A WEB-BASED DECISION SUPPORT SYSTEM FOR AUTOMATED CUTTING TOOL SELECTION COMBINING CONSTRAINT SATISFACTION AND MULTI-CRITERIA DECISION-MAKING

OLEKSANDR YEVDOKYMOV[1], VITALII KOLESNYK[1], JOZEF PETERKA[2], STEPAN BODNAR[3]

[1]Sumy State University, Department of Manufacturing Technologies, Machines and Tools, Sumy, Ukraine

[2]Slovak University of Technology in Bratislava, Faculty of Materials Science and Technology, Trnava, Slovak Republic

[3]Sumy State University, Department of Information Technology, Sumy, Ukraine

DOI: 10.17973/MMSJ.2026_03_2025122

jozef.peterka@stuba.sk

This paper presents the development of a web application for automated cutting tool selection in machining, addressing the limitations of manual, experience-based choice that is time-consuming and difficult to standardize. The system is implemented as a three-tier client–server application, featuring a Java backend, a PostgreSQL database, and a React frontend. A proposed hybrid decision-making procedure that contains constraint satisfaction (CSP) is first used to filter feasible tool–holder–adapter sets, followed by multi-criteria decision-making (Analytic Hierarchy Process for criteria weighting, Weighted Sum Model and Technique for Order of Preference by Similarity to Ideal Solution for ranking) to select the most suitable alternative. The database combines Entity-Relationship and Entity-Attribute-Value models to store machines, parts, materials, and user-generated technical solutions, enabling the accumulation and reuse of experience. The approach is validated on a semi-finishing turning case of Inconel 718, where the system automatically ranks available sets and generates an order statement.

KEYWORDS

cutting tool, tool selection, web application, multi-criteria decision-making, analytic hierarchy process, inconel 718

1 INTRODUCTION

The selection of cutting tools plays a significant role in the machining process of parts and products [Navaneethan, 2024]. The correct selection of tools and equipment has a significant impact on the quality of finished parts, processing time, and the cost of the processing process [Zarkti, 2017]. The selection of cutting tools is a rather complex process, which is influenced by many factors, such as the experience of the employee (technological engineer, machine operator) performing the selection, machining method used [Pokorny, 2012], surface quality requirements [Peterka, 2004], the economic capabilities of the enterprise, the compliance of the equipment with certain requests and capabilities during processing [Navaneethan 2024, Oral 2004]. The rapid development of processing technologies has led to a significant increase in the number of manufacturers, suppliers and nomenclature of cutting tools. Catalogs of leading manufacturers of cutting tools (Walter, Sandvik Coromant, Iscar) contain thousands of different positions, which in turn significantly complicates the process of manual selection (especially using paper catalogs) [Xu 2011, Wang 2018]. This leads to an increase in time costs and a decrease in the efficiency of the selection process. Another significant problem is the lack of official representatives of manufacturers in certain countries' markets, resulting in the sale of their products through local distributors. This creates a substantial number of tool catalogs from local suppliers. Additionally, the influence of the human factor on the selection process can lead to errors and inconsistencies [Wu 2016]. To address these issues, automated methods for selecting cutting tools and equipment are being increasingly employed, utilizing various algorithms and artificial intelligence techniques. A significant advantage of automated systems is the reduction of time required for the selection process, the possibility of 24/7 use, and the reduction of the impact of human errors [Leo Kumar 2019]. Currently, methods based on knowledge bases [Arezoo 2000, Prasad 2016, Alberti 2011] and geometric feature matching [Zhou 2019] are actively used in cutting tool selection applications. The disadvantage of using knowledge bases is the need for constant manual updates, which can result in non-relevant compliance with modern requests. GFM methods, in turn, are based on recognizing the geometry of the part and require accurate input geometric data, which is particularly relevant for complex-profile parts, but does not fully account for economic factors. At the same time, multi-criteria decision-making (MCDM) methods are quite promising for use in automated selection applications [Navaneethan 2024]. The use of MCDM methods (AHP, WSM, TOPSIS) enables the consideration of various criteria and converts them into a quantitative rating of alternatives [Taherdoost 2023].

In modern industry, the concept of cloud manufacturing is increasingly being implemented. This, in turn, creates a demand for integrating decision-making algorithms into accessible and convenient software interfaces. The transition from locally installed software to web applications enables the tool selection process to be performed on any platform with Internet access at any time [Yevdokymov 2025].

Thus, this article aims to expand the methodological foundation of methods for implementing multi-criteria decision-making in the automated cutting tool selection process. The implementation of MCDM methods in web applications remains relevant, considering the needs of modern production and the concept of Industry 4.0. It is assumed that the combination of the hybrid decision-making procedure and multi-criteria optimization methods based on AHP (Analytic Hierarchy Process) for criteria weighting, WSM (Weighted Sum Model) and TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) for ranking will reduce the shortcomings of existing solutions related to the influence of the human factor on the process of selecting a cutting tool for machining processes for turning and milling operation.

2 METHODS

To implement the web application for cutting tool selection, a three-level client-server architecture is used. Java is used as the server programming language (backend of the web application), enabling the efficient processing of complex calculation algorithms and operates reliably with arrays of typed data. It is proposed to use SQL to store data about users, tools, machines, and parts. SQL enables the effective structuring of parametric data. React is used to implement the graphical interface of the application (client part). React provides a convenient client-side interface for construction. Development environments, categorized by programming languages and architecture components used in the application development process, are summarized in Tab. 1.

Table 1. Implementation tools

Architecture component	Programming language	Development environment
Backend	Java	Intellij Idea 2525.2
Frontend	React	VS Code 1.104
Database	SQL	PostgreSQL 17.6

To implement the tool selection process in the application, a two-stage procedure is proposed. At the first stage, the method of satisfying the constraints of the CSP (Constraint Satisfaction Problem) is used. The CSP method allows filtering the set of solutions by excluding options that do not meet the technological conditions of the processing operation. At the next stage, multi-criteria decision-making (MCDM) methods are used to rank the filtered set of alternatives. It is proposed to use combinations of AHP (Analytic Hierarchy Process) methods to determine the weights of the ranking criteria, and WSM (Weighted Sum Model) and TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) to rank sets according to the specified parameters. The proposed algorithm scheme (Fig. 1) is described in a previous study [Yevdokymov 2025].

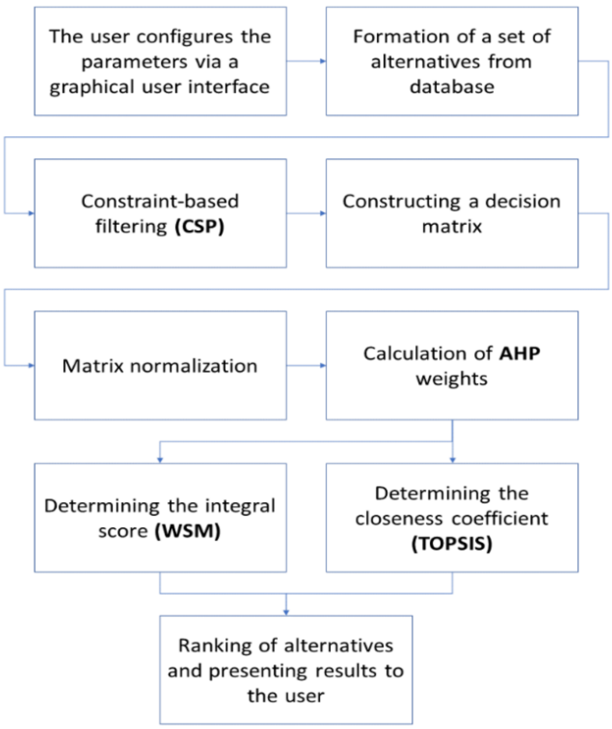

Figure 1. Scheme of application selection algorithms

To implement the program algorithms, at the first stage it is necessary to form a set for further ranking and filtering:

$$S = \{s_1, s_2, \dots, s_n\} \quad (1)$$

where S – the set of combinations of tool equipment.

$$S = \{s_i = (T_i, H_i, A_i)\} \quad (2)$$

where T_i – cutting tool, H_i – holder or shank, A_i – adapter (if required).

Then, filtering is performed through user interaction with the graphical interface. The user specifies the material group, processing type, hierarchical classification of processing type and cooling. The filtering formula has the following form:

$$F = \{s_i = (T_i, H_i, A_i) \in S \mid f_{cst}(s_i; M.P, O, G, C) = 1\} \quad (3)$$

where F – is the admissible (filtered) set, S – is the set of all possible combinations, $f_{constraints}$ – is a boolean (1 or 0) function indicating whether the set meets the technical requirements, M – is the material group according to ISO 513 [International Organization for Standardization 2012a], P – is the machining type, O – is the hierarchical classification of the machining type, G – is the machine type, C is the cooling type.

After forming the set of filtered alternatives, the system proceeds to ranking the selection results using the MCDM method [Yevdokymov 2025]. For that purpose each alternative is represented as a vector of values:

$$d_i = (d_{i1}, d_{i2}, \dots d_{im}) \quad (4)$$

where $d_{ij} \in \mathbb{R}$ – is the value of the alternative s_i for criterion C_j; m – is the number of criteria.

A vector of indicators is formed for each alternative. Next, all vectors d_{ij} are combined into a decision matrix $D \in \mathbb{R}^{n \times m}$:

$$D = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1m} \\ d_{21} & d_{22} & \dots & d_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \dots & d_{nm} \end{bmatrix} \quad (5)$$

where n — is the number of alternatives in the set F.

To normalize the decision matrix, the application uses the normalization criteria specified in Tab. 2.

Table 2. Normalization Criteria [14]

Criterion	Notation	Type of Criterion
Price	C_1	Minimize
Availability	C_2	Maximize
Rating	C_3	Maximize
Experience	C_4	Maximize

The decision matrix is normalized to bring different criteria onto a standard scale of [0, 1], enabling comparison. The resulting matrix is the normalized decision matrix:

$$R = [r_{ij}] \in [0,1]^{n \times m} \quad (6)$$

For each criterion, if higher values are better:

$$r_{ij} = \frac{d_{ij} - \min d_{ij}}{\max d_{ij} - \min d_{ij}} \quad (7)$$

If lower values are better:

$$r_{ij} = \frac{\max d_{ij} - d_{ij}}{\max d_{ij} - \min d_{ij}} \quad (8)$$

First, it is necessary to construct a matrix of pairwise comparisons. For example, if we have n criteria $C_1, C_2, \dots, C_n$ we construct a square matrix $A \in \mathbb{R}^{n \times m}$, where each element a_{jk} means how much criterion j critical than criterion k:

$$A = [a_{jk}] \quad (9)$$

where a_{jk} – relative preference of criterion j over k.

The pairwise comparison matrix proposed for use in the application is given in Tab. 3.

Table 3. Pairwise Comparison Matrix

	Price C1	Avail. C2	Rating C3	Exp. C4
Price	1	1/3	1/7	1/5
Avail.	3	1	1/5	1/3
Rating	7	5	1	3
Exp.	5	3	1/5	1

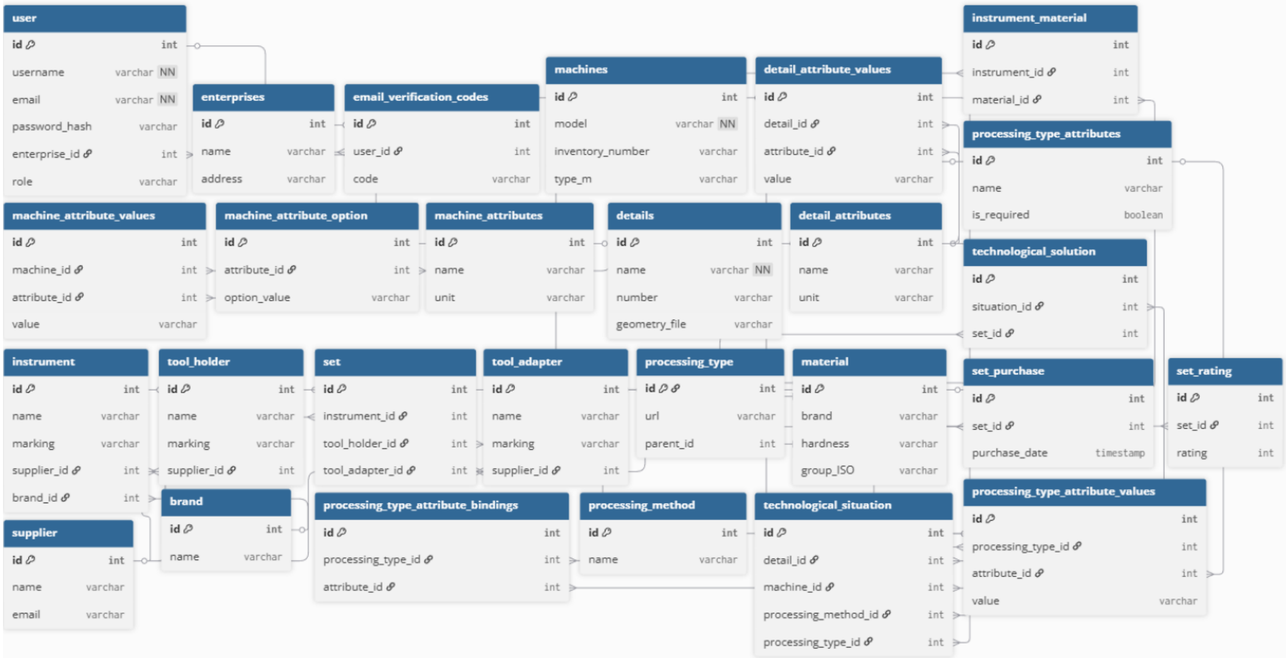

Figure 2. Structure of the application database

The next step is to perform column normalization, which involves scaling the elements to a scale that allows row comparisons. Each component of the matrix a_{jk} is normalized by column k:

$$\widetilde{a_{jk}} = \frac{a_{jk}}{\sum_{i=1}^{n} a_{jk}} \tag{10}$$

This means that each element is divided by the sum of the corresponding column, i.e., all columns will have a sum = 1. As a result the normalized matrix Ã will be obtained. On this basis the arithmetic mean of each row of the normalized matrix can be calculated:

$$w_j = \frac{1}{n}\sum_{k=1}^{n} \widetilde{a_{jk}},\ w = (w_1, \dots, w_n) \tag{11}$$

The weight vector $w = (w_1, \dots, w_n)$ describes the relative importance of each criterion, derived from the expert assessments**.** The consistency check is performed according to equation:

$$(Aw)_j = \sum_{k=1}^{n} a_{jk} \cdot w_k \tag{12}$$

The calculation of the eigenvalue λ_{max}:

$$\lambda_{max} = \frac{1}{n}\sum_{j=1}^{n} \frac{(Aw)_j}{w_j} \tag{13}$$

The consistency index is calculated:

$$CI = \frac{\lambda_{max} - n}{n-1} \tag{14}$$

Consistency ratio :

$$CR = \frac{CI}{RI} \tag{15}$$

After determining the weights, the system proceeds to evaluate alternatives using the WSM and TOPSIS methods. The WSM method, the decision matrix $R = [r_{ij}] \in [0,1]^{n\times m}$, and the weight vector $w = (w_1, \dots, w_n)$ determined by the AHP method are used for calculations. The integral assessment of each alternative s_i is calculated according to the equation:

$$U_i = \sum_{j=1}^{m} w_j \cdot r_{ij} \tag{16}$$

Or in vector form:

$$U_i = \vec{r_i} \cdot \vec{w} \tag{17}$$

Where $\vec{r_i} = (r_{i1}, r_{i2}, \dots, r_{im})$ – normalized vector of alternative s_i; $\vec{w}$ – vector of criteria weights; $U_i \in [0,1]$ – integral alternative score.

The obtained value of the integral score U_i always belongs to the interval [0,1], since both the normalized values w_j and the weights lie within [0,1], while and the sum of the weights is 1. The value of U_i reflects the overall utility or compliance of the alternative with all criteria, taking into account their relative weights. To empower calculation using the TOPSIS method, it is necessary to have a decision matrix $R = [r_{ij}] \in [0,1]^{n\times m}$ and a weight vector $w = (w_1, \dots, w_n)$ determined by the AHP method. The TOPSIS method consists of five main steps:

1. First, the normalized decision matrix is constructed:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}} \tag{18}$$

2. Next, the weighted normalized matrix is formed:

$$v_{ij} = w_j \cdot r_{ij} \tag{19}$$

3. The ideal and anti-ideal points are then determined:

$$A^+ = \left\{\max_i v_{ij}\right\}; \qquad A^- = \left\{\min_i v_{ij}\right\} \tag{20}$$

4. Subsequently, the distances to the ideal and anti-ideal solutions are calculated:

$$D_i^+ = \sqrt{\sum_{j=1}^{4} (v_{ij} - A_j^+)^2}, \quad D_i^- = \sqrt{\sum_{j=1}^{4} (v_{i=j} - A_j^-)^2} \tag{21}$$

5. The index of proximity to ideals is calculated:

$$C_i = \frac{D_i^-}{D_i^- + D_i^+}, \quad C_i \in [0,1] \tag{22}$$

To populate the application database, user experience is continuously collected, processed, and stored. The application connects users, including technologists and warehouse personnel, within small, medium, and large enterprises, as well as individual users, who can access an anonymized, cumulative database of technical solutions (without links to specific enterprises, orders, or parts). This shared knowledge base enables users and companies to exchange experiences and consider them in their own decision-making processes. By leveraging the application's algorithms and user interfaces, it becomes possible to reduce the time required for cutting tool selection, lower tool procurement costs, and more effectively manage technological processes, machine fleets, orders, and tools.

3 RESULTS

3.1 Application database

The database plays an important role in the application's core. It is used to accumulate user experience, which is critical for the application selection algorithms. In addition to technical solutions, the database stores data about parts, machines, technological transitions, user data, and attributes. The database is designed using ER (Entity-Relationship) modeling. For technical characteristics of machines and parts, where the structure of parameters can change, the EAV (Entity-Attribute-Value) pattern is used, allowing the system to be scaled when adding new types of machine and part parameters (Fig 2).

User-related information is stored in the user's entity, while the enterprise's entity contains data on the companies to which these users are affiliated. Authentication support is provided by the *email_verification_codes* entity, which stores verification codes used for logging into the application (Fig 2).

The machines entity stores information about the machine tools, and the details entity contains data on parts, including their type, geometry, and key characteristics. To ensure flexibility in describing technical parameters, an extended attribute-based approach is applied. Attribute definitions are stored in the *machine_attributes* and *detail_attributes* tables, their specific values in *machine_attribute_values* and *detail_attribute_values*, and additional selectable options in auxiliary directories such as *machine_attribute_option* (Fig 2).

```
public List<TechnologicalSolutionDTO> getSetByProcessingTypeIdAndProcessingMethodIdAndMaterialIdAndMachineTypeAndCoolingMethodIsPresent(

        Long processingTypeId, Long processingMethodId,
        Long materialId,
        Long machineId, Long coolingMethodId)
// getting the set by parameters
{
    List<TechnologicalSituation> situations = techSitRepo.findTopSituations(processingMethodId, processingTypeId, materialId, machineId, coolingMethodId);
    if (situations.isEmpty()) {
        throw new RuntimeException("Situations not found");
    }

    List<SetEntity> sets = situations.stream().map(TechnologicalSituation::getSet).toList();
    ToolSBA_MCDC mcdc = new ToolSBA_MCDC(sets);
```

Figure 3. CSP method in the application code

Data on the cutting tool are stored in the instrument entity, whereas *tool_holder* and *tool_adapter* describe tool holders and adapters, respectively. The materials of parts and their properties, as well as the materials used for the tools, are maintained in the material and *instrument_material* tables. Information on cutting tool manufacturers and local suppliers is consolidated in the brand and supplier entities (Fig 2).

The *processing_type* entity represents technological operations, and the corresponding processing methods are described in the *processing_method* entity. Process parameterization is implemented through the attribute system, *processing_type_attributes*, and their corresponding values, *processing_type_attribute_values*, with the linkage mechanism defined in *processing_type_attribute_bindings* (Fig 2).

The outcome of the selection algorithms is the formation of a tool set (set), which includes a tool, a holder, and an adapter. Each set is associated with specific application conditions recorded in *the technological situation and formalized as a technological solution*. The system additionally supports feedback and logistics functionalities through the entities *set_rating* (evaluation of set effectiveness) and *set_purchase* (purchase-related information) (Fig 2).

3.2 Implementation of algorithms

In the first stage, the set from the application database is filtered using the CSP method, which was implemented in the application code (Fig 3).

According to the code, the system filters a set of alternative sets from the application database to select only the result suitable for the technological transition. Filtering criteria: *"processingTypeId"* – hierarchical classification of the transition according to graphic thumbnails, *"processingMethodId"* – type of processing in the transition (rough, semi-finishing, finishing), *"materialId"* – material group according to ISO 513, *"machineId"* – machine type, *"coolingmethodId"* – cooling type. After that, the system proceeds to rank the selection results using MCDM methods.

First, a matrix of pairwise comparisons is constructed using the AHP method (Tab. 3). The authors propose the following ranking criteria: price, availability, rating, and number of orders for the set. The ranking criteria were considered and implemented in the AHP method application code (Fig. 4).

```
static final double[][] AHPMatrix = new double[][]{
        {1.0, 1.0/3.0, 1.0/7.0, 1.0/5.0},
        {3.0, 1.0, 1.0/5.0, 1.0/3.0},
        {7.0, 5.0, 1.0, 3.0 },
        {5.0, 3.0, 1.0/3.0, 1.0 }
};
static final int SCALE_FOUR = 4;
static final int SCALE_THREE = 3;
static final int SCALE_TWO = 2;

public static final Criterion PRICE      = new Criterion( key: "price",      Direction.COST);
public static final Criterion STOCK      = new Criterion( key: "stock",      Direction.BENEFIT);
public static final Criterion RATING     = new Criterion( key: "rating",     Direction.BENEFIT);
public static final Criterion EXPERIENCE = new Criterion( key: "experience", Direction.BENEFIT);

static final List<Criterion> CRITERIAS = List.of(PRICE, STOCK, RATING, EXPERIENCE);

static final Map<String, ToDoubleFunction<SetEntity>> EXTRACTORS = Map.of(
        k1: "price",     SetEntity s -> ToolSBA_MCDC.toDouble(s.getPrice(), defaultValue: 0.0),
        k2: "stock",      SetEntity s -> s.getInStock() ? 1.0 : 0.0,
        k3: "rating", SetEntity::getRating,
        k4: "experience", SetEntity s ->
        {
            var q = s.getQuantityScore();
            return q != null ? q.doubleValue() : 0.0;
```

Figure 4. Code of the implementation of the AHP method

The "AHPMatrix" function determines the weights of the criteria of the filtered sets according to the described values (Table 3). Where "Criterion "price", Direction.COST" – criterion "price", minimization; "Criterion "stock", Direction.BENEFIT" – criterion "availability", maximization; "Criterion "rating", Direction.BENEFIT" – criterion "rating", maximization; "Criterion "experience", *"Direction.BENEFIT"* – criterion "number of orders", maximization;

After calculating the AHP pairwise comparison matrix, the calculation is performed using the WSM and TOPSIS methods. The Fig. 5 shows the implementation of the WSM method.

```java
public static void computeTOPSIS()  1 usage
{
    final int n = weightedNormalizedMatrix.length;
    final int m = weightedNormalizedMatrix[0].length;
    double[] antiIdealVector = new double[m];
    double[] idealVector = new double[m];
    double[] idealDistance = new double[n];
    double[] antiIdealDistance = new double[n];

    // 3) Find anti- and ideal points (arrays modify in the method)
    findIdealAndAntiIdealPointVectors(idealVector, antiIdealVector);

    // 4) Calculate distances to ideals (arrays modify in the method)
    calculateDistanceToIdeals(idealDistance, antiIdealDistance, DoubleBuffer.wrap(idealVector).asReadOnlyBuffer() ,
            DoubleBuffer.wrap(antiIdealVector).asReadOnlyBuffer());

    // 5) Calculate proximity index and get maps for  each set with its proximity index
    calculateProximityCoeff(DoubleBuffer.wrap(idealDistance).asReadOnlyBuffer(),
            DoubleBuffer.wrap(antiIdealDistance).asReadOnlyBuffer(), alts);

}
```

Figure 5. Implementation of the WSM method

```java
public static void computeWSM()  1 usage
{
    final int n = normalizedWSMSolutionMatrix.length;
    final int m = normalizedWSMSolutionMatrix[0].length;
    double sum = 0.0;
    double integralRate = 0.0;
    Map<Alternative, Double> score = new LinkedHashMap<>();
    if(n != alts.size())
        throw new IllegalArgumentException("normalizedMatrix HAS NOT the SAME size with ALTS");
    AHPResult ahpResult = AHP.checkConsistency();
    if(!ahpResult.isConsistent)
        throw new IllegalArgumentException("AHP IS NOT consistent");
    for(int i = 0; i < n; ++i)
    {
        for(int j = 0; j < m; ++j)
            sum += Data.weightVector[j] * normalizedWSMSolutionMatrix[i][j];

        integralRate = round(sum, ConstData.SCALE_FOUR, RoundingMode.HALF_UP);

        if(integralRate < 1e-9 || integralRate > 1e9)
            throw new IllegalArgumentException("Integral rate must be within 0 and 1");

        Alternative alt = alts.get(i);
        score.put(alt, integralRate);

        sum = 0;
    }
    wsmScore = score;
}
```

Figure 6. TOPSIS method implementation

The function *"AHP.checkConsistency*()" – checks the consistency of the AHP pairwise comparison matrix for further use in the WSM and TOPSIS methods. "sum += Data.weihgtVector [j] * normalizedWSMSolutionMatrix [i] [j];" calculates the integral value of the alternative. *"ConstData.SCALE_FOUR"* – the accuracy parameter that sets the rounding of the results, in this case to 4 decimal places. *"integralRate"* – the calculated integral value in the range from 0 to 1.

Fig. 6 presents the implementation of the TOPSIS closeness coefficient to the ideal solution in the application code.

The *findIdealAndAntiIdealPointVectors(IdealVector, antiIdealVector)* function determines the coordinates of the positive and negative ideal solutions by searching for the minimum and maximum values in the decision matrix. The *calculateDistanceToIdeals(idealDistance, antiIdealDistance, DoubleBuffer.wrap(idealVector)*.asReadOnlyBuffer(), DoubleBuffer.wrap(antiIdealVector).as ReadOnlyBuffer()) function computes the distance of each alternative from the positive and negative ideal points. The calculateProximityCoeff (DoubleBuffer.wrap(idealDistance).asReadOnlyBuffer(), DoubleBuffer.wrap(antiIdealDistance).asReadOnlyBuffer(), alts) function then calculates the closeness coefficient to the ideal solution for each alternative. Based on these coefficients, the results are ranked, and the user is presented with a rating and a list of selected tool sets in the application's graphical user interface (frontend). To validate the results, select the tool in the application using the graphical interface. For greater clarity, this algorithm is illustrated in the graphical interface (Fig. 7).

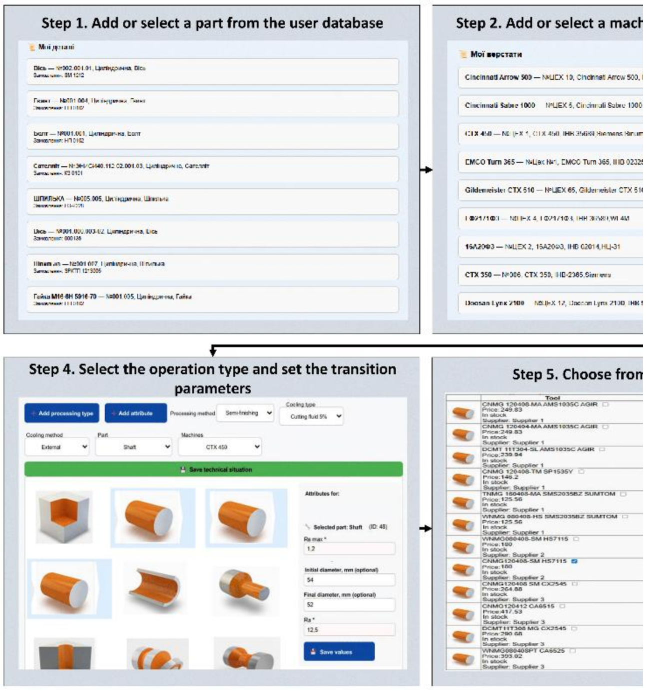

Figure 7. Selection algorithm in the application

To perform tool selection, the user first specifies the part parameters, including its type, dimensions, and order number. The defined part is then saved and becomes available in the "Part" tab as an element of the parts library. If the user is an employee of an enterprise, a local part database can be formed, allowing other users within the same enterprise to access shared parts as needed (step 1).

Each user maintains an individual machine tool library. For enterprise users, it is additionally possible to create a common machine database accessible to all employees of that enterprise. Machines are added via the graphical interface. When registering a machine, the user first specifies the machine type (turning or milling). For a turning machine, the user is prompted to enter or select the following data: inventory number, workshop (shop) number, model, and CNC system. Subsequently, technical characteristics are specified, including mounting type, holder type, maximum power, maximum spindle speed, tool type, and the presence of a sub-spindle and driven tooling. The machine is then saved to the user's machine library (step 2). For a milling machine, the user similarly specifies the inventory number, workshop number, model, and CNC system. The following technical characteristics are then entered: maximum spindle speed, maximum power, holder standard, holder type, pull-stud type, and the number of machine axes. After completing these fields, the machine is saved to the user's library.

Materials in the database are organized into groups according to the ISO 513 [International Organization for Standardization 2012a] standard. Because the workpiece material critically influences both the machining process and tool selection, it plays a central role in the CSP-based filtering procedure. Material groups are used as a key filtering attribute. The following ISO 513 groups are adopted: P – steels (carbon and alloy), M – stainless steels (austenitic, martensitic, ferritic), K – cast irons (grey, malleable, high-strength, hardened), N – non-ferrous metals (aluminium, copper, bronze, brass, magnesium), S – heat-resistant alloys (nickel- and cobalt-based) and titanium, and H – hardened materials (above 45–50 HRC).

Within the application, the user selects the part material from an interface that already contains more than 160 materials based on standards such as DSTU, AISI, SAE, and ISO. If the selected material is present in the database, its group and hardness are automatically populated from the stored properties. The user may also add a new material via the dedicated interface by specifying the material name, hardness limits, and ISO material group (step 3).

Once data on parts, machines, and material have been defined, the user proceeds to the "technical situation" stage. Here, the machining method (roughing, semi-finishing, finishing) and cooling type (5% cutting fluid, 10% cutting fluid, oil, air, or dry machining) are specified. The cooling supply mode (external or internal) is also selected. The user then chooses a part from the parts library and a machine from the machine library for the given operation step. Using graphical thumbnails, the user selects the technological route for the operation and defines the geometric machining parameters, which depend on the chosen machining method. The specified values are saved, and the user may either continue adding further technological transitions or request a technical solution for the current transition (step 4).

In the *"technical solutions"* stage, the user can either retrieve existing solutions from the accumulated database of user-generated solutions or add a new solution to this shared repository. When adding a new tool set, the user defines a custom combination of tools for the selected transition. A link to the product page on the supplier's website is provided, enabling the application to automatically extract available data (item number, designation, supplier). The tool name must be chosen from a predefined list of typical names, and the tool material is specified.

For a tool holder, the following information is entered: name (e.g., turning tool holder, boring bar holder, parting tool holder), holder designation (ISO 5608 [International Organization for Standardization 2012b], e.g., PCLNL2525M12, MCLNL2020K12), item number on the supplier's website, link to the product page, supplier (selected from a list or added by the user if not present), and brand (manufacturer). If the brand is absent from the list, it can also be added via the corresponding interface. For the cutting tool, the user specifies: name (e.g., carbide insert, threading insert, parting insert), tool designation (for inserts, ISO 1832 [International Organization for Standardization 2017] code such as CNMG120408), item number, link to the supplier's website, cutting tool material (e.g., cemented carbide, high-speed steel), supplier, and manufacturer. If required by the tool design, an adapter can also be specified.

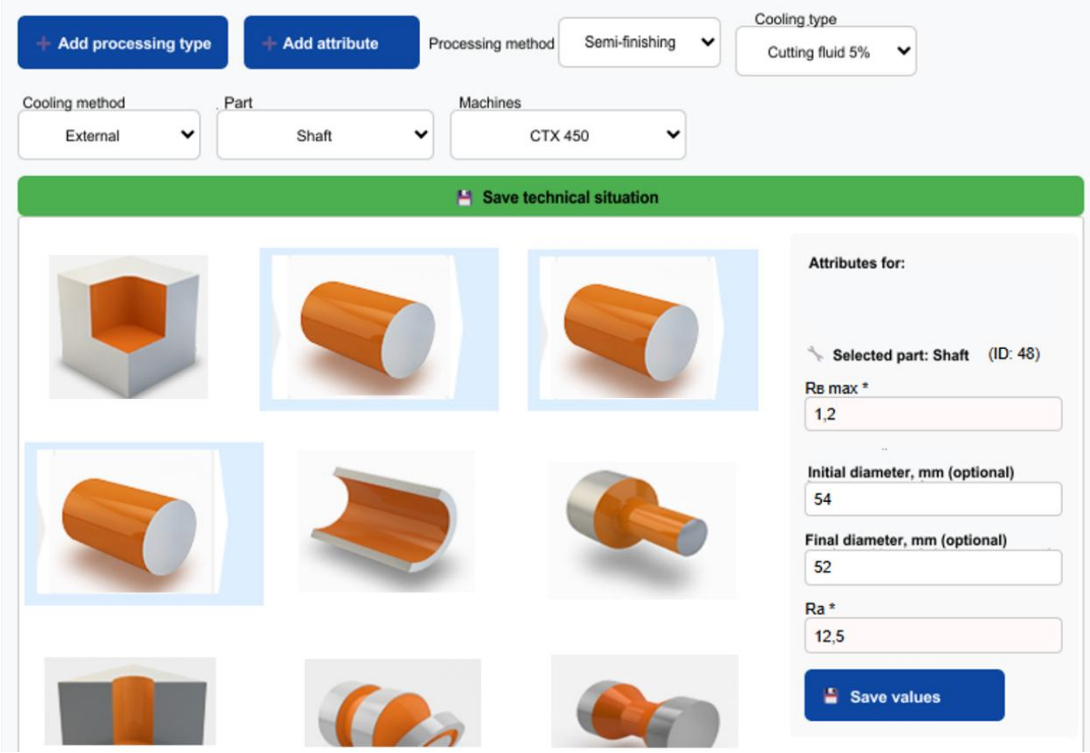

Figure 8. The “technical solutions” tab of the application with input data for experimental selection

After filtering and ranking, the system generates a technical solution in the form of a table containing recommended tool sets for machining the part at the given transition or set of transitions (step 5). The user then selects the most appropriate option or combines several options to generate an order form. This form may be printed, saved as a PDF, or sent directly to the supplier through the application interface to place a tool order (step 6).

The experimental demonstration of the selection procedure is conducted for a single turning pass. The input data are as follows: workpiece material – Inconel 718 (ISO material group S: heat-resistant alloys), machining type – semi-finishing, machine – CTX 450, machining diameter – 52 mm, and cooling – cutting fluid at 5%. During calculation, price and availability correspond to the values specified on the supplier’s website at the time of computation. The input data for this transition are shown in the graphical interface (the “technical situation” tab) (Fig. 8).

Set Id	ToolHolder marking	ToolInstrument marking	ToolAdapter marking	WSM	TOPSIS	Combined
107	PCLNL 2525 M12C	CNMG 120408-MA AMS1035C AGIR		0,13	0,71	0,42
108	DCLNL-2525-M12	CNMG 120404-MA AMS1035C AGIR		0,13	0,72	0,43
109	SDJCL-2020-K11	DCMT 11T304-SL AMS1035C AGIR		0,15	0,79	0,47
110	PCLNL 2020 K12C	CNMG 120408-TM SP1535Y		0,01	0,04	0,03
111	MTJNL-2525-M16	TNMG 160408-MA SMS2035BZ SUMTOM		0,13	0,73	0,43
112	MWLNL-2020-K08	WNMG 080408-HS SMS2035BZ SUMTOM		0,13	0,74	0,44
113	MWLNL2525M08	WNMG080408-SM HS7115		0,18	0,99	0,58
114	MCLNL2020K12	CNMG120408-SM HS7115		0,18	1,00	0,59
119	DCLNL-2525M12	CNMG120408 SM CX2545		0,15	0,80	0,47
120	PCLNL 2525 M12C	CNMG120412 CA6515		0,12	0,69	0,41
121	SDACL-2020K11	DCMT11T308 MG CX2545		0,03	0,19	0,11
122	PWLNL 2020 K08C	WNMG080408PT CA6525		0,12	0,70	0,41

Figure 9. Internal calculation results inside the web application

To perform the selection after entering the data, click "Save Technical Situation" and go to the "Technical Solutions" tab. Then select the part in the table for which the selection and the necessary technological transition were performed. The system displays a ranked result with a rating in accordance with the previously described application algorithms. The Fig. 9 shows the backend calculation results for a given operation.

Where “Set Id” – is the set number in the web application database. “ToolHolder Marking” – marking of the holder of the set for which the calculation is performed, “ToolInstrument marking” – marking of the insert of the set for which the calculation is performed, “ToolAdapter marking” – marking of the adapter of the set for which the calculation is performed (if required). “WSM” – results of the calculation by the weighted sum method. “TOPSIS” – results of the calculation by the proximity method. “Combined” – the total WSM and TOPSIS rating for the set, which is displayed to the user. Fig. 10 shows the Selection Results in the application user interface.

Where *“Tool”* – marking of the insert of the set for which the calculation is performed, *“Holder”* – marking of the holder of the set for which the calculation is performed, *“Adapter”* – marking of the adapter of the set for which the calculation is performed (if necessary). *“Rating”* – overall WSM and TOPSIS rating for the set, which is displayed to the user (Combined) *“Price”* – cost per unit of the product in local currency, *"In stock/Not available"* – availability of the set item on the supplier's website, "Supplier" - supplier name (names changed).

The calculation data were taken into account by considering the integral assessment and the proximity coefficient for each set (Fig. 11).

According to the application’s selection algorithms, the best result for this turning operation on a part made of Inconel 718 is the tool set with ID 114 from Supplier 2. This set consists of the holder MCLNL2020K12 and the insert CNMG120408-SM HS7115, achieving an integral score of C_i = 0.18, a closeness coefficient to the ideal solution of U_i = 1, and an overall rating of

0.59 (Fig. 11). If the user is satisfied with one of the proposed sets, they can generate an order list directly from the interface. Alternatively, the user may form a custom combination by selecting tools and holders from different sets, or by choosing only individual components (tool, holder, adapter) when some elements are already available (for example, when a suitable holder and/or adapter is in stock and only cutting inserts are required).

	Tool	Holder	Adapter	Rating
	CNMG 120408-MA AMS1035C AGIR ☐ Price:249.83 In stock Supplier: Supplier 1	PCLNL 2525 M12C ☐ Price:2993.23 In stock Supplier: Supplier 1		0.416
	CNMG 120404-MA AMS1035C AGIR ☐ Price:249.83 In stock Supplier: Supplier 1	DCLNL-2525-M12 ☐ Price:2915.83 In stock Supplier: Supplier 1		0.427
	DCMT 11T304-SL AMS1035C AGIR ☐ Price:239.94 In stock Supplier: Supplier 1	SDJCL-2020-K11 ☐ Price:2163.33 In stock Supplier: Supplier 1		0.469
	CNMG 120408-TM SP1535Y ☐ Price:146.2 In stock Supplier: Supplier 1	PCLNL 2020 K12C ☐ Price:2951.52 Not available Supplier: Supplier 1		0.025
	TNMG 160408-MA SMS2035BZ SUMTOM ☐ Price:125.56 In stock Supplier: Supplier 1	MTJNL-2525-M16 ☐ Price:2915.83 In stock Supplier: Supplier 1		0.431
	WNMG 080408-HS SMS2035BZ SUMTOM ☐ Price:125.56 In stock Supplier: Supplier 1	MWLNL-2020-K08 ☐ Price:2822.09 In stock Supplier: Supplier 1		0.436
	WNMG080408-SM HS7115 ☐ Price:180 In stock Supplier: Supplier 2	MWLNL2525M08 ☐ Price:600 In stock Supplier: Supplier 2		0.581
	CNMG120408-SM HS7115 ☑ Price:180 In stock Supplier: Supplier 2	MCLNL2020K12 ☑ Price:500 In stock Supplier: Supplier 2		0.590
	CNMG120408 SM CX2545 ☐ Price:264.88 In stock Supplier: Supplier 3	DCLNL-2525M12 ☐ Price:2074.32 In stock Supplier: Supplier 3		0.471
	CNMG120412 CA6515 ☐ Price:417.53 In stock Supplier: Supplier 3	PCLNL 2525 M12C ☐ Price:3087.4 In stock Supplier: Supplier 3		0.407
	DCMT11T308 MG CX2545 ☐ Price:290.68 In stock Supplier: Supplier 3	SDACL-2020K11 ☐ Price:1671.84 Not available Supplier: Supplier 3		0.111
	WNMG080408PT CA6525 ☐ Price:393.02 In stock Supplier: Supplier 3	PWLNL 2020 K08C ☐ Price:3044.8 In stock Supplier: Supplier 3		0.411

Figure 10. Selection results in the application user interface (frontend)

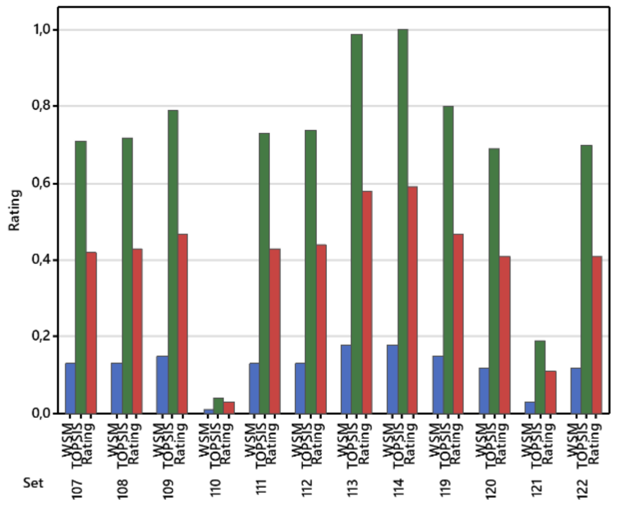

Figure 11. Set rating

For experimental validation, the set with the highest rating is selected and an order list is created. Supplier data already stored in the database are filled in automatically, whereas customer data are specified during registration; however, details such as the delivery address can be updated at the ordering stage. After completing these steps, the user can either print the generated order document or submit it electronically to the supplier via the application interface (Fig. 12).

Where *"Supplier"* – supplier's data and details. If the supplier is in the application database, the data is automatically pulled into the statement. *"Customer"* – customer's details and data are specified (company name, address, tax number, responsible person), "Product name" – standardized product name. *"Marking"* – standardized marking of the tool, *"Article"* – product article number on the supplier's website. *"Quantity"* – field for changing the number of units of the product being ordered.

Supplier		Customer	
Official Supplier Name	Supplier 2	Company name:	LLC "Workshop"
EDRPOU Code	12311158	Address	40000, Ukraine, Sumy Oblast, Sumy, Soborna St., 32
Address	79005, Ukraine, Lviv Oblast, Lviv, Ivana Franka St., 25	EDRPOU/TIN (Tax ID)	32111165
Tel.	+38011122258	Full Name	Petrov Petro Petrovych
Email	Supplier@gmail.com	Tel.	+38022258865
		Email	Workshop@gmail.com

№ п/п	Product name	Marking	Article	Unit	Quantity	Price, UAH	Amount, UAH
1	Turning tool holder	MCLNL2020K12	MCLNL2020K12	pcs	1	500	500
2	Carbide insert	CNMG120408-SM HS7115	CNMG120408-SM HS7115	pcs	1	180	180

Total excluding VAT: **680.00 UAH**

VAT 20%: **136.00 UAH**

Total including VAT: **816.00 UAH**

Amount in words: **eight hundred sixteen hryvnias zero kopiykas**

Approved by / **Petrov Petro Petrovych**

Date: **29.11.2025**

Download PDF | Print

Back | Send email | Next

Figure 12. Order statement

"*Price, UAH"* – cost per unit of the product in Ukrainian hryvnias. "Amount, UAH" – calculated amount by item (price and quantity). "Total including VAT" – final amount of the order to be paid including VAT. "Amount in words" – text display of the full order amount in words "Approved by" – name of the person who approved the statement. "Date" – date of invoice generation. "Send email" – button for sending the generated document to the supplier's email. "Download PDF" – button to download order information in PDF format.

4 DISCUSSION

The difference of the proposed application from similar solutions based on MCDM [Wang 2016, Phung 2019] is the combination of the CSP (constraint satisfaction logic) with a set of MCDM methods (AHP, WSM, TOPSIS). This, in turn, allows for the creation of a more flexible selection algorithm. In addition, most of the automated selection applications described in the studies are implemented as desktop [Wang 2016, Phung 2019, Duan 2021, Zhou 2019] «offline» applications or parts of CAM/CAD systems [Zarkti 2017, Ociepka 2015, Zhao 2002] while the proposed by the authors in this study offers an "online" web application, which can be accessed via a phone or computer using a browser at anytime from anywhere in the world. Compared to commercial systems of cutting tool manufacturers, such as Sandvik Coromant CoroPlus® Tool Guide [Sandvik Coromant 2025], ISCAR® Tool Advisor [Iscar 2025], Walter® GPS [Walter 2025], the proposed solution allows using online catalogs of local cutting tool suppliers, and not limiting the choice to one manufacturer, since the application is neutral to manufacturers and suppliers in terms of the selected solutions. Each user can add their technical solution for the proposed situation from any supplier or manufacturer to the database. This provides an objective comparison of competitive positions according to the specified criteria, making the application distinct in approach from web applications of aggregators such as MachiningCloud [MachiningCloud n.d.], which focus solely on specific tool manufacturers. Additionally, the logic of decision-making in commercial applications is often closed, making the selection process less transparent. While the methodology proposed by the authors is fully described in the study, in the future, the user will be able to independently choose the necessary criteria and their weights for selection via a graphical user interface.

5 CONCLUSIONS

1. Current tool-selection practices remain experience-dependent and unstandardized, creating the need for an automated, data-driven decision-support system capable of reducing selection time and improving the repeatability of machining decisions.
2. The three-tier architecture provides a scalable and modular foundation for real-time tool-selection workflows and supports future extensions without requiring structural redesign.
3.The hybrid ER/EAV database model ensures both robust data organization and high configurability, enabling rapid integration of new machining attributes and scenarios without altering the underlying schema.
4. The combined CSP filtration and MCDM ranking approach effectively narrows the solution space and delivers technically consistent, criteria-driven selection of tool sets for any defined machining task.
5. The developed interface and workflow significantly simplify user operations, enabling fast configuration of machining conditions and automated procurement actions through integrated supplier data.
6. The practical validation on Inconel 718 machining confirms the reliability of the system's recommendations and its ability to generate ready-to-use technological and procurement documentation.

ACKNOWLEDGMENTS

This research was supported by the Slovak research and educational projects VEGA No. 1/00391/24 and KEGA No. 007 STU 4/2025. The scientific research underlying this paper was conducted within the Ukrainian state-funded research project "Development of a smoke-automated control for the protection of evacuation vehicles and mobile posts" (State registration No. 0124U000538), funded by the Ministry of Education and Science of Ukraine. The preparation of this manuscript was supported by NSP SAIA, which provided favorable academic conditions for research activities. The authors also acknowledge professional assistance in manuscript preparation provided by the non-government organization INDUSTRY 5.UA.

REFERENCES

[Alberti 2011] Alberti, M., et al. Design of a decision support system for machine tool selection based on machine characteristics and performance tests. J. of Intell. Manufacturing, 2011, Vol. 22, No. 2, pp. 263-277.

[Arezoo 2000] Arezoo, B., et al. Selection of cutting tools and conditions of machining operations using an expert system. Computers in industry, 2000, Vol. 42, No. 1, pp 43-58.

[Duan 2021] Duan, Y., et al. A novel cutting tool selection approach based on a metal cutting process knowledge graph. The International Journal of Advanced Manufacturing Technology, 2021, Vol. 112, No. 11, pp. 3201-3214.

[International Organization for Standardization 2012a] Classification and application of hard cutting materials for metal removal with defined cutting edges – Designation of the main groups and groups of application. ISO 513:2012. Geneva: International Organization for Standardization, 2012.

[International Organization for Standardization 2012b] Turning and copying tool holders and cartridges for indexable inserts — Designation. ISO 5608:2012. Geneva: International Organization for Standardization, 2012.

[International Organization for Standardization 2017] Indexable inserts for cutting tools — Designation. ISO 1832:2017. Geneva: International Organization for Standardization, 2017.

[Iscar 2025] Iscar Tool Advisor (ITA) [online]. [cited 7 December 2025]. Available from: https://www.iscar.com/ita.

[Leo Kumar 2019] Leo Kumar, S.P. Knowledge-based expert system in manufacturing planning: state-of-the-art review. International Journal of Production Research, 2019, Vol. 57, No. 15-16, pp. 4766-4790.

[Lin 1997] Lin, A.C. and Wei, C.L. Automated selection of cutting tools based on solid models. J. of Mat. Processing Technology, 1997, Vol. 72, No. 2, pp. 317-329.

[MachiningCloud n.d.] MachiningCloud. Intelligent Manufacturing Data [online]. [cited 7 December 2025]. Available from: https://www.machiningcloud.com.

[Navaneethan 2024] Navaneethan, G., et al. A review of automated cutting tool selection methods. The

International Journal of Advanced Manufacturing Technology, 2024, Vol. 133, No. 3, pp. 1063-1082.
[Ociepka 2015] Ociepka, P. and Herbus, K. Application of CBR method for adding the process of cutting tools and parameters selection. In: IOP Conference Series: Materials Science and Engineering, 2016, Vol. 145, 022029. DOI: 10.1088/1757-899X/145/2/022029.
[Oral 2004] Oral, A. and Cakir, M.C. Automated cutting tool selection and cutting tool sequence optimisation for rotational parts. Robotics and Computer-Integrated Manufacturing, 2004, Vol. 20, No. 2, pp. 127-141.
[Peterka 2004] Peterka, J. A new approach to calculating the arithmetical mean deviation of a profile during copy milling. Strojniski Vestnik-Journal of Mechanical Engineering, 2004, Vol. 50, No. 12, pp. 594-597.
[Phung 2019] Phung, X.L., et al. Expert system based on integrated fuzzy AHP for automatic cutting tool selection. Appl. Sciences, 2019, Vol. 9, No. 20, 4308.
[Pokorny 2012] Pokorny, P., et al. The task of 5-axis milling. Technicki Vestnik-Technical Gazette, 2012, Vol. 9, No. 1, pp. 147-150.
[Prasad 2016] Prasad, K. and Chakraborty, S. A knowledge-based system for end mill selection. Advances in Production Engineering & Management, 2016, Vol. 11, No. 1, pp. 15-28. http://dx.doi.org/10.14743/apem2016.1.206.
[Sandvik Coromant 2025] Sandvik Coromant. CoroPlus® Tool Guide [online]. [cited 7 December 2025]. Available from: https://www.sandvik.coromant.com/en-gb/tools/coroplus-tool-guide.
[Taherdoost 2023] Taherdoost, H. and Madanchian, M. Multi-criteria decision making (MCDM) methods and concepts. Encyclopedia, 2023, Vol. 3, No. 1, pp. 77-87.
[Walter 2025] Walter AG. Walter GPS [online]. [cited 7 December 2025]. Available from https://www.walter-tools.com/en-us/gps.
[Wang 2016] Wang, R., et al. Research on intelligent tool selection system based on Optimization model. Key Eng. Materials, 2016, Vol. 693, pp. 1765-1771.
[Wang 2018] Wang, W., et al. Rule and branch-and-bound algorithm based sequencing of machining features for process planning of complex parts. J. of Intell. Manufacturing, 2018, Vol. 29, No. 6, pp. 1329-1336.
[Wu 2016] Wu, X., et al. Intelligent service platform of manufacturing process and tool based on data warehouse. Procedia Cirp, 2016, Vol. 56, pp. 338-343.
[Xu 2011] Xu, X., et al. Computer-aided process planning–A critical review of recent developments and future trends. International Journal of Computer Integrated Manufacturing, 2011, Vol. 24, No. 1, pp. 1-31.
[Yevdokymov 2025] Yevdokymov, O. and Kolesnyk, V. Development of a methodology for automated selection of cutting tools for a web application. Technical Sciences and Technologies, 2025, Vol. 3, No. 41, pp. 115-130.
[Zarkti 2017] Zarkti, H., et al. Methodology for cutting tool selection system based on STEP_NC. Part 1: development of tool management system according to ISO13399 standard. J Eng Res Appl, 2017, Vol. 7, No. 9, pp. 46-52.
[Zhao 2002] Zhao, Y., et al. Integration of CAD and a cutting tool selection system. Computers & industrial engineering, 2002, Vol. 42, No. 1, pp. 17-34.
[Zhou 2019] Zhou, G., et al. Deep learning enabled cutting tool selection for special-shaped machining features of complex products. Advances in Engineering Software, 2019, Vol. 133, pp. 1-11.

CONTACTS:

Jozef Peterka, Prof. Dr. Ing.
Slovak University of Technology in Bratislava
Faculty of Materials Science and Technology
Jana Bottu 2781/25, 917 23 Trnava, Slovak Republic
jozef.peterka@stuba.sk