Hawassa University Journals

Volume III
Pages: 69-89
Article Number: 851-Article Text-4219-1-2-20230613

www.journals.hu.edu.et/hu-journals/index.php/ejet/index

Ethiopian Journal of Engineering and Technology

Development of a decision support system for cutting tools planning and control using fuzzy case-based reasoning

Fentahun Moges Kasie; fentahunmk@gmail.com

Hawassa University, Institute of Technology, Department of Industrial Engineering, Hawassa, Ethiopia

Abstract.

Cutting tools management is one of the main issues in metal cutting operations. This important problem has not been adequately studied in the past. Most of the problems in cutting tools management were addressed using optimization techniques. This study proposed a decision support system (DSS) to articulate this problem by combining artificial intelligence (AI) and multiple attribute decision-making (MADM) tools. The proposed DSS retrieves the most similar historical cases to adapt their cutting tool requirements to the current part orders. The DSS integrates case-based reasoning (CBR), rule-based reasoning and fuzzy set theory (FST) in AI. It uses the analytic hierarchy process (AHP) and distance from target methods of multiple-attribute decision-making (MADM) in decision analysis. Cases were represented using an Object-Oriented (OO) approach to characterize cases for their tool set requirements. A numerical example was illustrated to show the soundness of the proposed methodological approach.

***Keywords**:* Decision support systems, case-based reasoning, analytic hierarchy process, fuzzy set theory, cutting tool planning.

1. Introduction

The next generation of manufacturing (Industry 4.0) is characterized by frequent changes of production requirements such as flexibility, responsiveness, improved quality of products and efficient utilization of resources. Cutting tools are one of the major components for metal cutting industries to meet these requirements. Managing the flow of these components is as significant as managing the flow of parts in flexible manufacturing systems (FMS) (Gray et al., 1993; Rahimifard and Newman, 1997; Özbayrak and Bell, 2003). Gray et al. (1993) and Rahimifard and Newman (2000) suggested that tool management strategies should be integrated with system design, planning and control activities to improve resources utilization and reduce operational costs. Cutting tools can contribute to 25-30% of the total operating costs (Gray et al., 1993; Buyurgan et al., 2004; Rahimifard and Newman, 1997). In order to address this problem, several studies were proposed in the past. These were reviewed in different studies (Buyurgan et al., 2004; Gray et al., 1993; Meseguer and Gonzalez, 2008). These proposed tool-planning approaches were based on linear and nonlinear optimization techniques, heuristics and domain knowledge-based expert systems. Optimization models are computationally intractable as the number of input variables increases. Heuristic algorithms are unable to find the global optimum solution. In rule-based expert systems, it is monotonous to represent the complex domain knowledge from experts in the form of rules alone (Aamodt and Plaza, 1994; Kolodner, 1992).

In addition, these previous methods are static in nature to accommodate knowledge uncertainties and imprecisions in dynamic situations where much is unknown and solutions are open-ended. In reality, tool assignment and control problems are unstructured and open-ended. Recent studies revealed that only about 20 percent information circulates in terms of structured and numeric data in companies; however, the remaining 80 percent information is hidden in unstructured forms (Tseng and Chou, 2006; Feki et al., 2013). The right part-cutting tools assignment strategies need to be flexible, simple and compressive to integrate both structured and unstructured data. These kinds of flexible, integrative, simple and comprehensive systems are strongly recommended in the Industry 4.0 (Wang et al., 2016; Rojko, 2017; Lu, 2017; Lee et al., 2015; Lee et al., 2014).

In order to accommodate these situations, this study proposed an intelligent decision support system (DSS) by integrating the fuzzy versions of case-based reasoning (CBR) and analytic hierarchy process (AHP). In additions, the proposed DSS uses many rules to support the case reasoning process. Cases were represented using an object-oriented (OO) approach to characterize cases for their tool set requirements. They were applied to retrieve prior cases that had the most similar assigned tool sets to the current part orders using case similarity measures. The proposed DSS used a fuzzy CBR method to represent unstructured (fuzzy) information from product and process attributes of part orders. It utilized a fuzzy AHP approach to prioritize case attributes in the case retrieval process. To the knowledge of the authors, this approach has not been used in the past in cutting tool assignment and control problems. This study illustrated a numerical analysis to test the soundness of the proposed DSS on a machining operation center using a computer-based laboratory environment.

The remainder of this paper is organized as follows: Section 1 reviews the literature. Section 3 describes the proposed system. In Section 4, a numerical example is illustrated. Section 5 discusses the findings. Finally, conclusions are forwarded in Section 6.

2. Literature review

2.1. Artificial intelligence (AI) in DSS

Artificial intelligence (AI) plays an important role in the Industry 4.0. by converting typical resources into intelligent objects that can sense, act, and behave within a smart environment (Zhong et al., 2017). Intelligent industries take the advantage of advanced information to achieve flexible, smart, and reconfigurable industrial processes to articulate dynamic and stochastic markets (Lee et al., 2018; Zhong et al., 2017). As Zhong et al. (2017) reviewed, AI technologies enable smooth flows of information as it is needed across holistic industrial supply chains. An intelligent DSS uses extensively computer-based methods in AI (Holsapple and Whinston, 1996). Based on this notion, this section discusses the interactions among AI elements such as CBR, fuzzy set theory (FST), rule-based reasoning (RBR) and OO case representation approach.

CBR is one of the popular analogical reasoning and machine learning paradigms in AI. It emerged in the beginning of 1980s from the works of Roger Schank on dynamic memory that focuses on remembering past episodes as cases and scripts as situation patterns for new problem solving and learning strategies (Schank, 1982). A new problem is solved by reusing and/or adapting successful experiences to the current similar situations and retaining it for future retrieval (Kolodner, 1991). CBR has been used in a variety of problem solving and interpretive tasks including design, planning, diagnosis, explanation, justification, classification, predicting, etc. (Kolodner, 1992). According to Watson and Marir (1994), CBR systems are preferred by practitioners and researchers because of four reasons such as an explicit domain model is not required for knowledge elicitation; identifying key case attributes is easier than creating an explicit model; large volumes of information can be managed using database management techniques; and case maintenance is easier.

CBR is one of the useful methodologies to develop advisory systems in intelligent DSS (Beemer and Gregg, 2008). Advisory systems provide recommendations to human decision makers in unstructured and complex situations. As Kolodner (1991), in uncertain and dynamic environments, where much is unknown and solutions are open-ended, CBR systems are preferred over other AI techniques because they can propose different solution alternatives to their users based on partially available knowledge. According to Aamodt and Plaza (1994), CBR systems are capable to utilize incremental learning from accumulated experiences to solve new problems, which means its effectiveness increases through time as more and more cases are retained in the case library. In addition, CBR systems can be efficiently trained using relatively small amount of data; however, other AI systems like ANNs cannot do so (Oh and Kim, 2007).

Aamodt and Plaza (1994) described their general CBR cycle in terms of four "Re"s in Figure 1. (a) *Retrieve* the most similar prior case to the current problem, (b) *Reuse* the knowledge in the retrieved case; (c) *Revise* the retrieved prior case in order adapt to the new case; and (4) *Retain* the final solution as the learned case for future retrieval.

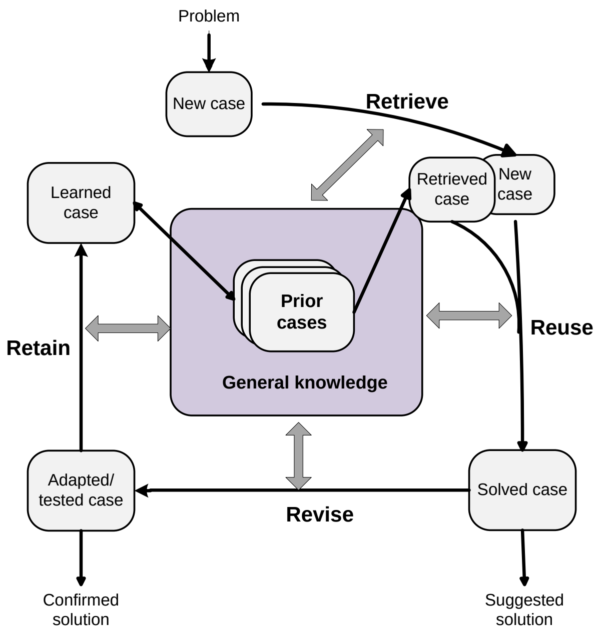

Figure 1. CBR cycle adopted from (Aamodt and Plaza, 1994)

General domain knowledge represented in the form of rules in ruled-based expert systems is usually required to support the CBR process. RBR uses domain knowledge from well-defined theory as rules to infer about the similarity between new and past cases. It may range from weak to strong depending upon the problem type (Aamodt and Plaza, 1994). Guiding rules may be developed and integrated into CBR systems to improve their performances. Integrating rule-based reasoning (RBR) and CBR is one of the popular strategies to make CBR systems more productive (Prentzas and Hatzilygeroudis, 2007; Golding and Rosenbloom, 1996; Kasie et al., 2017a).

According to Bergmann et al. (2006) and Kolodner (1993), a case was defined as a formalized piece of knowledge representing the reasoners" previous experiences. Watson and Marir (1994) stated that prior cases are represented in terms of their several features as problem descriptions and corresponding solutions as solution descriptions. Because real situations are usually uncertain and vague situations, knowledge can be reasonably expressed using fuzzy sets to grade the degree of membership of objects within [0,1] rather than crisp values of {0, 1} (Zedah, 1965). A case is said to be fuzzy if at least one of its features is expressed using linguistic terms, fuzzy numbers or fuzzy sets (Zimmermann, 2001). Incorporating FST into CBR systems enhances the decision-making process through utilizing imprecise experiences stored in the form of cases (de Mántaras and Plaza, 1997). FST increases the flexibility and applicability of CBR approaches in real situations (Li and Ho, 2009; Faez et al., 2009). In addition, in situations in which an integrated application of CBR and RBR is required, the rules in decision-making are vaguely described using linguistic terms such as strong similarity, weak similarity, etc. to define the level of similarities between new cases and prior cases (see e.g. Kasie et al., 2017a; Kasie and Bright, 2019). Kasie et al. (2017a) reviewed recent applications in fuzzy CBR in various problem domains.

The right case representation approach is required to meet the objectives of case reasoners. Several case representation approaches were proposed in the past. However, an OO approach is widely accepted by CBR system software developers (Watson and Marir, 1994) because of its structured and compact-data representation capability to address memory related issues, software reusability and easiness for users to understand (Pal and Shiu, 2004). OO case representation methods are particularly useful in complex problem domains in which cases/objects with different structures occur and each object is described by a set of features (Bergmann and Stahl, 1998). They provide more flexibility and modularity to the system in consideration through utilising the inheritance principles (Bergmann et al., 2006).

2.2. Multiple-attribute decision-making (MADM) in CBR

Cases are represented in terms of their multiple features in case libraries. A real case retrieval usually uses MADM methods. MADM is used to rank a finite set of decision alternatives using well-defined criteria (Kahraman et al., 2008). In CBR, prior cases are treated as alternative solutions and case features are treated as multiple attributes(Chang et al., 2008). The roles of MADM in CBR are to: (a) weight case attributes; (b) find case similarities between new and prior cases; and (c) select the most similar prior cases that match to the current problems.

Traditional MADM methods treat both attribute values and their weights as crisp numbers (Chen and Hwang, 1992). In reality, such kinds of approaches are not convincing because the values of attributes and their weights can be expressed in terms of linguistics terms, fuzzy numbers and fuzzy sets. In order to address such complex situations, the current MADM approaches incorporate fuzziness associated with human decision-making strategies. Bellman and Zadeh (1970) initially articulated the concepts of FST into MADM problems. In addition, Baas and Kwakernaak (1977) proposed the first fuzzy MADM approach that was widely accepted as the classical fuzzy MADM framework. The fuzzy versions of MADM studies were reviewed and elaborated (see e.g. Chen and Hwang, 1992; Ribeiro, 1996; Carlsson and Fullér, 1996; Kahraman et al., 2008; Mardani et al., 2015)).

2.2.1. Weighting case attributes

In MADM analyses, the determination of the weights of attributes is a crucial part for a multi-attribute value analysis (Weber and Borcherding, 1993). Attributes weighting requires domain knowledge elicitation to make the case reasoning meaningful (Park and Han, 2002). A key factor in the case retrieval process is weighting case attributes (An et al., 2007; Pal and Shiu, 2004). In this regard, the AHP is an important knowledge and experience elicitation method to prioritise decision-making criteria (Saaty, 1994). It is a systematic approach to acquire and represent experts" domain knowledge for rating case attributes (Park and Han, 2002). Presently, the AHP a popular MADM method with vast applications (Forman and Gass, 2001; Demirel et al., 2008; Xu et al., 2007; Lee et al., 2008). Vaidya and Kumar (2006) reviewed its different applications.

The AHP has unique capabilities to decompose and structure any complex decision problems hierarchically; determine the relative importance of attributes/sub-attributes using pairwise comparisons; represent human judgements in terms of numerical values; measure the consistency of pairwise comparisons; and hierarchic composition/synthesis (Forman and Gass, 2001; Wind and Saaty, 1980; Zahedi, 1986; Saaty, 2003). According to Ho (2008), the popularity of the AHP is because of its easiness to use, flexibility and capability for integrating with other approaches. Ishizaka and Labib (2011) and Ho (2008) reviewed the developments of the AHP applications and its integrated applications with other methods respectively. Some recent studies revealed that the uses of integrating the AHP and CBR systems to prioritise case attributes (see e.g. Kuo, 2010; An et al., 2007; Changchien and Lin, 2005; Faez et al., 2009; Wu et al., 2008; Park and Han, 2002). FST was not directly addressed in the classical AHP (Chen and Hwang, 1992). The classical AHP was extended into its fuzzy version to address vagueness in human decision-making (Van Laarhoven and Pedrycz, 1983; Buckley, 1985). In addition, Demirel et al. (2008) reviewed several applications of fuzzy AHP methods.

2.2.2. Similarity measure and case retrieval

Distance from target method is one of the widely accepted MADM approaches because it is simple, easy to understand and straightforward to describe its idea (Chen and Hwang, 1992; Kahraman, 2008). In CBR systems, the target is the current problem and solution alternatives are prior cases. Distance-based case retrieval approaches mostly calculate the Euclidean distance between any two cases using feature-value pairs, which constitute the required cases. The most similar case is selected using this calculated distance (Liao et al., 1998; Kumar et al., 2009). A prior case with the shortest distance from the target problem is the most similar case. This case retrieval approach is known as the Nearest Neighbor (NN) pattern matching function using the weighted Euclidean distance measure. Many case retrieval methods were proposed in the past namely NN, inductive learning, knowledge guided and validated approaches (Pal and Shiu, 2004). Among these, the NN is the most common and popular pattern recognition function in *n*-dimensional Euclidean space (Pal and Shiu, 2004;Park and Han, 2002; Faez et al., 2009).

When different types of attributes constitute cases, the best way to measure the distance between cases is finding the distance/similarity measures with respect to individual case attributes and then calculating the cumulative weighted distance/similarity between two cases using the normalized weights of case attributes and the individual distance measures (Kolodner, 1993; Watson, 1999). Slonima and Schneider (2001) presented different equations for measuring similarities with respect to different types of case attributes such as crisp, range and fuzzy values. In addition, Faez et al. (2009) applied three different approaches to measure similarities for crisp and fuzzy case attributes.

2.2.3. Fuzzy ranking

A number of fuzzy ranking methods were proposed to defuzzify and rank fuzzy values in MADM analyses. Most of these proposed approaches are computationally cumbersome and intractable when a large number of alternatives and attributes are considered. In order to articulate this problem, Chen and Hwang (1992) reviewed the pros and cons of the current fuzzy ranking approaches and proposed a new fuzzy MADM approach to reduce the computational difficulties of the reviewed approaches. In their new approach, the following three steps are included. (a) any linguistic terms should be projected into their equivalent trapezoidal/triangular fuzzy numbers, which are scaled into any real numbers within the range of [0, 1]; (b) these fuzzy numbers should be converted into their estimated crisp values using the right fuzzy ranking approaches; and (c) an appropriate MADM approach must be applied depending upon the problem type.

This approach avoids some computational difficulties by converting any fuzzy data into crisp values before any MADM operations are undertaken. Although, its inputs are either fuzzy data or a combination of fuzzy and crisp data, its outputs are usually crisp numbers in the range of [0, 1]. Any complex problems with a combination of fuzzy data and crisp data can be easily accommodated with the help of this approach.

3. Proposed DSS

This section presents the methods applied in this study based on the theories discussed in Section 2 and the research problem stated in Section 2. The methods were used to propose an intelligent DSS. Its flow diagram is presented in Figure 2, which was adapted from Kasie et al. (2017b) . It was assumed that similar part orders require similar cutting tool sets when part orders are appropriately characterized. The principal methodological approach in this research was a CBR approach. The interactions among CBR, RBR, FST and MADM to develop the proposed DSS are discussed in this section.

3.1. Case construction and representation

In this paper, part orders were treated as fuzzy cases with multiple attributes. The crucial attributes of these orders were identified to construct both prior and new cases. The proposed DSS used Microsoft Excel tools to structure the features of part orders into case attributes. These tools are usually simple and efficient to perform simple matrix operations. In addition, Excel tools are easy to integrate them with Java applications. The attributes were used to determine case similarities between the current and prior cases for tool sets assignment strategies. This implemented the assumption that similar part orders demanded the same tool sets for their key operations. In this case, a lathe operation center was considered as a case. Three primary part attributes were identified and named part geometry, material and operation types. The primary attributes were hierarchically branched into sub-attributes. The identified case attributes were expressed using numerical values, nominal values, descriptive and linguistic terms. Because some attributes were described using linguistic terms, the part orders were treated as fuzzy cases.

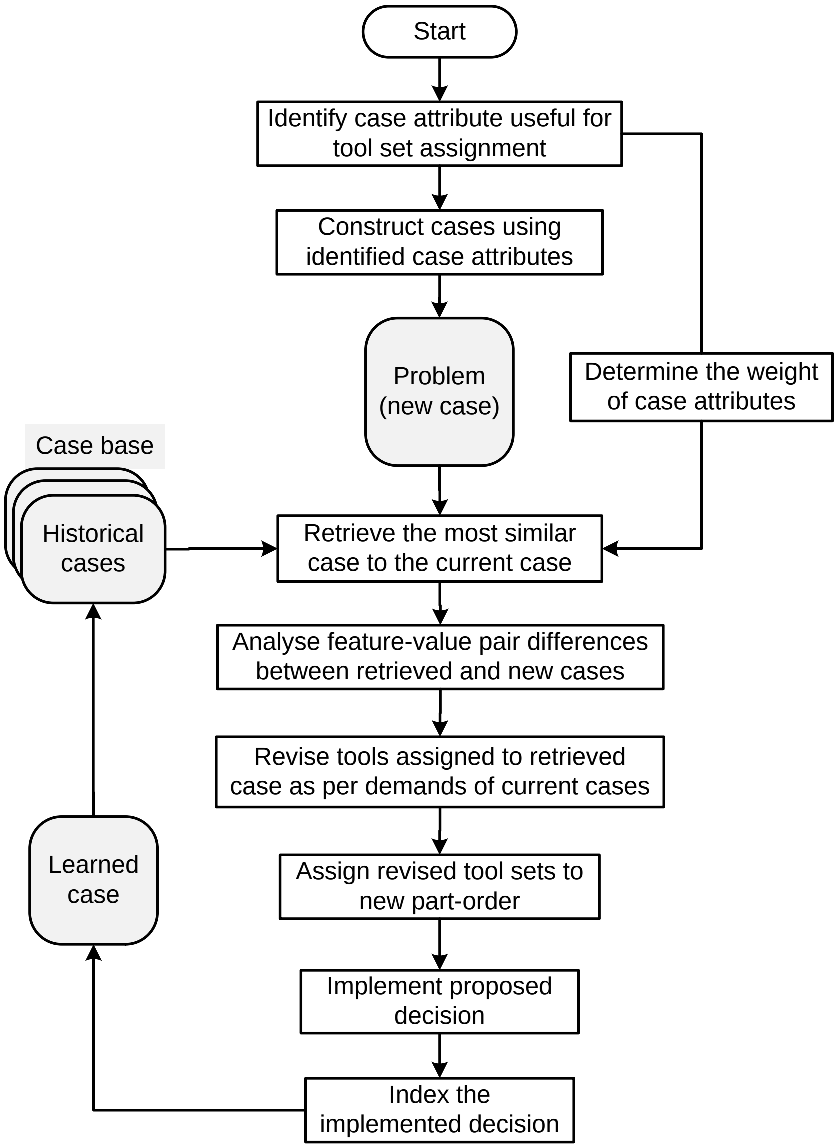

Figure 2: Flow diagram of the proposed DSS

Prior cases were represented using the identified key attributes as problem descriptions and their assigned tool sets as solution descriptions. New cases incorporated their problem descriptions alone and their solutions descriptions were retrieved and adapted from retrieved cases. A case representation scheme for the current part orders and prior cases is presented in Figure 3. An OO case representation approach was applied to create cases in the Java platform. This platform was employed because it is enriched with many in-built library classes and methods, simple and clean for managing memory issues in such complex situations.

In this case representation, linguistic terms were converted into their equivalent fuzzy numbers within [0, 1] with the help of the proposed eleven conversion scales indicated in Figure 4. Kasie et al. (2017a) applied the same conversion scales in a different problem domain using the proposal by Chen and Hwang (1992). The variable x is any real number within [0, 1] and $\mu(x)$ is the degree of membership of x to the linguistic terms in Figure 4. A case representation scheme for the current product orders and prior cases in the proposed DSS is depicted in Figure 3. The prior case representations included an additional resource named an assigned tool set as a solution description. The remaining components were used as problem descriptions, which were common to both new and prior cases.

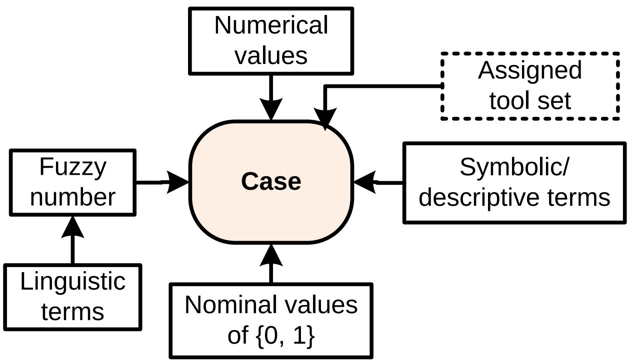

Figure 3. Fuzzy case representation scheme of part orders

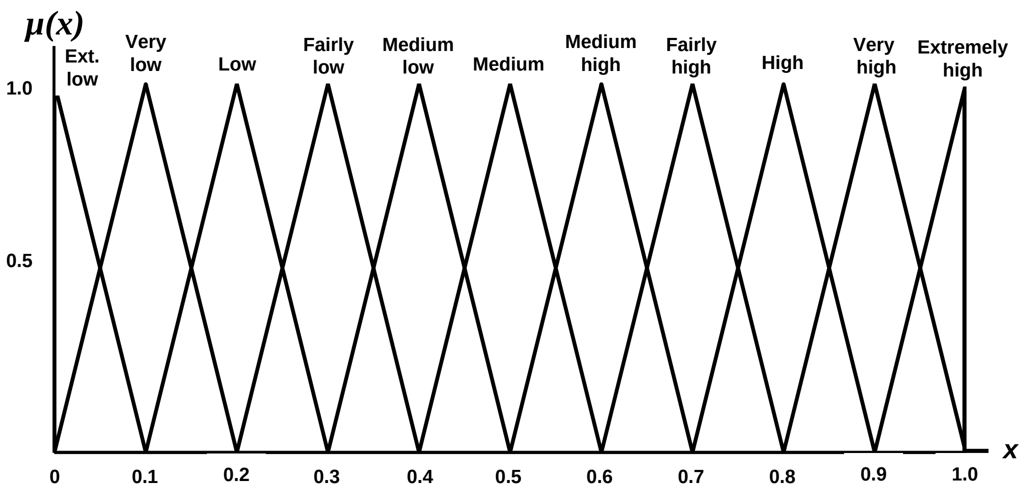

Figure 4. Conversion of linguistic attributes into fuzzy numbers (Kasie et al., 2017a)

3.2. Weighting case attributes and fuzzy ranking

After identifying the key part attributes, it was required to prioritize these case attributes. This was because not every attribute could have the same contribution to the case similarity measure. Weighting case attributes requires domain knowledge elicitation from experts to make meaningful the case reasoning process. In this aspect, the fuzzy AHP is popular to prioritize case attributes (Park and Han, 2002; Saaty, 1994). Using pairwise comparisons, the preference of one attribute over the other was expressed in terms of fuzzy linguistic terms like "equally preferred", "moderately preferred", "strongly preferred", etc., which are purely subjective and linguistic terms to define their boundaries.

Table 1 presents the relationships among fuzzy AHP-based linguistic terms, their equivalent fuzzy numbers and fuzzy reciprocals. The fuzzy numbers and their reciprocals were converted into their corresponding standard fuzzy numbers by dividing them with the maximum value of the universe of discourse, the number 10, according to the proposed approach by Chen and Chen (Chen and Chen, 2009). Table 2 shows the relationships among the linguistic terms and their standard forms within [0, 1]. Similar approaches were applied in other studies in different problem domains (Wu et al., 2008; Kasie et al., 2017a).

The standard fuzzy numbers were transformed into their corresponding crisp values by adopting a fuzzy ranking approach proposed by Chen and Chen (2009). The following equation was proposed to defuzzify the required fuzzy numbers. This approach is simple and prefers precise fuzzy numbers when two or more fuzzy numbers have the same mean value. After determining the crisp score of trapezoidal fuzzy numbers, A_{cs}, the classical AHP approach was used to prioritize case attributes in the same approach as Kasie et al. (2017a).

$$A_{cs} = \frac{A_{mean}}{1+A_{std}} \tag{1}$$

Where A_{mean} and A_{std} are the mean and standard deviation of a standard fuzzy number respectively.

Table 1. Linguistic terms, their equivalent fuzzy numbers and standardized fuzzy numbers (Kasie et al., 2017a).

AHP-based fuzzy linguistic terms	Equivalent		Standardized	
	Fuzzy number	**Fuzzy reciprocal**	**Fuzzy number**	**Fuzzy reciprocal**
Exactly equal	(1, 1, 1)	(1, 1, 1)	(0.1, 0.1, 0.1)	(1/10, 1/10, 1/10)
Equally preferred	(1, 1, 2)	(1/2, 1, 1)	(0.1, 0.1, 0.2)	(1/20, 1/10, 1/10)
Intermediate	(1, 2, 3)	(1/3, 1/2, 1)	(0.1, 0.2, 0.3)	(1/30, 1/20, 1/10)
Moderately preferred	(2, 3, 4)	(1/4, 1/3, 1/2)	(0.2, 0.3, 0.4)	(1/40, 1/30, 1/20)
Intermediate	(3, 4, 5)	(1/5, 1/4, 1/3)	(0.3, 0.4, 0.5)	(1/50, 1/40, 1/30)
Strongly preferred	(4, 5, 6)	(1/6, 1/5, 1/4)	(0.4, 0.5, 0.6)	(1/60, 1/50, 1/40)
Intermediate	(5, 6, 7)	(1/7, 1/6, 1/5)	(0.5, 0.6, 0.7)	(1/70, 1/60, 1/50)
Very strongly preferred	(6, 7, 8)	(1/8, 1/7, 1/6)	(0.6, 0.7, 0.8)	(1/80, 1/70, 1/60)
Intermediate	(7, 8, 9)	(1/9, 1/8, 1/7)	(0.7, 0.8, 0.9)	(1/90, 1/80, 1/70)
Extremely preferred	(8, 9, 10)	(1/10, 1/9, 1/8)	(0.8, 0.9, 1.0)	(1/100, 1/90, 1/80)

3.3. Case retrieval

The case retrieval process utilized the attribute-value pairs of the current and prior cases, and normalized weights as its input variables. A number of case retrieval methods have been proposed in the past. This study used the Nearest Neighbor (NN) pattern matching function using the inverse of the weighted Euclidean distance. This approach is popular and minimizes the limitations of other approaches as reviewed in Section 2.2.2. The weights of part attributes were normalized i.e. the sum of weights, $\Sigma w_i = 1.0$.

The weighted Euclidean distance between a target (new) part order p and a prior part order q, $dist(p,q)$ in n-dimensional Euclidean vector space was calculated as:

$$dist(p,q) = \sqrt{\sum_{i=1}^{n}[w_i * dist(a_i^p, a_i^q)]^2},\ dist(a_i^p, a_i^q) \in [0,1] \tag{2}$$

Where:
n is the number of case attributes.
w_i is the normalised weight of the ith case attribute.
$dist(a_i^p, a_i^q)$ is the distance measure between case p and case q with respect to the ith case attribute alone.
a_i^p and a_i^q are the values of the ith attribute for cases p and q respectively.

In this study, $dist(a_i^p, a_i^q)$ was calculated first with respect to every attribute and the cumulative $dist(p,q)$ was calculated using w_i and $dist(a_i^p, a_i^q)$ values as indicated in Equation (2). The $dist(a_i^p, a_i^q)$ values were determined depending upon the nature of the individual case attributes.

In the case of numerical attributes:

$$dist(a_i^p, a_i^q) = \frac{|a_i^p - a_i^q|}{a_{i,max} - a_{i,min}},\ a_i^p \ \&\ a_i^q \in [a_{i,min}, a_{i,max}] \tag{3}$$

Where $a_{i,min}$ and $a_{i,max}$ are the minimum and maximum value of the ith attribute respectively.

For nominal/descriptive attributes:

$$dist(a_i^p, a_i^q) = |a_i^p - a_i^q| = \begin{cases} 1\ if\ a_i^p \neq a_i^q \\ 0\ if\ a_i^p = a_i^q \end{cases} \tag{4}$$

In the case of fuzzy attributes, trapezoidal fuzzy numbers were considered and Equation (5) was adapted from a method of similarity measure of generalized fuzzy numbers proposed by Hejazi et al. (2011). The method combines the geometric distance, perimeter, area and height of the fuzzy numbers. In this case, all fuzzy numbers considered in this study are normal and convex (they have equal heights = 1, see Figure 4) as the approaches applied in Faez et al. (Kasie and Bright, 2019, Faez et al., 2009). When trapezoidal fuzzy numbers are in a standard form of $a_i^p = (a_{i,1}^p, a_{i,2}^p, a_{i,3}^p, a_{i,4}^p)$ and $a_i^q = (a_{i,1}^q, a_{i,2}^q, a_{i,3}^q, a_{i,4}^q)$; and $0 \leq a_{i,1}^p \leq a_{i,2}^p \leq a_{i,3}^p \leq a_{i,4}^p \leq 1$ and $0 \leq a_{i,1}^q \leq a_{i,2}^q \leq a_{i,3}^q \leq a_{i,4}^q \leq 1$; then the individual distance from the target case was calculated as:

$$dist(a_i^p, a_i^q) = 1 - \left[\left(1 - \sum_{k=1}^{4}\frac{|a_{i,k}^p - a_{i,k}^q|}{4}\right) * \frac{min\left(P(a_i^p), P(a_i^q)\right)}{max\left(P(a_i^p), P(a_i^q)\right)} * \frac{min\left(A(a_i^p), A(a_i^q)\right)+1}{max\left(A(a_i^p), A(a_i^q)\right)+1}\right] \tag{5}$$

Where:

$P(a_i^p)$ and $P(a_i^q)$ are the perimeters; and $A(a_i^p)$ and

$A(a_i^q)$ the areas of trapezoidal fuzzy attributes of case p and case q respectively. The perimeters and areas of the trapezoidal fuzzy attributes were calculated using the following perimeter and area formulas:

$$P(a_i^p) = \sqrt{(a_{i,2}^p - a_{i,1}^p)^2 + 1} + \sqrt{(a_{i,4}^p - a_{i,3}^p)^2 + 1} + (a_{i,3}^p - a_{i,2}^p) + (a_{i,4}^p - a_{i,1}^p) \quad (6)$$

$$A(a_i^p) = \frac{1}{2}(a_{i,3}^p - a_{i,2}^p + a_{i,4}^p - a_{i,1}^p) \quad (7)$$

The $P(a_i^q)$ and $A(a_i^q)$ values were calculated in the same way.

From Equation (2), the values of $dist(a_i^p, a_i^q)$ are in the range of [0, 1]. The upper bound of the Euclidean distance, $dist_{ub}(p,q)$, is found when all the values of $dist(a_i^p, a_i^q)$ = 1. Similarly, the lower bound of the Euclidean distance, $dist_{lb}(p,q)$, is found when all the values of $dist(a_i^p, a_i^q)$ = 0 i.e. when cases p and q are identical (p = q). With reference to Equation (2), the $dist_{ub}(p,q)$ and $dist_{lb}(p,q)$ values were determined as:

$$dist_{ub}(p,q) = \sqrt{\sum_{i=1}^{n} w_i^2} \quad (8)$$

$$dist_{lb}(p,q) = 0 \quad (9)$$

Because distance and similarity are inversely related, the similarity between two cases p and q, $sim(p,q)$, can be found as follows (Liao et al., 1998; Kumar et al., 2009):

$$sim(p,q) = \frac{1}{1+\alpha[dist(p,q)]} \quad (10)$$

Where α is a positive constant. Its value depends on the inverse proportionality of similarity and distance. In this case, α = 1.0 was used by assuming that the inverse proportionality ratio is one to one (1:1).

From the inverse relationship, the lower bound of the similarity measure, $sim_{lb}(p,q)$, is found when the upper limit of the distance measure occurs. The $sim_{lb}(p,q)$ from Equation (8) and Equation (10) can be calculated as

$$sim_{lb}(p,q) = \frac{1}{1+dist_{ub}(p,q)} = \frac{1}{1+\sqrt{\sum_{i=1}^{n} w_i^2}} \quad (11)$$

Similarly, the upper bound of similarity between part orders p and q, $sim_{ub}(p,q)$, is found when the lower bound of distance occurs. From Equation (9) and Equation (10)

$$sim_{ub}(p,q) = \frac{1}{1+dist_{lb}(p,q)} = \frac{1}{1+0} = 1 \quad (12)$$

Referring to Equation (8) to Equation (10), the $sim(p,q)$ value is within [$sim_{lb}(p,q)$, 1.0]. All the above equations including basic rules were coded in the Java platform and incorporated in the proposed DSS. Using these equations, the DSS generated a list of similarity measures between the current case and prior cases while a new order was entering into the system. The DSS selected the maximum similarity measure on the similarity list using the Java library method "max(list)", which returns the maximum value from a list, in the "java.util.Collections" class. Using this returned value, any retrieved case with a higher similarity value to the current problem was selected for reuse and adaptation.

3.4. Case reuse and adaptation

After the case retrieval, the next step was revising the retrieved case for a solution proposal. In this regard, the proposed DSS was designed to present case attribute difference for case comparison and revision. Based on this difference, some cutting tools assigned to the retrieved cases were reused using attribute similarities. Others tools were either deleted or replaced and new tools were added to adapt the retrieved cases for the new part orders. Several (If…, Then….) rules were developed from the general domain-dependent knowledge to support the decision-making process. With reference to Figure 2, the proposed DSS was intended to incorporate the following rules based on the sameness of two cases.

- If the current and retrieved cases are identical ($sim(p,q)$ = 1), then the retrieved tool set should be directly reused without any revisions.
- If they are not identical ($sim(p,q)$ < 1), then revisions based on differences in attributes should be undertaken.

The second rule was expanded to consider specific attribute changes in order to incorporate other new cutting tools. For example, when new attributes are added to and/or deleted from a new order, then specific tools should be added to and/or deleted from the retrieved tool set by exerts.

3.5. Case retaining

Case retaining is one of the crucial tasks in CBR systems. It is useful to retrieve previously implemented decisions for future reuse and adaptations. A case library was created and implemented with the help of the "java.util.ArrayList" class in the Java platform. The implemented cases were indexed using "add (object)" function, which is one of the in-built methods of the Java "java.util.ArrayList" class. This method appends a new element at the end of a list.

4. Numerical analysis

The section implements the methods proposed in Section 3 using a numerical example. The numerical example was illustrated in a computer-based environment using a lathe-machine operations center, which was assumed to produce several rotating shafts for various purposes. This machining center was selected because of its simplicity for illustration.

4.1. Case representation and weighting attributes

Thirteen part attributes were identified by human experts to represent part orders as cases using an OO method. These attributes were assumed that they could strongly influence part-cutting tool assignment activities in specific metal cutting processes. They were structured hierarchically (Table 2) to weight their importance. The hierarchy incorporates three primary attributes: (a) part geometry, (b) material property, and (c) required operations types. They are crucial attributes for assigning cutting tools for specific part orders in metal cutting. The primary attributes were subdivided into their succeeding sub-attributes. The normalized weights of the major attributes and sub-attributes at their specific levels were evaluated using a fuzzy AHP. The hierarchical evaluation was performed using the concepts presented in Table 1 and Equation (1). The normalized weights of the thirteen case attributes were proportionally calculated and presented in Table 2. The detail of this calculation is similar to as presented by the authors in (Kasie et al., 2017a).

The thirteen part attributes were represented using numerical, nominal and fuzzy data (see Table 3). The diameter (*Di*) and turn-depth (*TD*) of workpieces were represented using numerical values in millimeter. The tolerance limit (*TL*) and surface finish (*SF*) of products and the hardness (*HD*) of construction materials were described in terms of linguistic terms and the terms were converted into fuzzy numbers referring to Figure 2. These features are mostly described using linguistics terms such as high, medium, low, etc. in machining processes rather than specific units. The material type (*MT*) and heat treatment type (*HT*) of construction materials were described in symbolic terms. Material compositions are usually described by specific terms such as carbon steel, aluminum, stainless steel, etc. Similarly, heat treatments are typically classified as normalized, annealed, etc. Machining operation types such as turning (*Tu*), facing (*Fa*), thread-cutting (*Th*), drilling (*Dr*), boring (*Bo*) and tapping (*Ta*) were expressed using nominal values of {0, 1.0}, to indicate whether a specific operation is required to produce a product.

Table 2. Hierarchy of case attributes and their weights

Attribute			Weight	
Primary	Middle	End	w_i calculation	w_i
Part geometry (0.297)	-	*Di* (0.229)	0.279x0.229	0.068
		TD(0.229)	0.279x0.229	0.068
		TL(0.271)	0.279x0.271	0.081
		SF(0.271)	0.279x0.271	0.081
Material property (0.332)	-	*MT*(0.371)	0.332x0.371	0.123
		HT(0.297)	0.332x0.297	0.110
		HD(0.332)	0.332x332	0.099
Operation types (0.371)	External (0.522)	Fa(0.254)	0.371x0.522x0.254	0.049
		Tu (0.421)	0.371x0.522x0.421	0.081
		Th (0.325)	0.371x0.522x0.325	0.063
	Internal (0.478)	Dr (0.372)	0.371x0.478x0.372	0.066
		Bo(0.314)	0.371x0.478x0.314	0.056
		Ta(0.314)	0.371x0.478x0.314	0.056

Four product orders (*PO*1-*PO*4) as new cases and two prior cases (*PC*1 and *PC*2) as prior cases are presented in Table 3 to illustrate the numerical example in the proposed DSS. The numbers of new and prior cases were limited for illustration purposed; however, the proposed system could manage a large number of cases. *TS*1 and *TS*2 were initially assigned tool sets to *PC*1 and *PC*2 respectively.

Table 3. Structured thirteen attributes of part orders/cases

Part order	Case/part order attributes													TS
	MT	*HT*	*TD*	*Di*	*TL*	*SF*	*HD*	*Tu*	*Fa*	*Th*	*Dr*	*Bo*	*Ta*	
*PO*1	Carbon steel	Normalize	35	120	0.5,0.6,0.7,0.8	0.8,0.9,0.9,1.0	0.5,0.6,0.7,0.8	1	0	1	0	1	0	-
*PO*2	Alloy steel	Anneal	45	160	0.6,0.7,0.8,0.9	0.5,0.6,0.6,0.7	0.6,0.7,0.8,0.9	1	0	1	1	0	1	-
*PO*3	Carbon steel	Normalize	30	120	0.5,0.6,0.7,0.8	0.7,0.8,0.9,1.0	0.5,0.6,0.7,8.0	1	0	1	0	1	1	-
*PO*4	Alloy steel	Anneal	43	170	0.7,0.8,0.9,1.0	0.5,0.6,0.6,0.7	0.6,0.7,0.8,0.9	1	1	1	1	0	1	-
*PC*1	Alloy steel	Anneal	40	150	0.7,0.8,0.8,0.9	0.6,0.7,0.7,0.8	0.6,0.7,0.8,0.9	1	0	0	1	0	1	*TS*1
*PC*2	Carbon steel	Normalize	25	90	0.6,0.7,0.8,0.9	0.8,0.9,0.9,1.0	0.5,0.6,0.7,0.8	1	1	1	0	1	0	*TS*2

4.2. Case similarity and retrieval

After determining the normalized weights of case attributes, the $sim(p,q)$ value was calculated using Equation (2). From the normalized weight (w_i) from Table 2, the proposed system automatically generated the value of $sim_{lb}(p,q)$ = 0.7766 and $sim_{ub}(p,q) = 1.0$ with the help of Equation (9) and Equation (10) respectively. Referring to these two values, the value of $sim(p,q)$ was determined within [0.7766, 1.0] for this case. Distances with respect to the individual attributes were calculated using from Equation (3) to Equation (5) depending on the nature of attributes.

The first two symbolic variables (*MT* and *HT*) were converted into nominal values of {0, 1} using Equation (4). In addition, some rules were developed and a Java in-built method was applied. If the material composition of the new and prior cases is described using identical strings, their distance measure is the numeric string "0"; otherwise, it is "1". The same approach was applied to the heat treatment. The Java library method "Integer.parseInt(numeric string)" that changes numeric strings into the same integer numbers was applied to return the integer values of {0, 1}. After applying the rules and the method, the individual distances, $dist(a_1^p, a_1^q)$ and $dist(a_2^p, a_2^q)$, were determined using Equation (4) for the first and second attributes respectively.

The individual distances for the next two numerical variables/attributes (*TD* and *Di*) such as $dist(a_3^p, a_3^q)$, and $dist(a_4^p, a_4^q)$ respectively were calculated using Equation (3). The lower limit values were arbitrarily set as 15.0 *mm* and 60.0 *mm* and the upper limit values were set as 45.0 *mm* and 180.0 *mm* for turning depth and diameter respectively.

The three fuzzy attributes (*TL*, *SF* and *HD*) were converted into their estimated trapezoidal/triangular numbers (see Fig. 4). For example, in the case of *PO*2 and *PC1*, the hardness of construction materials, *HD* = (0.6, 0.7, 0.8) is equivalent to the fuzzy term "fairly highly". The same conversion was employed to the remaining two fuzzy attributes. Eq. (5) was used to calculate individual distances, $dist(a_5^p, a_5^q)$, $dist(a_6^p, a_6^q)$ and $dist(a_7^p, a_7^q)$ from the 5th to the 7th attributes. For the five nominal attributes, from the 8th to the 13th attributes, Equation (4) was applied to calculate the individual distances from $dist(a_8^p, a_8^q)$ to $dist(a_{13}^p, a_{13}^q)$.

Using the normalized weights and individual distances, the weighted Euclidean distance between new and prior orders/cases was calculated by implementing Equation (2). Applying an inverse relationship between distance and similarity, the case similarity measures were computed using Equation (10). A list of similarity within the upper and lower bounds [0.7766,1.0] was generated from this computation. The Java in-built method “max(list)” was applied to select a prior case with the maximum similarity with the current case. For example, as the first new part order $PO1$ entered into the system, the proposed DSS found the similarities between $PO1$ and the two initial training samples as $sim(PO1, PC1)$ = 0.8331, and $sim(PO1, PC2)$ = 0.9082 using Equation (10). The maximum similarity from this list was returned as *java.util.Collections.max*(0.8331, 0.9082) = 0.9082. The most relevant and similar previous case to *PO*1 was *PC*2. This implied that the retrieved tool set was the one that was assigned to *PC*2 or *TS*2 (Table 4). This retrieved tool set was revised and retained for future retrieval. Using the same approach, the maximum similarities and the best similar cases (tool sets) were determined for the remaining three orders.

*PO*2*: java.util.Collections.max*(**0.9783**, 0.8387, 0.8314) = 0.9983 (*TS*1*)*
*PO*3*: java.util.Collections.max*(0.8394, 0.8950, **0.9440**, 0.8380) = 0.9440(*TS*3*)*
*PO*4*: java.util.Collections.max*(0.9489, 0.8428, 0.8273, **0.9523**, 0.8334) = 0.9523 (*TS*4*)*

To access the retrieved tool set (TS), the proposed DSS utilized two Java in-built methods in combination, “get(integer)” and “indexOf(object)” in the “java.util.ArrayList” class. These two functions were employed to return a case in its case library at a specific index and the index of the first matching case respectively. The index (location) of the retrieved case in its case-base and the index of the maximum similarity from its similarity list were identical in this system. For example, in case of *PO*2, the maximum value was the first element on the similarity list, and this implied that the most similar case in the case base to *PO*2 was the first prior case.

While the new orders from *PO*1 to *PO*4 were entering into the proposed system, the number of elements on the cases in the case-base increased by one after each new order was processed. For example, as the second new order *PO*2 just arrived at the system, the number of elements on the similarity list grew from 2 into 3 by considering *PO*1 as a learned case for future retrieval. As the last order arrived at the system, the number of prior cases in the case library was six. The results are summarized in Table 4.

Table 4. Summarized results from proposed DSS

New part (*PO*)	**Retrieved case**	**Similarity value**	**Retrieved TS for revision**	**Number of cases in case-base**	
				As *PO* arrived	**After *PO* processed**
*PO*1	*PC*2	0.9082	*TS*2	2	3
*PO*2	*PC*1	0.9783	*TS*1	3	4
*PO*3	*PC*3 (learned)	0.9440	*TS*3	4	5
*PO*4	*PC*4 (learned)	0.9523	*TS*4	5	6

As a retrieved tool set was assigned to a new case, the DSS added the copy of the new case into its case-base using the “Cloneable” interface by overriding the “clone()” function in the Java “Object” class. After processing each order, the previous case similarity list was cleared to create a new similarity list for the next new arrival using the Java in-built function “clear()”. It was used to keep the numbers of cases in the case library equals to the number of similarity values on the similarity list for each case retrieval. New cases were indexed in the case libraries using the library method “add(object)”, which appends the current case at the end of the list. To disclose the number of cases available every time in the case libraries, another library method “size()” was utilized. These three library methods were utilized from the “java.util.ArrayList” class.

4.3. Case revision and decision proposal

The proposed DSS was capable to presents case attribute variations between the current and retrieved prior cases for revisions. The rules proposed in Section 3.4 were implemented based on these variations. For instance, considering $PO2$, the similarity between the new and the retrieved cases was determined as $sim(PO2, PC1)$ = 0.9783 < 1.0. This implied that these two cases were not identical and revisions were enviable. In the case adaptation stage, cutters can be removed, replaced or added. When $PO2$ entered into the system, the major different between this case and the retrieved case $PC1$ was the thread-making operation. $PO2$ required this operation but it was not in $PC1$. In this case, a new cutter for thread making was added into the retrieved tool set $TS1$ and retained as $PC4$ with $TS4$ for future retrieval. Similarly, $PC3$ and $PC4$ were the revisions of $PC2$ and $PC1$ as $PO1$ and $PO2$ were retrieved as the best similar prior cases respectively (see Table 4).

5. Discussion

It was reviewed that part-cutting tool assignment is one of the complex issues in manufacturing when a new part order is received. Several mathematical and heuristic optimization approaches were proposed to solve cutting tool planning problems. These models were very complex and intractable to solve in real industrial situations. The proposed DSS in this study was relatively simple and highly flexible to articulate the stated problem situations using machine-learning algorithms. The proposed system started with two prior cases as training samples and it was gradually increased into six cases as the fourth order was processed (Table 4). This implied that the system was designed to update continuously the number of cases in its case library. This characteristic of the proposed system improved its effectiveness through time to accommodate dynamically changing manufacturing situations. In addition, this proposed system was efficiently trained using a few cases (two prior case in the numerical example); however, other systems like ANNs could not accommodate this problem (Oh and Kim, 2007).

From the methodological perspective, this study combined CBR, FST, RBR and MADM approaches for case retrieval and adaptation. This combination was useful to make the proposed DSS flexible in dynamic and uncertain situations. In case retrieval, revision and retaining, relatively simple analytical models, user-defined functions and in-built Java library methods were implemented in the Java platform. This implied that the proposed system utilized relatively easier methods to assign cutters to new products at operational levels of resource planning. In addition, the case representation in this research was simple, flexible, comprehensive and easy to understand by its users using an OO approach. Four different forms of case attributes (numerical, categorical, symbolic and verbal terms) were included to make the system flexible enough (Table 3 and Figure 4); however, the proposed DSS could articulate any other forms of knowledge and experiences depending upon the needs of the users of the system.

The uncertainty and vagueness associated with human reasoning and thoughts were articulated using fuzzy set theory. In the numerical example, it was realised that fuzzy case attributes could accommodate more flexibilities than numerical-valued attributes. For example, the numerical attributes named turning depth and diameter could not accommodate changes when new part orders unpredictably entered into the system with attribute values above the upper limits and/or below the lower limits of these attribute values. However, in the case of the fuzzy attributes such as tolerance limit, hardness and surface smoothness, there were no any upper and lower limit restrictions (Table 3).

Using the same approach, the proposed system can incorporate any other machining operations such as milling, drilling, grinding etc. To do this, system developers or users are required to identify and structure hierarchically key product attributes, which can influence part-cutting tool assignment activities. These attributes are used to create cases for part-cutter assignment. The weights such attributes must be hierarchically evaluated. For example, the shape of the workpiece can be very essential to the other machining operations because their workpieces can be different from cylindrical shapes. This shows that the performance of the proposed DSS is highly flexible depending upon its developers" capabilities to represent experts" knowledge and judgments. If case attributes are suitably selected and their weights are properly evaluated using the knowledge and experiences of experts at specific operations, the system can support the decision makers in the right way.

From the managerial perspective, operational managers can plan the required cutting tools in parallel to their product plans. They can enumerate the available cutters and purchase or manufacture the required cutting tools in the planning phases. This can minimize the unnecessary holding and downtime costs by stabilizing the flows of cutting tools for planned production periods. This improves the utilization of resources.

6. Conclusion

In this study, a novel DSS was proposed to articulate the problems of cutting tool assignment and control using a fuzzy CBR as the principal methodology. The proposed DSS was capable to retrieve the most similar prior cases to the current part orders. As presented in Table 4, the DSS started with two prior cases in the case library and finally it was increased into six cases after the fourth (final) order was processed. It was found that the DSS developed in this research was able to update continuously the number of cases in its case library. This feature could improve the effectiveness of the DSS by increasing the number of alternative solutions for the newly arriving orders through time to accommodate dynamically changing manufacturing situations. Further, it was shown that the proposed DSS was efficiently trained using a few well-defined cases as leaned experiences.

An OO approach was utilized to represent fuzzy cases using the combination of numerical values, nominal values, symbolic and fuzzy linguistic terms. The case representation was useful to accommodate the flexibility required in the DSS. A fuzzy CBR methodology was used to represent imprecise and uncertain knowledge in the case representation and retrieval, and decision-making processes. Instead of measuring precisions such as tolerance and surface smoothness in micrometers for cutting tool assignment, it was meaningful to describe them in terms of linguistic terms. In addition, a fuzzy AHP was used to elicit the knowledge and judgements of experts for weighting case attributes. It was implied that such kinds of knowledge representation approaches could emulate human thoughts in order to process imprecise knowledge in manufacturing situations rather than using optimization methods as sole solution approaches for cutting tool-planning problems.

Using similar procedures, the DSS could be extended to other machining operations such as milling, grinding, drilling, etc. by identifying and weighting the right case attributes as specific operation centers. Although the numerical example was illustrated using a few new and prior cases, the proposed DSS was capable to accommodate any number of part orders scheduled, prior cases and case attributes in order to address real product mix variation in manufacturing situations.

In the future, the proposed DSS can be tested in industrial environments using historical data from several metal cutting operation centers to validate its accuracy. In this case, more detained information can be required to represent part orders as cases. Depending on the pattern of industrial situations, this approach can be applied to classify or cluster cases. In this case, ***k***-NN algorithm can be used rather than retrieving a single past case. In addition, detail knowledge-based rules will be included to make the case retrieval process more effective and efficient in the future.

References

Aamodt, A. and Plaza, E. (1994), “Case-based reasoning: Foundational issues, methodological variations, and system approaches”, *AI Communications*, Vol.7 No.1, pp. 39-59.

An, S.-H., Kim, G.-H. and Kang, K.-I. (2007) “A case-based reasoning cost estimating model using experience by analytic hierarchy process”, *Building and Environment*, Vol.42 No.7**,** pp. 2573-2579.

Baas, S.M. and Kwakernaak, H. (1977), "Rating and ranking of multiple-aspect alternatives using fuzzy sets", *Automatica*, Vol.13 No.1**,** pp.47-58.

Beemer, B. A. and Gregg, D. G. (2008), “Advisory systems to support decision making”, in: Burstein, F. and Holsapple, C. W. (Eds), *Handbook on Decision Support Systems 1: Basic Themes*, Springer-Verlag, Berlin and Heidelberg, pp. 511-528.

Bellman, R.E. and Zadeh, L.A. (1970), “Decision-making in a fuzzy environment”, *Management Science*, Vol.17 No.4**,** pp. 141-164.

Bergmann, R., Kolodner, J. and Plaza, E. (2006), “Representation in case-based reasoning”, *The Knowledge Engineering Review*, Vol.20 No.3, pp. 209-213.

Bergmann, R. and Stahl, A. (1998), "Similarity measures for object-oriented case representations", in: Smyth, B. and Cunningham, P. (Eds), *Advances in Case-Based Reasoning*, Springer-Verlag, Berlin and Heidelberg, pp. 25-36.

Buckley, J.J. (1985) “Fuzzy hierarchical analysis”, *Fuzzy Sets and Systems*, Vol.17 No.3, pp.233-247.

Buyurgan, N., Saygin, C. and Kilic, S.E. (2004), “Tool allocation in flexible manufacturing systems with tool alternatives”, *Robotics and Computer-Integrated Manufacturing*, Vol. 20 No 4**,** pp. 341-349.

Carlsson, C. and Fullér, R. (1996), “Fuzzy multiple criteria decision making: Recent developments”, *Fuzzy Sets and Systems*, Vol.78 No.2**,** pp.139-153.

Chang, P.-C., Liu, C.-H. and Lai, R.K. (2008), “A fuzzy case-based reasoning model for sales forecasting in print circuit board industries”, *Expert Systems with Applications*, Vol.34 No.3, pp.2049-2058.

Changchien, S.W. and Lin, M.-C. (2005), “Design and implementation of a case-based reasoning system for marketing plans”, *Expert Systems with Applications*, Vol.28 No.1, pp. 43-53.

Chen, S.-J. and Hwang, C.-L. (1992), *Fuzzy Multiple Attribute Decision Making: Methods and Applications*, Springer-Verlag, Berlin.

Chen, S.-M. and Chen, J.-H. (2009), “Fuzzy risk analysis based on ranking generalized fuzzy numbers with different heights and different spreads”, *Expert Systems with Applications*, Vol.36 No.3, pp. 6833-6842.

De Mántaras, R.L. and Plaza, E. (1997), “Ccase-based reasoning: An overview”. *AI Communications,* Vol. 10 No.1 , pp. 21-29.

Demirel, T., Demirel, N.C. and Kahraman, C. (2008), “Fuzzy analytic hierarchy process and its application”, i*n:* Kahraman, C. (Eds), *Fuzzy Multi-criteria Decision Making: Theory and Applications with Recent Developments,* Springer, New York, NY, pp. 53-83.

Faez, F., Ghodsypour, S.H. and O”brien, C. (2009), “Vendor selection and order allocation using an integrated fuzzy case-based reasoning and mathematical programming model”, *International Journal of Production Economics*, Vol.121 No.2, pp.395-408.

Feki, J., Messaoud, I.B. and Zurfluh, G. (2013),“Building an XML document warehouse”, *Journal of Decision Systems*, Vol.22 No.2**,** pp.122-148.

Forman, E.H. and Gass, S.I. (2001), “The analytic hierarchy process: An exposition”, *Operations Research,* Vol. 49 No. 4 , pp. 469-486.

Golding, A.R. and Rosenbloom, P.S. (1996), "Improving accuracy by combining rule-based and case-based reasoning", *Artificial Intelligence*, Vol.97 Nos.1/2**,** pp.215-254.

Gray, A.E., Seidmann, A. and Stecke, K.E. (1993) “A synthesis of decision models for tool management in automated manufacturing”, *Management Science,* Vol. 39 No. 5, 549 - 567.

Hejazi, S.R., Doostparast, A. and Hosseini, S.M. (2011), “An improved fuzzy risk analysis based on a new similarity measures of generalized fuzzy numbers”, *Expert Systems with Applications*, Vol.38 No.8, pp. 9179–9185.

Ho, W. (2008), “Integrated analytic hierarchy process and its applications - A literature review”, *European Journal of Operational Research*, Vol.186 No.1, pp. 211-228.

Holsapple, C.W. and Whinston, A.B. (1996), *Decision Support Systems: A knowledge-based Approach*, West Publishing Company.

Ishizaka, A. and Labib, A. (2011), “Review of the main developments in the analytic hierarchy process”, *Expert Systems with Applications,* Vol. 38 No.11, pp. 14336–14345.

Kahraman, C. (2008), “Multi-criteria decision making methods and fuzzy sets”, in Kahraman, C. (Ed.), Springer, New York, NY, pp.1-18.

Kahraman, C. (2008), *Fuzzy Multi-criteria Decision Making: Theory and Applications with Recent developments*, Springer, New York, NY.

Kasie, F.M. and Bright, G. (2019), “Integrating fuzzy case-based reasoning and discreteevent simulation to develop a decision support system for part-fixture assignment and fixture flow control”, *Journal of Modelling in Management,* Vol.14 No.2, pp.312-338.

Kasie, F.M., Bright, G. and Walker, A. (2017a), “An intelligent decision support system for on-demand fixture retrieval, adaptation and manufacture”, *Journal of Manufacturing Technology Management,* Vol.28 No.2**,** pp.189-211.

Kasie, F.M., Bright, G. and Walker, A. (2017b), “Estimating cost of new products using fuzzy case-based reasoning and fuzzy analytic hierarchy process”, in Chen, C.-H., Trappey, A.C., Peruzzini, M., Stjepandić, J. and Wognum, N. (Eds), *Transdisciplinary Engineering: A paradigm Shift*, IOS Press, Nieuwe Hemweg, pp.969 - 976.

Kolodner, J.L. (1991), “Improving human decision making through case-based decision aiding”, *AI Magazine,* Vol. 12 No.2**,** pp.52–68.

Kolodner, J.L. (1992), “An introduction to case-based reasoning”, *Artificial Intelligence Review,* Vol.6 No.1, pp.3-34.

Kolodner, J.L. (1993), *Case-Based Reasoning*, Morgan Kaufmann Publishers, San Francisco, CA.

Kumar, K.A., Singh, Y. & Sanyal, S. (2009), “Hybrid approach using case-based reasoning and rule-based reasoning for domain independent clinical decision support in ICU”, *Expert Systems with Applications,* Vol. 36 No. 1, pp.65-71.

Kuo, T.C. (2010), “Combination of case-based reasoning and analytical hierarchy process for providing intelligent decision support for product recycling strategies”, *Expert Systems with Applications,* Vol. 37 No.8, pp. 5558-5563.

Lee, A.H.I., Chen, W.-C. and Chang, C.-J. (2008), “A fuzzy AHP and BSC approach for evaluating performance of IT department in the manufacturing industry in Taiwan”, *Expert Systems with Applications*, Vol.34 No.1, pp. 96-107.

Lee, J., Bagheri, B. and Kao, H.-A. (2015), “A Cyber-physical systems architecture for Industry 4.0-based manufacturing systems”, *Manufacturing Letters,* Vol.3 No.1**,** pp.18-23.

Lee, J., Davari, H., Singh, J. & Pandhare, V. (2018), "Industrial Artificial Intelligence for industry 4.0-based manufacturing systems", *Manufacturing Letters*, Vol. 18, No. 1, pp. 20-23.

Lee, J., Kao, H.-A. and Yang, S. (2014). "Service innovation and smart analytics for Industry 4.0 and big data Environment", *Procedia CIRP*, Vol.16 No.1**,** pp.3-8.

Li, S.-T. and Ho, H.-F. (2009), "Predicting financial activity with evolutionary fuzzy case-based reasoning", *Expert Systems with Applications*, Vol.36 No.1, pp.411-422.

Liao, T.W., Zhang, Z. and Mount, C.R. (1998), "Similarity measures for retrieval in case-based reasoning systems", *Applied Artificial Intelligence: An International Journal*, Vol.12 No.4, pp.267-288.

Lu, Y. (2017), "Industry 4.0: A survey on technologies, applications and open research issues", *Journal of Industrial Information Integration,* Vol.6 No.1**,** pp.1-10.

Mardani, A., Jusoh, A. and Zavadskas, E.K. (2015), "Fuzzy multiple criteria decision-making techniques and applications - Two decades review from 1994 to 2014", *Expert Systems with Applications,* Vol. 42 No. 8, pp. 4126-4148.

Meseguer, A. and Gonzalez, F. (2008), "A methodology for cutting-tool management through the integration of CAPP and scheduling", *International Journal of Production Research,* Vol. 46 No. 6, pp. 1685-1706.

Oh, K.J. and Kim, T.Y. (2007), "Financial market monitoring by case-based reasoning", *Expert Systems with Applications,* Vol.32 No.3, pp.789–800.

Özbayrak, M. and Bell, R. (2003), "A knowledge-based decision support system for the management of parts and tools in FMS", *Decision Support Systems*, Vol.35 No.4, pp.487-515.

Pal, S.K. and Shiu, S.C.K. (2004), *Foundations of Soft Case-Based Reasoning*, John Wiley & Sons, New Jersey, NJ.

Park, C.-S. and Han, I. (2002), "A case-based reasoning with the feature weights derived by analytic hierarchy process for bankruptcy prediction", *Expert Systems with Applications*, Vol.23 No.2, pp.255-264**.**.

Prentzas, J. and Hatzilygeroudis, I. (2007), "Categorizing approaches combining rule-based and case-based reasoning", *Expert Systems*, Vol.24 No.2, pp. 97-122.

Rahimifard, S. and Newman, S.T. (1997), "Simultaneous scheduling of workpieces, fixtures and cutting tools within flexible machining cells", *International Journal of Production Research*, Vol.35 No.9, pp.2379-2396.

Rahimifard, S. and Newman, S.T. (2000), "A reactive multi-flow approach to the planning and control of flexible machining facilities", *International Journal of Computer Integrated Manufacturing*, Vol.13 No.4, pp.311-323.

Ribeiro, R.A. (1996), "Fuzzy multiple attribute decision making: A review and new preference elicitation techniques", *Fuzzy Sets and Systems*, Vol.78 No.2, pp.155-181.

Rojko, A. (2017), "Industry 4.0 concept: Background and overview", *International Journal of Interactive Mobile Technologies (iJIM),* Vol.11 No.5**,** 77-90.

Saaty, T.L. (1994), "How to make a decision: The analytic hierarchy process", *Interfaces*, Vol.24 No.6, pp.19-43.

Saaty, T.L. (2003), "Decision-making with the AHP: Why is the principal eigenvector necessary", *European Journal of Operational Research*, Vol. 145 No. 1, pp. 85-91.

Schank, R. (1982), *Dynamic memory: A theory of reminding and learning in computers and people,* Cambridge University Press, New York, NY.

Slonima, T.Y. and Schneider, M. (2001), "Design issues in fuzzy case-based reasoning", *Fuzzy Sets and Systems*, Vol.117 No.3, pp.251-267

Tseng, F.S.C. and Chou, A.Y.H. (2006), "The concept of document warehousing for multi-dimensional modeling of textual-based business intelligence", *Decision Support Systems,* Vol.42 No.2**,** pp.727-744.

Vaidya, O.S. and Kumar, S. (2006), "Analytic hierarchy process: An overview of applications", *European Journal of Operational Research*, Vol. 169 No. 1, pp. 1-29.

Van Laarhoven, P.J.M. and Pedrycz, W. (1983), "A fuzzy extension of Saaty"s priority theory", *Fuzzy Sets and Systems*, Vol. 11 No. 3, pp. 229-241

Wang, S., Wan, J., Zhang, D., Li, D. and Zhang, C. (2016), "Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination", *Computer Networks,* Vol.101 No.1**,** pp.158-168.

Watson, I. (1999), "Case-based reasoning is a methodology not a technology", *Knowledge-Based Systems,* Vol. 12 No. 1**,** pp. 303-308.

Watson, I. and Marir, F. (1994), “Case based reasoning: A Review”, *Knowledge Engineering Review*, Vol.9 No.4, pp.327-354.

Weber, M. and Borcherding, K. (1993), “Behavioral influences on weight judgments in multiattribute decision making”, *European Journal of Operational Research,* Vol. 67 No. 1**,** pp. 1-12.

Wind, Y. and Saaty, T.L. (1980), “Marketing applications of the analytic hierarchy process”, *Management Science*, Vol. 26 No. 7**,** pp. 641-658.

Wu, M.-C., Lo, Y.-F. and Hsu, S.-H. (2008), “A fuzzy CBR technique for generating product ideas”, *Expert Systems with Applications*, Vol.34 No.1, pp.530-540.

Xu, L., Li, Z., Li, S. and Tang, F. (2007), “A decision support system for product design in concurrent engineering”, *Decision Support Systems*, Vol. 42 No. 4, pp. 2029-2042.

Zahedi, F. (1986), “The analytic hierarchy process: A survey of the method and its applications”, *Interfaces*, Vol. 16 No. 4, pp. 96-108.

Zedah, L.A. (1965), “Fuzzy sets”, *Information and control*, Vol. 8 No. 3, pp. 338-353.

Zhong, R.Y., Xu, X., Klotz, E. and Newman, S.T. (2017), "Intelligent manufacturing in the context of Industry 4.0: A review", *Engineering,* Vol. 3 No. 1**,** pp. 616-630.

Zimmermann, H.J. (2001), *Fuzzy Set Theory—and Its Applications*, Springer, New York, NY.