DEVELOPMENT OF DECISION TOOL FOR A PROBLEM-SOLVING ACTIVITY USING PAIRWISE COMPARISON TECHNIQUE TO SUPPORT AN INTUITIVE DECISION

N. Tanaiutchawoot [1,✉], N. Bursac [2], J. Gross [3], S. Rapp [1] and A. Albers [1]

[1] Karlsruhe Institute of Technology, Germany, [2] TRUMPF GmbH+Co. KG, Germany, [3] Duale Hochschule Baden-Württemberg Stuttgart, Germany

✉ narucha.tanaiutchawoot@partner.kit.edu

Abstract

Pairwise comparison is basically used to prioritize alternatives and select a solution in product development. Decision errors can happen when the amount of processing information increases. This research proposes an assisted decision tool to prioritize alternatives by using the pairwise comparison technique and a Quicksort algorithm. This software was evaluated in the product development workshop that aims to select the component to further develop in the new product generation. This tool provides high-reliability results, reduces procedure time and is a user-friendly interactive interface.

Keywords: design tools, decision making, problem solving, new product development, pairwise-comparison technique

1. Introduction

During product development, decision makers encounter many problems such as selecting product design and material, selecting strategies for the project and improving the production efficiency. The methods and duration time to handle these problems depend on the situations, environments and complexities of the problems. To arrive at the solution for the problem and make the project move forward, a suitable decision-making method is required. Pairwise comparison is a well-known problem-solving method in the solution selection activity. This method requires an intuitive decision to compare alternatives in pairs and to judge which alternative is preferred. The number of comparisons increases when the number of alternatives increases, which reduces the efficiency of decision-making. To support decision-making within the pairwise comparison method, the assisted decision tool is developed to prioritize alternatives using a pairwise comparison technique and a Quicksort algorithm. This tool can reduce time in the decision-making procedure with a reliable result by automatically ranking alternatives based on the previous selections. Therefore, comparing all pairs is not required to provide the final ranking result. Next, in Section 2, the basic knowledge about the problem-solving activity, decision-making in product development and general decision tools are described. Then the objective is proposed in Section 3. The methodology and the results are explained in Section 4 and Section 5, respectively. In the last section (Section 6), this tool is analyzed and suggestions are made for further development.

2. Literature review

2.1. Problem-solving analysis and methods

According to Dörner (1976), there are 3 components in a problem, which are the undesirable initial state, the desired final state and a barrier that prevents the transformation from initial to final state. Dörner described problem-solving as a transformation of specific circumstances with the help of specific operators. The problem is highly cross-linked when the variables or attributes are depending on each other. Under continuous changing between the whole and the details on processes, problem-solving in engineering is more complicated. Therefore, a problem solver switches between a different degree of details in the mental problem representation to handle the complexity and connect of iterative way (Wynn et al., 2007). This makes a process of generating knowledge and identifying options for the solution procedure more crucial. From these conceptual ideas, many methods for problem-solving were developed such as the TOTE-scheme (Miller et al., 1986), the general problem-solving process by Pahl/Beitz (Pahl and Beitz, 2013) and the SPALTEN method (Albers et al., 2016). The TOTE-scheme shows an element of activities in 4 phases, which are tested, operate, test and exit. These phases are the function in a loop. When the target is achieved, the loop will be exited. The general problem-solving by Pahl/Beitz comprises a basic scheme that is divided into individual working steps starting from defining a task or problem until finding the solution. These operations can be executed sequentially and repeated depending on the requirement of each situation. The SPALTEN method is a method for problem-solving in a universally applicable solution. This method has been developed and applied in many applications, whether the problem occurs in the form of an emergency or planning situation. This tool is also suitable for different degrees of complexity and enables the determination of time, effort and safety-optimal solutions to problems in the product development process (Albers and Braun, 2011). There are 7 working steps in SPALTEN, which are S: Situation Analysis (Situationsanalyse), P: Problem Containment (Problemeingrenzung), A: Alternative Solution (Alternative Lösungen), L: Selection by Solution (Lösungsauswahl), T: Consequences Analysis (Tragweitenanalyse), E: Make a decision and Realization (Entscheiden und Umsetzen) and N: Recapitulate and Learn (Nachbereiten und Lernen). The steps for problem-solving are used depending on the occurrence of that problem. For example, if the cause of a problem has already arisen in an earlier step of the SPALTEN, then the step of alternative solution can be directly used to generate multiple alternatives for solving that problem. Because of the universality and flexibility of the SPALTEN method in different abstraction levels, this method is possibly applied in the entire product development process and several activities that are operated during these phases (Saak, 2006).

2.2. Decision making in PGE-product generation engineering

In the product development process, many decisions are required to create alternatives and find the final solution to solve several problems (Saak, 2006). There are many methods to make a decision under different situations, such as the decision tree (Magee, 1964) and fuzzy logic (Büyüközkan and Feyzıoğlu, 2004). These methods can provide a solution from many criteria and high technical calculation in algorithm, which require a lot of information, knowledge and time. Therefore, they are usually applied in specific situations (Kihlander, 2011). On the other hand, an intuitive decision is commonly used to solve the problem with limited information and in an uncertain environment (Kihlander and Ritzén, 2009). This type of decision can provide a fast response with high reliability of the solution at a specific time, information and environment (Albar and Jetter, 2009). This is a good tool to analyze the problems when a crisis situation occurrs (Muenzberg et al., 2017). This type of decision, however, can be influenced by heuristic decision and biases which lead to different behavior of selection and decision (Bursac et al., 2018). Therefore, types of decision for solving several problems vary depending on the complexity of problem, alternatives, environment and time. Pairwise comparison (David, 1988) is a systematic problem-solving method that requires both rational and intuitive decisions to evaluate and select alternatives. The final solution comes from comparing alternatives in pairs and judging which (of each) alternative is preferred. The order of alternatives can be presented randomly or sequentially, and the results will be analyzed after finishing comparing all alternatives. The number of comparisons also

increases when the number of alternatives is higher, which takes a lot of effort and time for the decision maker and leads the intuitive decision into the wrong direction.

2.3. General software-assisted decision-making

Software-assisted decision-making is computer applications that help decision makers define choices and take decisions by ranking, prioritizing or choosing from alternatives. Most software-assisted decision-making is based on MCDM-Multi-Criteria Decision-Making that involves evaluating and combining the alternatives' characters from two or more criteria (Belton and Stewart, 2002). Another algorithm is the AHP (Analytic Hierarchy Process), which also has a basic background from the MCDM method. These programs usually include a variety of features and tools which are similar, such as pairwise comparison, sensitivity analysis, group evaluation and web-based implementation (Weistroffer and Li, 2016). Examples of these programs are 1000minds (Ombler and Hansen), Ahoona (Courtesy of Ahoona Corp, 2013), iDecide (iDecide project, 2016) etc. Most of them require criteria, weight and information to compare alternatives. Moreover, the process to run the evaluation is complex and requires much time.

3. Objective

As described in the literature review, the pairwise comparison technique is an effective problem-solving tool in the solution selection activity. Most studies focus on comparing techniques such as an order of alternatives (Bozóki et al., 2013) and measurement techniques (qualitative/quantitative) (Krajbich et al., 2010). However, a method to reduce the effort required to make a comparison by reducing the number of comparisons, which supports an intuitive decision, is ignored. This paper aims to develop a decision tool to support problem-solving in the solution selection activity based on the pairwise comparison technique. This tool can increase the proceeding time to compare alternatives and provide an effective result. Moreover, this tool requires low effort from the user to evaluate and rank alternatives. With this objective, the following research questions are raised:
1) How can a decision tool be developed to assist intuitive decision-making in a problem-solving activity to prioritize alternatives based on the pairwise comparison technique?
2) Which technique can be applied to reduce the effort required to compare alternatives in a pairwise comparison technique?

4. Procedure

4.1. Basic requirement

In the first step, the programming language and program structure are analyzed with respect to basic requirements for the decision tool. These requirements were listed based on decision tools from the literature and the researcher's experience, which are: 1) The decision tool should be able to process online and be accessed from different platforms. 2) The decision tool must have a database to store the data and result from the evaluation. 3) The decision tool should be able to support an evaluation in a multi-project. Users can create different projects in the same database, which is a private database for individual users. Results from each project will be recorded in this database but in different folders. From these requirements, the best implementation option is a web server application that is able to additionally store and organize the data such as a file image. A local database is, therefore, created as a local database in a Web server in XAMPP and the program is developed using PHP and HTML language. In the beginning step of this program, all information in this tool will be collected in one database named "spalten_tool".
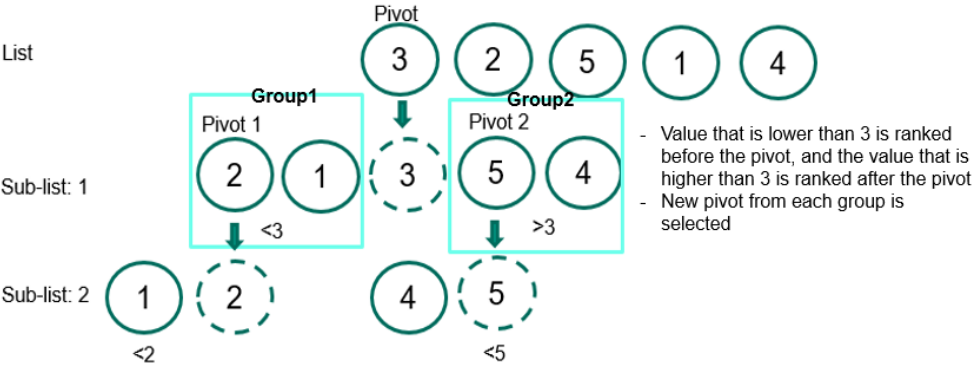
4.2. Implementing an algorithm in the decision tool to prioritize alternatives by using the pairwise-comparison technique and the Quicksort algorithm

This decision tool has been developed to prioritize alternatives for a selection of solution in the problem - solving activity. An algorithm to prioritize alternatives in this tool is a pairwise comparison algorithm, which aims to analyze and select alternatives in pairs. If n is assumed to be a total number of alternatives,

comparing times is equal to $\sum_{i=0}^{i=n}(i-1)$. That means comparing times will be increasing every n-1. Then, the alternatives are prioritized according to the number of times selected. This behavior requires a lot of effort from the decision maker to sort alternatives, especially when criteria are implemented. The ability of consideration and decision are decreasing when many alternatives and information are provided, specifically when an intuitive decision is applied for making a decision. Reducing the amount of information to provide the user is, therefore, an idea to assist decision-making when using the pairwise comparison technique to prioritize alternatives. One algorithm that is suitable to propose alternatives in each round and rank alternatives by is the sorting algorithm. This algorithm concept is to rearrange an array of alternatives according to the intensity of selection. There are many types of sorting algorithms such as Shellsort, Bubble Sort, Quicksort, Merge Sort and Heapsort. Each type of sorting algorithm provides different advantages and disadvantages. However, Quicksort, that provides a fast solution and requiring small additional amounts of memory to perform sorting factors, is selected. In the next step, the Quicksort algorithm will be described to help the reader understand an algorithm in the software that was developed from this algorithm. Then the algorithm in software-assisted decision-making will be explained.

4.2.1. Quicksort algorithm

This algorithm is based on the principle of "divide-and-conquer". It works by setting a pivot element that divides the list of elements into two parts lists. All components are compared with the pivot element and ranked before and after the pivot element. Then the sub-lists are defined pivot element of each group and recursively sorted by this pivot element of each group. The processes are repeated until all elements are ordered. There is no strict rule for selecting the pivot element, but the selection comes from the strategy and performance of the algorithm. Figure 1 shows an example of the Quicksort algorithm to reorder the value of 1 (the lowest value) to 5 (the highest value) by selecting the first element to be the pivot element.

Figure 1. An example of the Quicksort algorithm

4.2.2. Applied Quicksort algorithm in the software assisted decision-making

Figure 2 shows an example of an algorithm in software-assisted decision-making. A pivot element in the program is selected from the middle alternatives when the number of alternatives is odd. If the number of alternatives is even, the alternative, on the left side of the middle alternatives, is selected to be the pivot. The sorting alternative in our program is ranking alternatives based on selection behavior: 1) selected alternative, 2) rejected alternative and 3) no difference in quality between both alternatives. Alternatives in pairwise comparison are assigned to be a pivot and a competitor. If the competitor is selected, the competitor is assigned to the right side of the pivot. On the other hand, if the pivot is selected, the competitor is assigned to the left side of the pivot. If both competitor and pivot are equal in quality, the competitor is attached with the pivot and has the same value as the pivot. These examples of each case can be seen in the 1st, 2nd and 3rd rounds in Figure 2 consequently. In each comparison, the competitor is changing for the next round, but the pivot is not changed until all alternatives in a group are compared to the pivot.
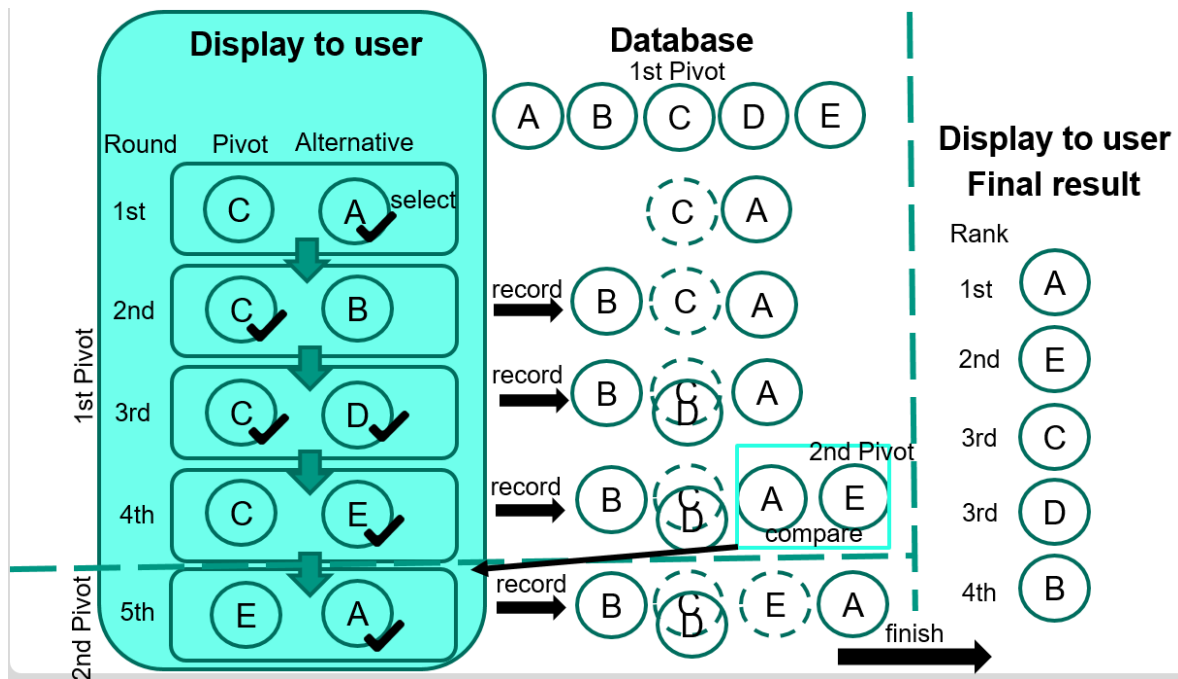
Figure 2. An example of an algorithm in software-assisted decision making

When all alternatives are compared, the alternatives (competitors) are separated into 2 groups that are on the left side and on the right side of the pivot as shown in the 4th round in Figure 2. Then the next pivot and competitors are selected within their own groups using a similar algorithm.

In the end, a result of this software shows the ranking of all alternatives from the selection. A decision maker can evaluate alternatives in the second round when another decision maker would like to make an evaluation in the same project. The final result will combine the result from the first and the second round. The decision maker can also reset the result and restart the evaluation process. However, results from this software can only assist the decision maker to prioritize alternatives based on intuitive decision instead of proposing a final alternative based on criteria.

4.3. Graphical user interfaces and components in the decision tool

6 main pages are developed in the decision tool as shown in Figure 3.

- Page 1 Overall project is the page that shows all project lists. A decision maker can create many projects for prioritizing alternatives in the same database, which support the basic requirement
- Page 2: Number of alternatives are required to generate a template on the next page
- Page 3: Implemented alternative information is the page to add alternative information. A name is required to identify alternatives. Description and image are additional options.
- Page 4: Overview of alternative information has shown all alternative information before starting a pairwise comparison process. A decision maker can recheck alternative information before starting the evaluation and possibly add an alternative in the evaluation.
- Page 5: An evaluation process and alternative selection is the page to compare alternatives in pairs using pairwise comparison and the Quicksort algorithm. A pivot alternative is on the left side and a competitor is on the right side. A decision maker clicks on the favorite alternative. The "Equal button" is provided when a decision cannot be made or both alternatives have the same value.
- Page 6: Result of alternative prioritization shows alternatives in order of ranking. If alternatives are equal in the pairwise comparison process, the result will show the same rank. The value in "Inkrement" columns, that is shown on Page 6 in Figure 3, describes a number of iterations to make an evaluation. If the decision maker continues the second evaluation, the value in "Inkrement" will be 2.
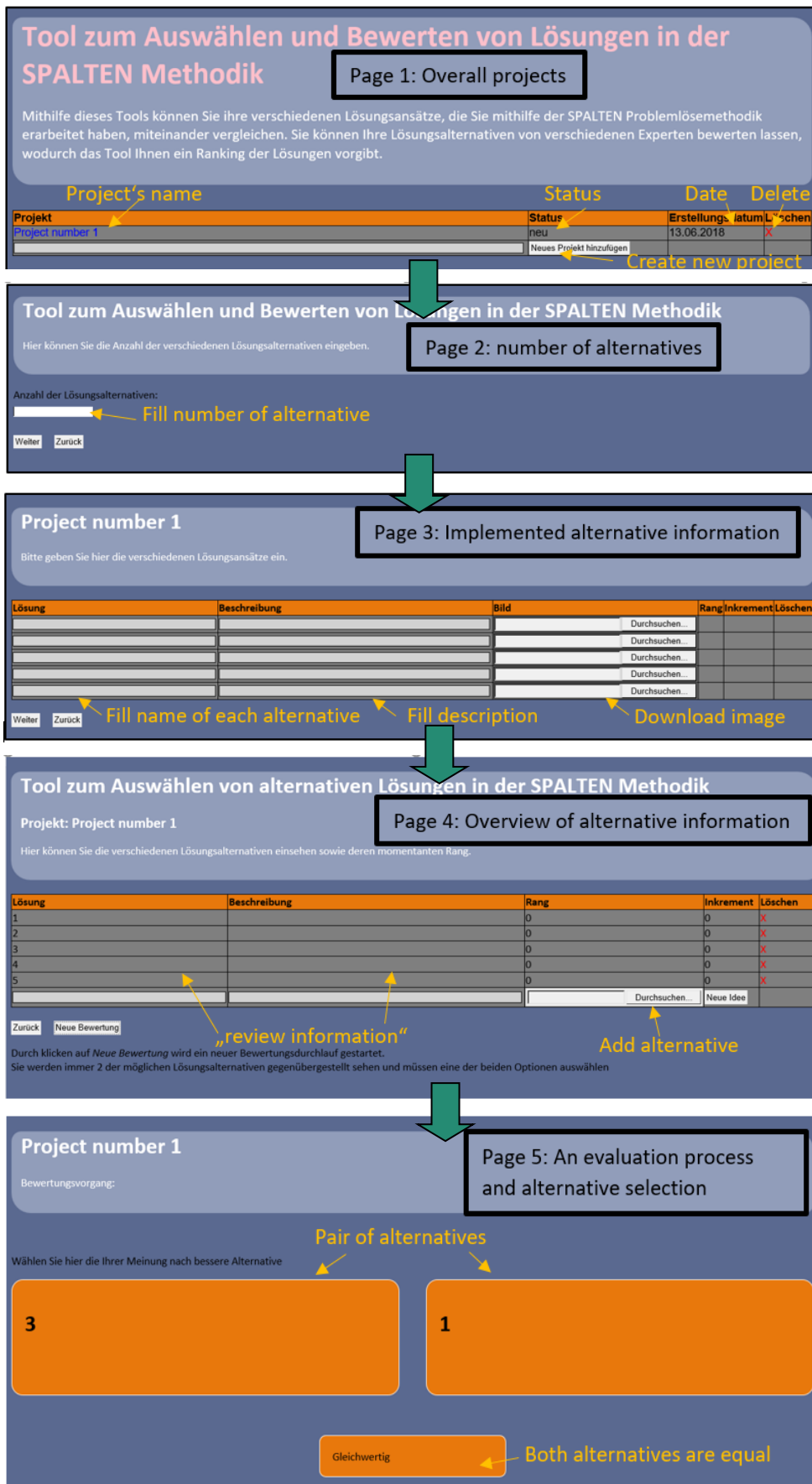
Figure 3. 6 main pages for the user interface in the decision tool

5. Evaluation and result

The decision tool was implemented in a workshop with a company partner to assist the decision maker in prioritizing alternatives for further development in a machine tool. Different world economic scenarios were used to describe their impact on a special product of this company. There are 10 different components of this machine, which were prioritized in the workshop under certain scenarios. Table 1 shows 10 different components and descriptions of this machine.

Table 1. 10 different components of the machine that were prioritized in software-assisted decision-making under certain scenarios

Components	Description
Operating element	Operating element that is attached directly to the machine
Small partial discharge	Containers in which small parts are ejected
Laser cutting machine	Laser cutting machine for cutting the sheets
Main partial discharge	Removal station of the main part
Raw sheet loading	Inserting the sheets into the processing machine
Smart Gate	Sensors and ejection cylinders solve the intelligently cut small parts from the residual grid
Smart Lift	Flexibly positional pins which press the cut components against the suction plate of the Sort Master
Software	Operating software with which the machine center is controlled
Sort Master	A suction element which intelligently sorts and stacks the cut components
Extending table	Extending table on which the cut and stacked components are extended

These components were prioritized under 5 different scenarios, which look at possible developments in the medium to long term future (20 years). Each scenario has different characters, such as a high recycling rate or low unit labor costs. The components of this machine were evaluated and prioritized related to each scenario. The results can possibly be the same.

The software-assisted decision-making was actively used during the workshop to determine a ranking for various components of the machine. After analyzing the results from the decision maker, the results from this decision tool are acceptable and then be applied in the next step of product development. This implies that the results from software-assisted decision-making are reliable. Moreover, the researcher made a survey to get some feedback from the user to evaluate the functionality and user compatibility of this program. Results from the survey show many advantages of this tool due to an interaction between the program and the user. Alternatives with descriptions and images assist the decision maker to be more understandable in alternatives. The pairwise comparison technique supports the decision maker to concentrate on details from both alternatives. The time to process this evaluation (excluding preparation) was lower than 8 minutes per scenario. The number of times to compare alternatives is also lower than $\sum_{i=0}^{i=n}(i-1)$ in each scenario when n=10 in this case, which help the proceeding time for evaluation to be faster. For example, the participant got the result of ranking after comparing 22 pairs within 4 minutes in the first scenario. Evaluating and ranking alternatives using this developed algorithm helps the participant to ignore some pairs of comparison, which are automatically ranked regarding selection behavior from the previous processes. However, the user recommended some functions that should be implemented in the next version of this program such as a "back" button within a pairwise comparison process, a record of all results from the previous comparing until current comparing during the pairwise comparison process and a frequency number of alternative selections. Moreover, the feasibility to access and transfer information and results between different programs should be developed. The example is to export a result to an excel file and import alternatives and descriptions from an excel file. The experiment should also be implemented by comparing the results and feedback when using this software-assisted decision-making and general methods such as a decision matrix to prioritize alternatives.

6. Discussion and conclusion

This paper presented a decision tool in the selection-solution stage in the problem-solving activity based on a pairwise comparison technique. This tool is developed based on a conceptual idea to reduce the number of comparisons in the pairwise comparison technique and to support an intuitive decision. Therefore, information about each alternative is separated in the required information (name) and

additional information (description and picture). The criteria are not described to simplify the task. The components that are applied in this tool can answer the first question about how to develop a decision tool to assist decision-making in the selection-solution activity based on the pairwise comparison technique. The Quicksort is a technique that answers the second question about which technique can be applied to reduce an effort for comparing alternatives in a pairwise comparison technique. This technique reduces the proceeding time and effort to prioritize and select an alternative. Moreover, a result of this technique provides reliability, which is proved by a mental crosscheck by the decision maker.

However, some functions such as a back button and the data history of previous selections are required for further development in the new tool version. The back button is required to edit the previous decision. The data history of previous selections can also support decision makers to be aware of his or her previous decision. This program is also limited to processing in a local system; therefore, the next version of this program should be online and should be able to support other devices such as a mobile phone or tablet that are the other technologies for social communication and general activities. Then, personal information and a security system should be considered and implemented in this tool.

References

Albar, F.M. and Jetter, A.J. (2009), "Heuristics in decision making", *PICMET'09-2009 Portland International Conference on Management of Engineering & Technology*. IEEE.

Albers, A. and Braun, A. (2011), "A generalised framework to compass and to support complex product engineering processes", *International Journal of Product Development*, Vol. 15 No. 1-3, pp. 6-25.

Albers, A. et al. (2016), "iPeM – Integrated Product Engineering Model in Context of Product Generation Engineering", *Procedia CIRP*, Vol. 50, pp. 100-105.

Belton, V. and Stewart, T. (2002), "Multiple criteria decision analysis: an integrated approach", *Springer Science and Business Media*.

Bozóki, S. et al. (2013), "Analysis of pairwise comparison matrices: an empirical research", *Annals of Operations Research*, Vol. 211 No. 1, pp. 511-528.

Bursac, N. et al. (2018), "Decision Heuristic in Pge-Product Generation Engineering", In *Proceedings of TMCE*.

Büyüközkan, G. and Feyzıoğlu, O. (2004), "A fuzzy-logic-based decision-making approach for new product development", *International journal of production economics*, Vol. 90 No. 1, pp. 27-45.

Courtesy of Ahoona Corp. (2013), "Ahoona helps you makea decision about anything!", Retrieved June 2, 2019, from https://www.ahoona.com/

David, H.A. (1988), *The Method of Paired Comparisons*, Oxford University Press, New York.

Dörner, D. (1976), "Problemlösen als informationsverarbeitung".

iDecide project (2016), "State of the art for inclusive decision making", available at: www.idecide-project.eu.

Kihlander, I. (2011), "Managing Concept Decision Making in Product Development Practice", *Doctoral dissertation*, KTH Royal Institute of Technology.

Kihlander, I. and Ritzén, S. (2009), "Deficiencies in Management of the Concept Development Process: Theory and Practice", *Human Behavior in Design*, Vol. 9.

Krajbich, I., Armel, C. and Rangel, A. (2010), "Visual fixations and the computation and comparison of value in simple choice", *Nature neuroscience*, Vol. 13 No. 10, p. 1292.

Magee, J.F. (1964), "Decision trees for decision making", *Harvard Business Review*.

Miller, G.A., Galanter, E. and Pribram, K.H. (1986), "Plans and the structure of behavior", Adams Bannister Cox.

Muenzberg, C. et al. (2017), "Identifying product development crises: The potential of adaptive heuristics" In *DS 87-2 Proceedings of the 21st International Conference on Engineering Design (ICED 17), Vol 2: Design Processes, Design Organisation and Management,* Vancouver, Canada, 21-25 August 2017, pp. 189-198.

Ombler, F. and Hansen, P. (2015), "1000Minds Software".

Pahl, G. and Beitz, W. (2013), "Konstruktionslehre: Methoden und Anwendung", Springer-Verlag.

Saak, M. (2006), "Entwicklung eines Konzeptes und eines Prototypen für ein rechnergestütztes Werkzeug zum effizienten Einsatz der Problemlösungsmethodik 'Spalten'. Development of a concept and of a prototype for a computer-aided tool for the efficient employment of the problem solving methodology 'Spalten'", *IPEK, 2006*.

Tversky, A. and Kahneman, D. (1974), "Judgment under uncertainty. Heuristics and biases", *science*, Vol. 185 No. 4157, pp. 1124-1131.

Weistroffer, H.R. and Li, Y. (2016), "Multiple criteria decision analysis software", in *Multiple Criteria Decision Analysis*, Springer, pp. 1301-1341.

Wynn, D.C., Eckert, C.M. and Clarkson, P.J. (2007), "Modelling iteration in engineering design" In: *DS 42: Proceedings of ICED 2007, the 16th International Conference on Engineering Design, Paris,* France, 28-31 July 2007, pp. 693-694.