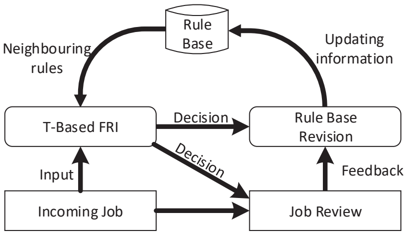
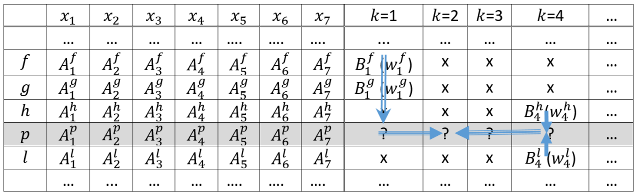


Fig. 1. Rule base adaptation

1) Rule Interpolation: The initialised rule base is very sparse. Suppose now there is a new collate and pack task which can be described as $x_1 = A_1^p, x_2 = A_2^p, \cdots x_7 = A_7^p$ as shown in the gray row in Table II. This new input is not covered by any rule in the rule base. Then the neighbouring rules need to be identified to support the interpolation of completion times for the given new task based on different lane setups [10]. Suppose that the closest neighbouring rules in the sparse rule base regarding the given input are explicitly shown in Table II, including:

$$\begin{aligned}
R_1^f &: \textbf{IF } x_1 \text{ is } A_1^f \text{ and } x_2 \text{ is } A_2^f \text{ and } \cdots \text{ and } x_7 \text{ is } A_7^f \\
&\quad \textbf{THEN } z_1 = B_1^f \ (w_1^f), \\
R_1^g &: \textbf{IF } x_1 \text{ is } A_1^g \text{ and } x_2 \text{ is } A_2^g \text{ and } \cdots \text{ and } x_7 \text{ is } A_7^g \\
&\quad \textbf{THEN } z_1 = B_1^g \ (w_1^g), \\
R_4^h &: \textbf{IF } x_1 \text{ is } A_1^h \text{ and } x_2 \text{ is } A_2^h \text{ and } \cdots \text{ and } x_7 \text{ is } A_7^h \\
&\quad \textbf{THEN } z_4 = B_4^h \ (w_4^h), \\
R_4^l &: \textbf{IF } x_1 \text{ is } A_1^l \text{ and } x_2 \text{ is } A_2^l \text{ and } \cdots \text{ and } x_7 \text{ is } A_7^l \\
&\quad \textbf{THEN } z_4 = B_4^l \ (w_4^l).
\end{aligned} \tag{4}$$

TABLE II. THE INFERENCE PROCESS FOR CASE 1



	x_1	x_2	x_3	x_4	x_5	x_6	x_7	k=1	k=2	k=3	k=4	...
	...	...	...	...	....	....	....	...	...	...	...	...
f	A_1^f	A_2^f	A_3^f	A_4^f	A_5^f	A_6^f	A_7^f	$B_1^f\ (w_1^f)$	x	x	x	...
g	A_1^g	A_2^g	A_3^g	A_4^g	A_5^g	A_6^g	A_7^g	$B_1^g\ (w_1^g)$	x	x	x	...
h	A_1^h	A_2^h	A_3^h	A_4^h	A_5^h	A_6^h	A_7^h	x	x	x	$B_4^h(w_4^h)$	...
p	A_1^p	A_2^p	A_3^p	A_4^p	A_5^p	A_6^p	A_7^p	?	?	?	?	...
l	A_1^l	A_2^l	A_3^l	A_4^l	A_5^l	A_6^l	A_7^l	x	x	x	$B_4^l\ (w_4^l)$	...
	...	...	...	...	....	....	....	...	...	...	...	...

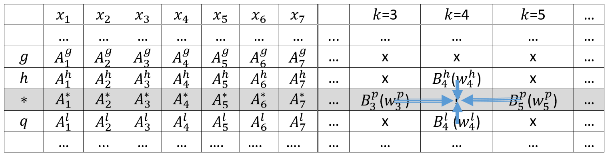
In order to estimate the time for completion by a small lane with one worker, the two closest rules R_1^f and R_1^g in the rule base are identified [10]. From this, the value of variable z_1 can be estimated by means of fuzzy extrapolation using the approach briefed in Section II-A. Similarly, the time for completion by a medium lane with four workers can be estimated using neighbouring rules R_4^h and R_4^l through fuzzy interpolation. From this, the time lengths of completion with 2-worker and 3-worker lanes can be interpolated from these two extrapolated and interpolated rules by artificially taking the number of workers on the lane as a rule antecedent attribute. The time lengths of completion based on any other lane setups can be either interpolated or extrapolated in the same way. Note that some of the interpolated/extrapolated rules that have been proven accurate which does not present in the existing rule base will be added in the rule base. This means that the rule base will become more and more dense after the deployment of the proposed system to enable the system to adaptively learn from practice. Then the choosing of the closest neighbouring rules for interpolation/extrapolation can be different with the situation discussed above.

Suppose that the part of the rule base in the running example has been updated as illustrated in Table. III (because interpolated/extrapolated rules R_3^p and R_4^p have been added in the rule base during the rule base revision which is discussed later in Section III-C2). Another new task presents which can be described as $x_1 = A_1^* \simeq A_1^p, x_2 = A_2^* \simeq A_2^p, \cdots x_7 = A_7^* \simeq A_7^p$ as shown in the gray row in Table III. In this case, it is clear that neighbouring rules R_4^h and R_4^l can be used for interpolation and denote the interpolated results as $B_4^{p\prime}$. In the same time, by artificially taking the number of workers on lanes as an extra rule attribute, the completion time can also be estimated from rules R_3^p and R_5^p and the result is denoted as $B_4^{p\prime\prime}$. In this case, the final result is a weighted aggregation of these two interpolated results in order to generate a global result, that is:

$$B_4^p = \lambda B_4^{p\prime} + (1-\lambda) B_4^{p\prime\prime}, \tag{5}$$

where λ is problem specific and 0.5 is used as a default value.

TABLE III. THE INFERENCE PROCESS FOR CASE 2



	x_1	x_2	x_3	x_4	x_5	x_6	x_7		k=3	k=4	k=5	...
	...	...	...	...	...	...	...	...	...	...	...	...
g	A_1^g	A_2^g	A_3^g	A_4^g	A_5^g	A_6^g	A_7^g	...	x	x	x	...
h	A_1^h	A_2^h	A_3^h	A_4^h	A_5^h	A_6^h	A_7^h	...	x	$B_4^h(w_4^h)$	x	...
$*$	A_1^*	A_2^*	A_3^*	A_4^*	A_5^*	A_6^*	A_7^*	...	$B_3^p(w_3^p)$		$B_5^p(w_5^p)$	...
q	A_1^l	A_2^l	A_3^l	A_4^l	A_5^l	A_6^l	A_7^l	...	x	$B_4^l(w_4^l)$	x	...
	...	...	...	...	....	....	....	...	...	...	...	...

2) Rule Base Revision: The rule base keeps being revised based on its performance whilst it performs fuzzy inferences after the system is deployed. This is mainly achieved using a feedback mechanism. Upon the completion of a collate and pack task, the real completion time is compared with the estimated completion time; and the comparison result is used to determine the quality of the interpolated rule expressed as the weight of the interpolated rule. In particular, the weight w of an interpolated/extrapolated rule is defined as:

$$w = e^{-\frac{(t-t')^2}{30}}, \tag{6}$$

where t and t' represent the actual and estimated completion times, respectively. The relation between $(t - t')$ and w is illustrated in Fig. 2. It is clear that the maximum value of weight is 1, which represents the time predicted by the system exactly matches the time spent in the real case.

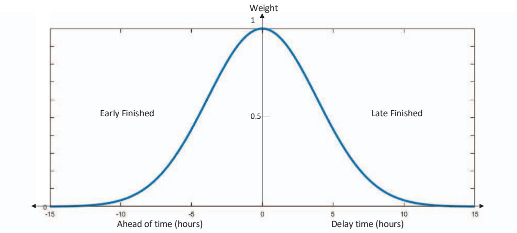


Fig. 2. The relation between weight and error

The interpolated rule may be used to replace an existing rule in the rule base, extend the existing rule base or simply be ignored. The framework of the rule base updating procedure is illustrated in Fig. 3. An interpolated rule that has been proven accurate will be added into the rule base if there is not any similar rules included in the rule base. Suppose that R^* is an interpolated rule, which is represented as "**IF** x_1 is A_1^* and x_2 is A_2^* and $\cdots$ and x_7 is A_7^* **THEN** $z_k = B_k^*$ (w_k^*)", and that rule R is a rule in the existing rule base which is represented as "**IF** x_1 is A_1 and x_2 is A_2 and $\cdots$ and x_7 is A_7 **THEN** $z_k = B_k$ (w)". The similarity degree $S(R^*, R)$ can then be computed as:

$$S(R^*, R) = \frac{\sum_{j=1}^{7} S(A_j^*, A_j) + S(B^*, B)}{7+1}, \tag{7}$$

where $S(A_j^*, A_j)$ represents the similarity degree between the j^{th} antecedents of the interpolated rule R^* and existing rule R, and $S(B^*, B)$ indicates the degree of similarity between the two consequences. Different approaches have been developed for similarity calculation between fuzzy sets [29]. For instance, if triangular fuzzy sets are employed, each fuzzy set A can be represented as $A = \{a_1, a_2, a_3\}$, where (a_1, a_3) is the support of the fuzzy set and a_2 is the core or normal point of the fuzzy set. In this case, the degree of similarity between two triangular fuzzy sets A_j^* and A_j can be calculated as:

$$S(A_j^*, A_j) = 1 - \frac{\sum_{k=1}^{3} |a_{jk}^* - a_{jk}|}{3}. \tag{8}$$

Given a similarity threshold δ, if the similarity degrees between the interpolated rule and all existing rules in the rule base are less than δ, a potential new rule for the rule base is identified. From this, if the weight of this newly interpolated rule w^* is greater than a given weight threshold φ (i.e., $w^* > \varphi$), this newly interpolated rule will be added into rule base; otherwise, the interpolated rule will be ignored due to its poor quality. If the similarity degree between the interpolated rule and one or more existing rules are greater than the given threshold δ, the set of similar rules will be determined. In this case, the system will compare the weights of all these rules,

TABLE IV. THE INITIALISED RULE BASE

No.	Antecedents							Consequents							
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8
1	(0.05,0.15,0.25)	(0.05,0.15,0.25)	(1,2,3)	(0.8,1.5,1.8)	(1,2,3)	(1,2,3)	(1,2,3)	7.2(1)							
2	(0.05,0.15,0.25)	(0.05,0.15,0.25)	(4,6,7)	(3.0,3.5,4.0)	(5,7,8)	(4,6,8)	(3,4,5)	12.0(1)							
3	(0.2,0.35,0.5)	(0.4,0.6,0.8)	(7,9,10)	(8.2,10.5,11.2)	(6,7,8)	(7,9,10)	(5,6,7)				4.9(1)				
4	(0.4,0.6,0.8)	(0.7,1,1)	(12,14,15)	(18.5,20.8,22.5)	(7,9,10)	(13,14,16)	(7,8,9)				8.9(1)				
5	(0.7,1,1)	(0.7,1,1)	(17,18,20)	(28.5,30.5,31.2)	(10,11,12)	(18,20,21)	(9,10,11)								6.0(1)

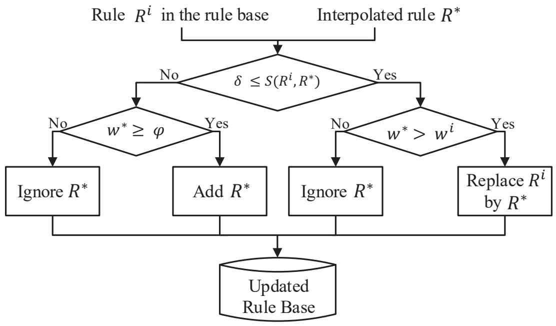


Fig. 3. The flowchart of rule base revision

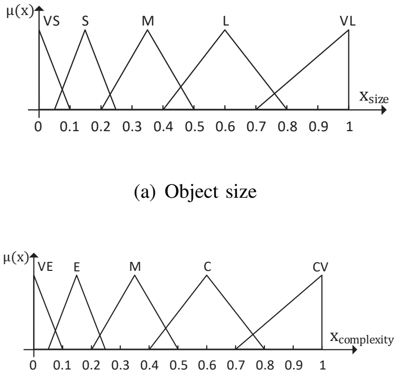
and only keep the rule with the highest weight value in the identified set. This action ensures that only the most accurate rule within the set is kept in the rule base, such that the rule base is concise and also of good generalisation ability. By following the rule base revision progress, the number of rules in the rule base and accordingly the system complexity can be controlled by adjusting the similarity threshold δ and weight threshold φ.

IV. ILLUSTRATIVE EXAMPLES

Three illustrative examples are taken in this section to demonstrate the working procedure of the proposed system, due to the sensitivity of the the real-world commercial data. In these examples, the value of λ in Equation 5 is pre-defined as 0.5, the threshold for similarity degree (δ) is set to 0.8, and the threshold (φ) for weight comparison is set to 0.8. For simplicity and to facilitate understanding, only triangular membership functions are used in the example to represent fuzzy sets.

A. Model Construction

Suppose that the collate and pack area is able to accommodate 8 different sizes of lanes. As listed in Table I, the model takes 7 inputs which jointly describe a given manual task, and it predicts the time of completion for a particular type of lane of certain size. In particular, two inputs, including the object size and the task complexity, take values from fuzzy partitioned variable domain based on the domain knowledge as shown in Fig. 4; and other inputs are fuzzy numbers. These fuzzy numbers are usually provided by the in-house job estimating team. For instance, for a given task, the in-house job estimating team may estimate that the number of parts is between 5 and 8, but it is most likely to be 7, though the production department has the final say and may even route through different machines. Then a triangular fuzzy number (4, 7, 8) will be used as the input for the attribute of the number of parts.



(a) Object size

(b) Task complexity

Fig. 4. Fuzzy partition of antecedent attributes

B. Scenario 1

Suppose that the initialised rule base only contains 5 fuzzy rules (R_1^1, R_1^2, R_4^3, R_4^4, *and* R_8^5), which is extracted from the domain knowledge of the area manager, as shown in Table IV. There is a collate and pack task given as $I = (x_1 = (0.1, 0.2, 0.3), x_2 = (0.1, 0.2, 0.3), x_3 = (4, 5, 6), x_4 = (2.0, 2.2, 2.8), x_5 = (3, 4, 5), x_6 = (2, 4, 5), x_7 = (2, 3, 5))$, which obviously does not overlap with any rule antecedents in the given rule base. In order to enable global planning and scheduling by an intelligent job shop scheduling system, the completion times of the task on various lanes are estimated first using FRI (and the transformation-based FRI is particularly used in this example).

There are two rules available with the consequence being the completion time on a lane with 1 worker, two rules available with consequence being the completion time on a lane with 4 workers, and one rule available with consequence being the completion time on a lane with 8 workers in the initialised rule base. From these rules, the completion time required for the given task on a lane with 1 worker (i.e., $z_1 = 9.37$) can be interpolated from neighbouring rules R_1^1 and R_1^2; and that on a lane with 4 workers (i.e., $z_4 = 2.24$) can be extrapolated using neighbouring rules R_4^3 and R_4^4. From this, the time of completion on the rest of lanes for the given task can either be interpolated or extrapolated based on the previously generated rules R_1^* and R_4^* by artificially taking the number of workers on the lane as an input attribute. The generated results are listed in Table V.

After feeding the results into the job shop scheduling system (which is out the scope of this paper), a globally

TABLE V. Times of Completion for Scenario 1

Antecedents							Consequents							
x_1	x_2	x_3	x_4	x_5	x_6	x_7	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$
(0.1,0.2,0.3)	(0.1,0.2,0.3)	(4,5,6)	(2.0,2.2,2.8)	(3,4,5)	(2,4,5)	(2,3,5)	**9.37**	**4.69**	**3.12**	**2.24**	**1.87**	**1.56**	**1.34**	**1.17**

TABLE VI. The Updated Rule Base

No.	**Antecedents**							**Consequents**							
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$
1	(0.05,0.15,0.25)	(0.05,0.15,0.25)	(1,2,3)	(0.8,1.5,1.8)	(1,2,3)	(1,2,3)	(1,2,3)	7.2(1)	**5.36(0.8)**		**1.7(0.8)**				
2	(0.1,0.2,0.3)	(0.1,0.2,0.3)	(4,5,6)	(2.0,2.2,2.8)	(3,4,5,)	(2,4,5)	(2,3,5)	**9.37(0.95)**			**2.24(0.98)**				
3	(0.05,0.15,0.25)	(0.05,0.15,0.25)	(4,6,7)	(3.0,3.5,4.0)	(5,7,8)	(4,6,8)	(3,4,5)	12.0(1)	**8.97(0.8)**		**2.9(0.8)**				
4	(0.2,0.35,0.5)	(0.4,0.6,0.8)	(7,9,10)	(8.2,10.5,11.2)	(6,7,8)	(7,9,10)	(5,6,7)				4.9(1)				
5	(0.4,0.6,0.8)	(0.7,1,1)	(12,14,15)	(18.5,20.8,22.5)	(7,9,10)	(13,14,16)	(7,8,9)				8.9(1)				
6	(0.7,1,1)	(0.7,1,1)	(17,18,20)	(28.5,30.5,31.2)	(10,11,12)	(18,20,21)	(9,10,11)								6.0(1)

TABLE VII. The Details of Similarity Degree Calculation

i	$S(x_1^*, x_1^i)$	$S(x_2^*, x_2^i)$	$S(x_3^*, x_3^i)$	$S(x_4^*, x_4^i)$	$S(x_5^*, x_5^i)$	$S(x_6^*, x_6^i)$	$S(x_7^*, x_7^i)$	$S(z_1^*, z_1^i)$	$S(R_4^*, R_4^i)$
1	0.75	0.75	0.4	0.59	0.5	0.54	0.6	0.77	**0.61**
2	0.75	0.75	0.88	0.65	0.6	0.6	0.82	0.78	**0.73**

optimised solution was generated which has planned the task on a lane with one worker, and such a lane was finally scheduled in the collate and pack area for the given task. Upon the completion of the task, the actual time taken for this task is $t = 9.85$. From this, the weight of the interpolated rule is calculated as $w_1^* = 0.95$ based on Equation 6. Therefore, the newly interpolated rule can be represented as:

$$\begin{aligned} R_1^* : \textbf{IF } & x_1 \text{ is } (0.1, 0.2, 0.3) \text{ and } x_2 \text{ is } (0.1, 0.2, 0.3) \text{ and} \\ & x_3 \text{ is } (4, 5, 6) \text{ and } x_4 \text{ is } (2.0, 2.2, 2.8) \text{ and} \\ & x_5 \text{ is } (3, 4, 5) \text{ and } x_6 \text{ is } (2, 4, 5) \text{ and} \\ & x_7 \text{ is } (2, 3, 5) \\ & \textbf{THEN } z_1^* = 9.37 \ (0.95). \end{aligned} \tag{9}$$

According to the rule revision procedure discussed in Section III-C2, once a new rule has been interpolated, the similarity degree between the newly interpolated rule and the existing rules in the rule base are computed to determine the actions for rule base revision. Given the newly interpolated rule as shown above in Equation 9, the similarity degrees between it and all existing rules in the rule base are calculated. Note that there are only rules R_1^1 and R_1^2 whose consequences are the time of completion on a lane with one worker. The sub-results and the final results of the calculation based on Equation 7 are summarised in Table VII. As the similarity degrees are less than the given threshold $\delta = 0.8$ and the weight of the rule ($w_1^* = 0.95$) is greater than the predefined threshold $\varphi = 0.8$, this interpolated rule R_1^* is added into rule base.

C. Scenario 2

The rule base keeps being revised whilst the system performs. Suppose that the rule base has now been updated as shown in Table VI, and a new task, which is similar to the one discussed in Scenario 1, appears. Therefore, the time lengths of completion on different lanes need to be estimated. Of course, the completion time on lanes with 1 worker and and 4 workers can be directly acquired from rules R_1^2 and R_4^2. The time of completion on a lane with 2 workers can be interpolated either from R_2^1 and R_2^3, or R_1^2 and R_4^2 by artificially taking the number of workers as a rule antecedent. In particular, result $z_2^{2'} = 6.64$ is interpolated using neighbouring rules R_2^1 and R_2^3; and result $z_2^{2''} = 6.99$ is interpolated using neighbouring rules R_1^2 and R_4^2 by considering the number of workers as an input attribute. In this case, the final result is estimated as the average of the two interpolated results based on Equation 5, and the final aggregated result is $z_2^2 = 6.82$. The estimation of the completion time on other types of lanes can also be implemented in one of the ways discussed above, but the calculation details are omitted.

Suppose that the lane with 2 workers has been selected by the job scheduling system, and the actual time taken for this manual task is $t = 9.6$. Based on Equation 6, the weight of the interpolated rule is determined as $w_2^* = 0.79$. Once the newly interpolated rule is constructed, the similarity between this interpolated rule R_2^* and the existing rules R_2^1 and R_2^3 are calculated as: $S(R_2^*, R_2^1) = 0.59$ and $S(R_2^*, R_2^3) = 0.60$, which are both less than the given threshold $\delta = 0.8$. According to the rule base revision procedure as shown in Figure 3, the interpolated rule R_2^* therefore is ignored, and the rule base keeps unchanged.

D. Scenario 3

Suppose that now another collate and pack task appears as $I = (x_1 = (0.5, 0.7, 0.9), x_2 = (0.4, 0.7, 0.8), x_3 = (10, 11, 12), x_4 = (19.5, 21.2, 22.0), x_5 = (7, 8, 9), x_6 = (11, 12, 13), x_7 = (7, 8, 9))$. The times of completion on lanes with 1 worker, 2 workers and 4 workers for the given task can be either interpolated or extrapolated from the rule base shown in Table VI. In particular, the completion time on lanes with 1 worker and 2 workers can be extrapolated from neighbouring rules (R_1^2, R_1^3) and (R_2^1, R_2^3), respectively; and the completion time on a lane with 4 worker can be interpolated from neighbouring rules R_4^4 and R_4^5. The time estimation based on other lane setup can then be interpolated or extrapolated from the previously interpolated and extrapolated rules by artificially taking the number of rules as an input attribute. The estimated results for all lane setup are listed in Table VIII.

The job scheduling system finally selects the lane with 4 workers during the operation optimisation stage. After the job is completed, the weight for this interpolated rule is calculated as $w_4^* = 0.78$. The degrees of similarity between this interpolated rule and the existing rules are computed as: $S(R_4^*, R_4^1) = 0.26$, $S(R_4^*, R_4^2) = 0.40$, $S(R_4^*, R_4^3) = 0.46$, $S(R_4^*, R_4^4) = 0.73$, $S(R_4^*, R_4^5) = 0.82$. It is clear that a similar rule R_4^5 to the newly interpolated rule exists in the rule

TABLE VIII. TIMES OF COMPLETION FOR SCENARIO 2

Antecedents							Consequents							
x_1	x_2	x_3	x_4	x_5	x_6	x_7	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$
(0.5,0.7,0.9)	(0.4, 0.7, 0.8)	(10, 11, 12)	(19.5, 21.2, 22.0)	(7,8 ,9)	(11,12,13)	(7,8,9)	**27.20**	**13.60**	**9.07**	**6.7**	**5.44**	**4.53**	**3.89**	**3.40**

base as $S(R_4^*, R_4^5) = 0.82$. Then, the weight of the interpolated rule R_4^* is compared with that of the existing rule R_4^5, and $w_4^5 = 1 > w_4^* = 0.78$. Consequently, the interpolated rule is discarded and the rule base remains as it is.

V. CONCLUSIONS

This paper presented a fuzzy model which estimates the time of completion for manual collate, assembly and pack tasks on different lanes with a variety of product configurations and sizes. The completion time estimation provides the necessary prerequisite in implementation of intelligent job shop scheduling systems for POS and POP manufacturers where collate and pack tasks are still performed manually. The model is implemented by adapting the recently proposed experience-based fuzzy rule interpolation approach where the model is initialised by the limited domain knowledge and under constant enhancement after deployment. The working of the system was demonstrated using three illustrated examples.

Although promising, the work can be further improved in different directions. Firstly, the proposed system was developed particularly for POS and POP manufacturers in this paper, but the underpinning artificial intelligence approach is applicable to any industry with manual tasks involved in the manufacturing processes. Also, it is interesting to integrate the model here into a job shop scheduling system to provide a complete solution. In addition, it would be desirable to run simulations of factory loadings in real-time so that commercial management decisions can be made on the basis of accurate dispatch capabilities. This in turn could be used to drive recognised forms of Kaizen style process improvement in a beneficial feedback loop. Finally, it is desirable to apply the work to real-world applications to further validate and evaluate its effectiveness and efficiency.

REFERENCES

[1] Banu Çaliş and Serol Bulkan. A research survey: review of ai solution strategies of job shop scheduling problem. *Journal of Intelligent Manufacturing*, 26(5):961–973, 2015.

[2] Imran Ali Chaudhry and Abid Ali Khan. A research survey: review of flexible job shop scheduling techniques. *International Transactions in Operational Research*, 23(3):551–591, 2016.

[3] Ian Miguel and Qiang Shen. Fuzzy rrdfcsp and planning. *Artificial Intelligence*, 148(1):11 – 52, 2003.

[4] M.C. Gomes *, A.P. Barbosa-Pvoa, and A.Q. Novais. Optimal scheduling for flexible job shop operation. *International Journal of Production Research*, 43(11):2323–2353, 2005.

[5] Michael Pinedo. *Scheduling*. Springer, 2015.

[6] L. Kóczy and K. Hirota. Approximate reasoning by linear rule interpolation and general approximation. *International Journal of Approximate Reasoning*, 9(3):197–225, 1993.

[7] Q. Shen and L. Yang. Generalisation of scale and move transformation-based fuzzy interpolation. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 15(3):288–298, 2011.

[8] L.Yang and Q. Shen. Closed form fuzzy interpolation. *Fuzzy Sets and Systems*, 225:1 – 22, 2013. Theme: Fuzzy Systems.

[9] Z. Huang and Q. Shen. Fuzzy interpolative reasoning via scale and move transformations. *Fuzzy Systems, IEEE Transactions on*, 14(2):340–359, 2006.

[10] J. Li, H. P. H. Shum, X. Fu, G. Sexton, and L. Yang. Experience-based rule base generation and adaptation for fuzzy interpolation. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 102–109, July 2016.

[11] E. H. Mamdani. Application of fuzzy logic to approximate reasoning using linguistic synthesis. *Computers, IEEE Transactions on*, C-26(12):1182–1191, 1977.

[12] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(1):116–132, Jan 1985.

[13] Y. Tan, J. Li, M. Wonders, F. Chao, H. P. H. Shum, and L. Yang. Towards sparse rule base generation for fuzzy rule interpolation. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 110–117, July 2016.

[14] J. Li, Y. Qu, H. P. H. Shum, and L. Yang. *TSK Inference with Sparse Rule Bases*, pages 107–123. Springer International Publishing, Cham, 2017.

[15] Longzhi Yang, Zheming Zuo, Fei Chao, and Yanpeng Qu. Fuzzy interpolation systems and applications. In *Fuzzy Control Systems*. InTech, 2017.

[16] Z. Huang and Q. Shen. Fuzzy interpolation and extrapolation: A practical approach. *Fuzzy Systems, IEEE Transactions on*, 16(1):13–28, 2008.

[17] L.Yang and Q. Shen. Adaptive fuzzy interpolation. *Fuzzy Systems, IEEE Transactions on*, 19(6):1107–1126, Dec 2011.

[18] L. Yang and Q. Shen. Adaptive fuzzy interpolation with prioritized component candidates. In *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, pages 428–435, June 2011.

[19] L. Yang and Q. Shen. Adaptive fuzzy interpolation with uncertain observations and rule base. In *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, pages 471–478, June 2011.

[20] Shou-Hsiung Cheng, Shyi-Ming Chen, and Chia-Ling Chen. Adaptive fuzzy interpolation based on ranking values of polygonal fuzzy sets and similarity measures between polygonal fuzzy sets. *Information Sciences*, 342:176 – 190, 2016.

[21] L. Yang, F. Chao, and Q. Shen. Generalised adaptive fuzzy rule interpolation. *IEEE Transactions on Fuzzy Systems*, 2017. DOI: 10.1109/TFUZZ.2016.2582526.

[22] S. Jin, R. Diao, C. Quek, and Q. Shen. Backward fuzzy rule interpolation. *IEEE Transactions on Fuzzy Systems*, 22(6):1682–1698, Dec 2014.

[23] Chengyuan Chen, Neil Mac Parthalin, Ying Li, Chris Price, Chai Quek, and Qiang Shen. Rough-fuzzy rule interpolation. *Information Sciences*, 351:1 – 17, 2016.

[24] L. Yang, C. Chen, N. Jin, X. Fu, and Q. Shen. Closed form fuzzy interpolation with interval type-2 fuzzy sets. In *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*, pages 2184–2191. IEEE, 2014.

[25] J. Li, L. Yang, H. P. H. Shum, G. Sexton, and Y. Tan. Intelligent home heating controller using fuzzy rule interpolation. In *UK Workshop on Computational Intelligence*, 2015.

[26] L. Yang, J. Li, G. Fehringer, P. Barraclough, and G. Sexton. Intrusion detection system by fuzzy interpolation. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2017)*, 2017.

[27] J. Li, L. Yang, X. Fu, F. Chao, and Y. Qu. Dynamic qos solution for enterprise networks using tsk fuzzy interpolation. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2017)*, 2017.

[28] Lynwood A Johnson and Douglas C Montgomery. *Operations research in production planning, scheduling, and inventory control*, volume 6. Wiley New York, 1974.

[29] Shi-Jay Chen and Shyi-Ming Chen. Fuzzy risk analysis based on similarity measures of generalized fuzzy numbers. *IEEE Transactions on Fuzzy Systems*, 11(1):45–56, Feb 2003.