



HAL
open science

VA++ - The Next Generation of Value Analysis in TRIZ

Christian Iniotakis

► **To cite this version:**

Christian Iniotakis. VA++ - The Next Generation of Value Analysis in TRIZ. 22th International TRIZ Future Conference (TFC), Sep 2022, Warsaw, Poland. pp.29-38, 10.1007/978-3-031-17288-5_3. hal-04449776

HAL Id: hal-04449776

<https://inria.hal.science/hal-04449776v1>

Submitted on 15 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

VA++ - the Next Generation of Value Analysis in TRIZ

Christian Iniotakis ^[0000-0002-7951-2935]

Ulm University of Applied Sciences, Prittwitzstr. 10, D-89075 Ulm, Germany
christian.iniotakis@thu.de

Abstract. Around twenty years ago, two methods relevant for systematic innovation and improvement - TRIZ and Value Analysis - have been merged in a specific way, which was then incorporated in software packages about innovation and became part of certified TRIZ education as well. An intended key purpose of this nowadays established method of Value Analysis in TRIZ is to identify system components of low ideality - or value, respectively -, i.e. parts of the system, that do not give a satisfactory functional contribution in relation to their cost. Hence, Value Analysis should point out the sweet spots for subsequent improvement, innovation or even patent circumvention activities. Unfortunately, the commonly used standard approach for Value Analysis in TRIZ, which is based on a function ranking algorithm, leads to results, that are inconsistent and not trustworthy in general. This work illustrates these shortcomings and explains their origins. Derived from key requirements necessary for a meaningful concept, VA++, a new advanced approach for Value Analysis in TRIZ, is presented and validated.

Keywords: TRIZ, Value Analysis, Systematic Innovation, Function Analysis, Function Ranking, Ideality, Algorithm, CAI, Trimming.

1 Introduction

Ideality is a key concept in TRIZ, which is supposed to indicate, if a system - such as a product, a process, or even a business model - will prove successful under competitive circumstances on the long run. Ideality is typically described by the ratio of functionality divided by costs (cf. e.g. [1, 2]). Here, functionality is a measure for all the useful functions, i.e. the main function(s) and also further additional ones, the system delivers to the supersystem or its environment, respectively. Roughly speaking, functionality expresses, how good the system is considered to be in total. The system costs describe, how expensive the system is. Depending on the perspective and the actual problem setting, the costs might cover different aspects. For example, the costs could be represented by material and assembly costs, by the market price, or also by a much more general concept. In the latter case, usage and long-term maintenance costs might also be taken into account, as well as negative side effects and harms done by the system. Since ideality expresses, how good a system is per general costs to bear, it obviously is very similar to the well-known value concept in the framework of Value Analysis according to Miles [3]. As a consequence, ideality and value are often used

interchangeably. However, the notion 'value' has several more general meanings and might be interpreted in different ways by different individuals, whereas the term 'ideality' is quite unique. For this reason, the latter one will be predominantly used for the rest of this work. It is important to realize, that most TRIZ methods have the effect or are designed to increase ideality, e.g. by solving problems or improving the system. Thus, there is a significant implicit relation between TRIZ and Value Analysis, and a deeper methodical combination is very promising in general [4]. The focus of this work, however, is set on the approach of an explicit integration of Value Analysis into TRIZ based on Function Analysis.

2 Function Analysis

Function Analysis is one of the most important TRIZ methods. It allows to analyze systems in detail and is found in many newer typical TRIZ textbooks, e.g. in [1, 2]. Together with the proposal, to extend Function Analysis also to spatio-temporal parameters, Litvin et al. [5] give more details about its historic evolution. The first step of Function Analysis is to divide the system into basic parts called components. Likewise the environment is represented by its relevant parts, the so-called supersystem components. Then, all of these components are checked for the relevant functions, they perform on each other, in a systematic way. Identified functions are expressed in a very compact manner - typically a verb only -, which just represents the underlying and sometimes very complex functional details in form of a shorthand description. Applying Function Analysis to a given system eventually results in a complete picture containing all the system and involved supersystem components and their relevant functional interplay. Based on that, it becomes clear from the functional perspective, how the system works, in particular how it manages to provide its key functionality in form of the main functions, and how each component of the system contributes to it internally. For illustrative purposes, a simplified example showing the result of Function Analysis for a drone for autonomous package delivery is depicted in Fig. 1.

3 VA in TRIZ

Within TRIZ, a key reason for explicitly applying value analysis methods to a given system is to determine the individual values or idealities, respectively, of its components. The intention is to find suitable sweet spots for system improvement and innovation. For example, a common general approach is to identify components of low ideality as trimming candidates. Trimming denotes a TRIZ method, which has the goal to get rid of one or more system components, while the remaining system is still providing its functionality unaffectedly (cf. e.g. [1, 2, 6]). Obviously, this only works out, if the remaining components are thoroughly enabled to take on the functions of the trimmed ones, or if these functions are rendered unnecessary. In any case, this involves good ideas, and often requires very innovative system changes.

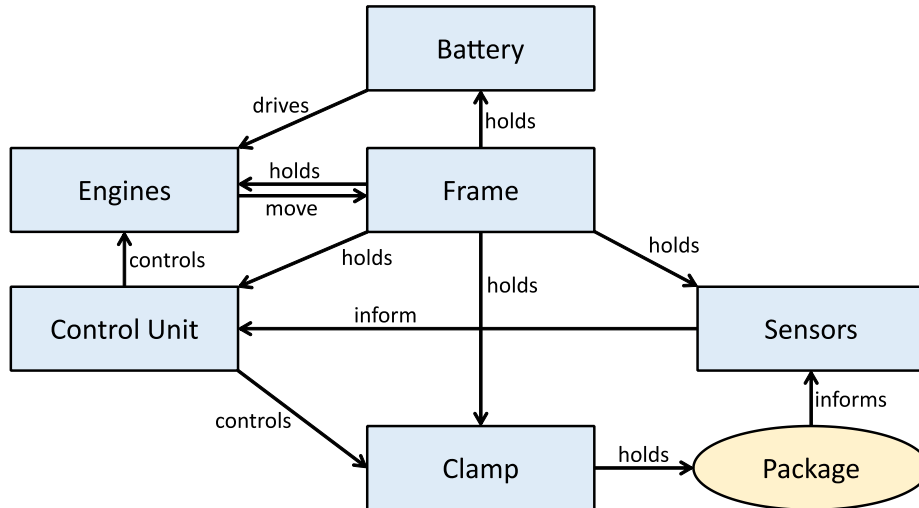


Fig. 1. Sample result of Function Analysis applied to a drone for autonomous package delivery.

The drone is represented by six top level system components (frame, control unit, engines, battery, sensor, and clamp), whereas the relevant supersystem component is the package. Note, that the main function of the drone to “transport package” is not explicitly stated here, but implicitly results from the full functional interplay: After the control unit knows about the delivery location, e.g. from an address written on the package, it controls the engines to move the frame to the desired position. The frame itself holds all of the other system components, in particular the clamp holding the package. Thus, the package eventually arrives where it should.

A component of low ideality is a natural candidate for trimming, if the system ideality should be increased in an efficient way, because it doesn't contribute much for its costs. Experts with a good intuition and deep knowledge of the system are sometimes able to directly point out components, which are likely to have a low individual ideality. Since in practice, unfortunately, those experts are not always available or able to do so, or there are several experts of different opinions, there is a strong need to put the ideality estimation on a more profound and systematic basis. Currently, there is one major approach known to a significant part of the TRIZ community, which was developed for that very purpose. It takes the results of Function Analysis as a key input, and is supposed to return the individual idealities of the system components as a main result. Actually, this specific algorithmic approach is also denoted as 'Value Analysis' in TRIZ, but will regularly be abbreviated as VA in the following, to avoid confusion with the more general methodical framework of Value Analysis in a wider sense. Standard VA as an established procedure can be broken down into three steps:

- Step 1: An empirical algorithm performs a so-called function ranking. Based on this, a functional value for each function is determined, which serves as a measure for its functionality.

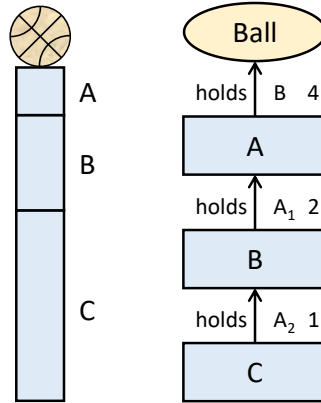


Fig. 2. (Left) A toy tower of three different bricks A, B, C, constructed to hold a ball. (Right) Result of the Function Analysis of the toy tower. For each function, both the function rank and the functional value according to standard VA are indicated.

- Step 2: Likewise, a functional value for each component is determined by summing up the functional values of all the functions performed by it.
- Step 3: The functional value of each component is taken as the basis for the final analysis. For example, the ideality of each component is directly evaluated by dividing its functional value by its costs, or the components are compared by simply placing them in a two-dimensional diagram of functional value over cost.

Involved in automated or semi-automated innovation software [7, 8, 9], this standard VA is supposed to provide “sophisticated Value Analysis algorithms to diagnose the weak elements” [9: 2]. As a consequence, it is also implemented in CAD software to extent it towards computer-aided invention (CAI) capabilities [10] and incorporated in processes for patent circumvention [11]. Furthermore, standard VA is part of renowned TRIZ education and certification according to the International TRIZ Association MATRIZ (cf. e.g. [6]) and applied in typical TRIZ innovation projects, e.g. [12, 13]. There are proposals for extending it, for example to include the impact of harmful functions [11] or to take care of different time settings [13]. Regarding Step 1, the most established original algorithm for function ranking will be used as a reference for standard VA in the following. It is described in the Appendix in more detail. There are some works with specific changes in ranking the functions or the actual way to calculate their functional value out of the function ranking (e.g. [11, 14]), but nevertheless a common feature still is, that the functions themselves and most of their underlying properties are completely ignored. In other words, all of the algorithms and procedures only take into account, that there is a functional relation of a given type, but in the end they do not really care about what this function actually is and does.

If the first two steps of standard VA are applied to the autonomous drone based on its Function Analysis as depicted in Fig. 1 for illustrative purposes only, some of the results are quite unexpected and remarkable. For example, the control unit controls

both the engines and the clamp, and it is reasonable to assume the first function to be much more complex, demanding, and important than the latter one. The VA function ranking indicates just the opposite: The functional value for controlling the clamp (4) is two times higher than the one for controlling the engines (2). Regarding the components, the functionality of the frame (12) is calculated to be even higher than the functionalities of control unit (6), engines (3), and battery (2) summed together. Moreover, also the clamp (6) itself even yields the same functional value as the control unit, and excels that of engines and battery summed together.

4 Testing VA

In the following, the established VA approach is tested on a very simple toy model system: A child builds a tower of bricks in order to hold a ball at its top. The higher the tower, the better. The model is chosen that simple on purpose, to allow an immediate and objective verification of results, which does not depend on deep expertise or long personal experience with the system. In contrast, highly complicated systems with huge numbers of components and functions are often subject to individual interpretation or bear an intrinsic complexity, such that a serious and objective verification of results is much harder or even impossible. Certainly, this is one important reason, why the shortcomings of standard VA in TRIZ have not been clearly recognized in previous applications.

As a first test scenario, the child is considered to build a tower of exactly three different bricks A, B, and C, such that the ball is held at a height of 70 cm above the floor. The bricks are different in length and cost only, all other brick properties are assumed to be fully identical. Both the tower and the result of its Function Analysis are depicted in Fig. 2. In addition, the function rank and the functional values according to Step 1 of standard VA are already attached to each function. The key characteristics of the three bricks, namely their individual lengths and their individual costs, are listed in Table 1, and according to Step 2, each brick gets the functional value of the function it performs. Finally, the ideality of each brick can be derived according to Step 3 by simply dividing its functional value by its costs. Note, that the idealities are given in dimensionless units, normalized to the maximum value. According to standard VA in TRIZ the result is clear: Brick C is supposed to have the lowest ideality by far, and would be recommended for trimming. In contrast, the idealities of brick B and A are assumed to excel that of brick C by a factor of two, or even three, respectively. Since ideality should indicate, what you get for the money, the recommendation of standard VA does not even work qualitatively, but fails completely. Obviously, the result contradicts common sense and basic logical thinking, which tells us the exactly opposite ideality ranking as seen in the right column of Table 1: Brick C has the highest ideality, it contributes 40 cm to the tower height for costs of 3 €. If this is set to 100%, brick B has an ideality of 50%, contributing only half of that for the same costs. Finally, brick A contributes 10 cm for 4 €. In relation to brick C, this yields an ideality of $10/4 \cdot 3/40 = 3/16$, which is approximately 19%.

Table 1. Characteristics of the three bricks A, B, and C. The two right-hand columns show the brick ideality in dimensionless, normalized units, according to standard VA in contrast to the ideality by pure common sense and logical thinking.

Brick	Length	Cost	Functional Value	Ideality (VA)	Ideality
A	10 cm	4 €	4	100%	~19%
B	20 cm	3 €	2	~67%	50%
C	40 cm	3 €	1	~33%	100%

In a hypothetical subsequent trimming scenario, standard VA would pose the challenge, to enable brick A and/or brick B to make up for the height of brick C. Even if an innovative solution can be found, its implementation must not involve an additional cost impact higher than 3 € - the costs of the trimmed brick C - to still result in a net increase of the overall tower ideality at the end. In contrast, guided by and based on the real idealities, the bricks A and B could simply be replaced by a second brick of type C, yielding a new two component tower which is even higher (+10 cm), drastically cheaper (-4 €), and thus clearly more ideal than before. This option - to increase the overall tower ideality by a factor of ~190% due to rather simple straightforward measures - relies on brick C only. Therefore it would certainly be overlooked following standard VA, which proposes to get rid of exactly that particular brick.

As a second test scenario, the child is assumed to build the tower with a set of N identical bricks. Since the costs are also the same for each brick, the ideality of a given brick is fully proportional to its functionality. With n denoting the position of the brick in the tower counted in a bottom-up direction, it is straightforward to see that the functionality according to Steps 1 and 2 of standard VA is given by a functional value of n for the brick at position $1 \leq n < N$, and a functional value of $N + 1$ for the brick at the top position $n = N$. As a consequence, each of the identical bricks exhibits a different ideality. In normalized units, the ideality ranges from $1/(N + 1)$ for the bottom brick at position $n = 1$ to 100% for the brick at the top position $n = N$. Therefore, the factor between the calculated idealities of the top and the bottom brick is $N + 1$, which obviously scales with the total number of bricks N . If the system of bricks should be improved by tackling the bricks of lowest ideality according to standard VA in TRIZ, there would be a distinct and strong recommendation to primarily focus on the bottom brick, while the top brick should not be touched at all. Since we speak about totally identical bricks, which are all involved in a comparable way and could even be interchanged in their positions without any system change, this result is not only weird, but again simply wrong. In contrast, all bricks need to have equal idealities, such that it's not possible to differentiate between them in this simple test case. For illustrative purposes, Table 2 shows the resulting numbers for a sample tower consisting of exactly 9 identical bricks.

With standard VA in TRIZ failing already on these simple test scenarios, there is no reasonable or obvious argument, why it should perform better in general, when the system and its functional interplay are even more complicated. The above example of the autonomous drone might serve as an illustrative indication for this as well.

Table 2. A set of $N = 9$ fully identical bricks of the same individual costs are piled up to a tower. According to standard VA the normalized ideality of the bricks covers a full range depending on the brick position, as depicted in the second line. The (maximum) ideality of the top brick ($n = 9$) is higher than the (minimum) ideality of the bottom brick ($n = 1$) by a factor of $10(N + 1)$. This result is again in contrast to the ideality by pure common sense and logical thinking, demanding the same ideality for all the identical bricks here.

Brick pos. $n \rightarrow$	1	2	3	4	5	6	7	8	9
Ideality (VA)	10%	20%	30%	40%	50%	60%	70%	80%	100%
Ideality	100%	100%	100%	100%	100%	100%	100%	100%	100%

5 What's wrong with VA?

It might seem obvious, that the VA function ranking algorithm as used according to Step 1 and described in more detail in the Appendix causes the problems. As discussed above, it determines functional values of each function out of the functional structure only, while not taking into account the real functional content and situation. This clearly has to affect the reliability and quality of the final results of standard VA in a negative way. It should be pointed out, that this problem also holds for all algorithmic variants, which do not rely on concrete properties and circumstances of the identified functions. As an actual example, simply adapting values or involving weighting factors in the functional ranking algorithm is not sufficiently appropriate to resolve this issue.

Moreover, a key message of this work is, that in Step 2 of VA, which is undisputed so far, there is a second problematic issue, that lies even deeper from a conceptual perspective. Not only in Mathematics and Theoretical Physics it is an important concept that the modelling of a given system could typically be done in various valid ways, while a meaningful result about the system must not depend on the individual choice of the model eventually. In our case, the system might be described by choosing different sets of components, and these components might themselves be decomposed into sub components or recombined to groups again. No matter, if these modelling steps are necessary for a proper system description, seem impractical or inappropriate, or are even performed purely virtually: A profound concept has to ensure that the functional value of a component is not at all affected by those modelling operations around it, be it directly involved or not. Expressed in a more theoretical way, a meaningful functional value of a component is required to be invariant under any type of valid component operations, such as e.g. combination and decomposition. This requirement also directly ensures the functional value of a system being invariant under scaling of the model, i.e. the actual level of system description. As a key result of this consideration, the simplest concept for the functional value δ_C of a component C , which meets this type of requirement, is found to be

$$\delta_C = \phi_{C,out} - \phi_{C,in}.$$

Here, $\phi_{C,out}$ is the total functional value of all functions performed *by* C , and $\phi_{C,in}$ represents the total functional value of all functions performed *on* C , analogously. In contrast, the functional value of a component according to Step 2 of standard VA is determined by

$$\delta_{C,VA} = \phi_{C,out} ,$$

and does not fulfill the requirement in general, leading to arbitrary or wrong results in most cases. It should be mentioned, however, that on some occasions the results coincide by chance. In the special case of a component, which does not require a (relevant) functional input, for example, its functional value is only determined by its functional performance. Also the difference in the functional values of two competing systems happens to be correct, as long as both of them rely on similar functional input from the surrounding supersystem components. So it turns out, that due to conceptual requirements, the functional value of a component should generally be determined by its net functional contribution and not by its absolute functional performance.

6 Validating VA++

Now, how do results for the simple brick tower test cases change, if Value Analysis is applied correctly? First of all, the functional values of all involved functions need to be determined properly, taking into account their actual content, and not by an algorithm, which neglects that. For the concrete system of the brick tower, we only deal with functions that hold either the next brick or the ball in a similar way. What differs, however, is the absolute height level (e.g. above the floor). In the first test scenario of three different bricks, brick C holds brick B at 40 cm, brick B holds brick A at 60 cm, and brick A holds the ball at 70 cm. In the second test scenario of N identical bricks, the brick at position n simply provides the “hold”-function at the level of $n \cdot h$, where h is the height of a single brick. These individual height levels provided by the functions are suitable for measuring their corresponding functional values. As a next step, functional values of each brick have to be determined based on its net functional contribution, instead of its functional performance. It is easy to see, that the functional contribution of each brick is to increase the level of the holding function exactly by its specific height. Evaluated in units of our free choice, the results may be written as $\delta_A = 10$ cm, $\delta_B = 20$ cm, and $\delta_C = 40$ cm in the first test case, and $\delta_n = h$ for each of the identical bricks in the second test case. Setting these functional values in relation to the brick costs, the correct idealities are perfectly reproduced for both test scenarios. So the explicit use of Value Analysis in TRIZ is expected to work fine, as long as the afore-mentioned two issues about the evaluation of functions and contribution of components are properly taken care of. Correcting these two shortcomings of VA results in VA++, the improved next generation of Value Analysis in TRIZ.

7 Resume and Outlook

This work shows, that standard VA in TRIZ does not give trustworthy results, for two different reasons. The first reason is, that functions are evaluated by an algorithm, which lacks system insight. The second reason is to use an inappropriate concept for determining the functional value of a component. The first issue has to be addressed by evaluating the functional value of functions on the basis of system knowledge, at least at an qualitative level. The second issue can simply be solved by changing to a different, meaningful concept, which takes into account the net functional contribution of a component and not its absolute functional performance. Following this corrected approach results in an improved next generation of Value Analysis in TRIZ termed VA++. Due to its solid conceptional footing, VA++ is not only supposed to enable access to correct results, but it is also expected to be of a much higher robustness with respect to typical questions in system modelling, that appear rather often in practical application. To give some concrete examples, the results of VA++ are unaffected by the pure modelling decision to attach two or more functions to one arrow, and also by the explicit incorporation of self loops in the function model. Furthermore, by derivation, VA++ is fully consistent with valid TRIZ operations, such as trimming of all types, and grouping and decomposition of components as well.

Appendix: Function Ranking and Function Values

According to Step 1 of standard VA in TRIZ, the functional value of each useful function is derived from an empirical function ranking algorithm, which is summarized here in short ([8, 9], also cf. [6]): Each main function gets the highest function rank denoted as basic, B. Any other function acting on the same object as a main function is considered to be basic, too. Each function acting on a component, which itself performs a basic function, is of rank A_1 , the second highest function rank. Each remaining function acting on a supersystem component is also of rank A_1 . For all the remaining functions, which are system internal only, the function rank is A_{n+1} if they act on a component, which itself performs a function of rank A_n . Thus, this procedure works stepwise, and eventually attaches the function ranks B, A_1 , ..., A_M in descending order to all of the involved functions. Here, M is a natural number which generally depends on the concrete functional pattern between the components of the system. Finally, function values are assigned to the functions, starting with the lowest function rank: $A_M \rightarrow 1$, $A_{M-1} \rightarrow 2$, ..., until $A_1 \rightarrow M$. Since basic functions are supposed to be especially valuable, they get the function value $B \rightarrow M + 2$.

References

1. Mann, D.: Hands-on Systematic Innovation. 2nd edn. IFR Press (2010).
2. Gadd, K.: TRIZ for Engineers: Enabling Inventive Problem Solving. Wiley (2011).
3. Miles, L.D.: Techniques of Value Analysis and Engineering. 3rd edn. Lawrence D. Miles Value Foundation (2015).

4. Cooke, J.: Improving the value of products and processes by combining value analysis techniques and lean methods with TRIZ. In: *Journal of the European TRIZ Association – INNOVATOR* 1(1), pp. 27–36. ETRIA (2014).
5. Litvin, S., Feygenson, N., Feygenson, O.: Advanced function approach. In: *Procedia Engineering* 9, pp. 92–102. Elsevier (2011). doi:10.1016/j.proeng.2011.03.103.
6. Litvin, S., Ikovenko, S., Lyubomirskiy, A., Stevenson, K.: *MATRIZ - Level 1 Training Manual*. GEN-TRIZ (2019).
7. Devoino, I.G., Koshevoy, O.E., Litvin, S.S., Tsourikov, V.: Computer based system for imaging and analyzing a process system and indicating values of specific design changes. Patent No. US 6,202,043 B1 (2001).
8. Arel, E.T., Verbitsky, M., Devoino, I., Ikovenko, S.: *TechOptimizer Fundamentals*. Invention Machine Corporation (2002).
9. Arel, E.T.: *TechOptimizer 4.0 User Guide*. Invention Machine Corporation (2003).
10. Chechurin, L.S., Wits, W.W., Bakker, H.M., Vaneker, T.H.J.: Introducing Trimming and Function Ranking to SolidWorks Based on Function Analysis. In: *Procedia Engineering* 131, pp. 184–193. Elsevier (2015). doi:10.1016/j.proeng.2015.12.370.
11. Li, M., Ming, X., He, L., Zheng, M., Xu, Z.: A TRIZ-based Trimming method for Patent design around. *Computer-Aided Design* 62, 20–30 (2015). doi:10.1016/j.cad.2014.10.005.
12. Adunka, R.: Function Analysis for Electronic Products. In: *Proceedings of the TRIZ Future Conference 2010*, pp. 165–171. Bergamo University Press (2010).
13. Wessner, J.: Value Analysis as practiced in TRIZ-based Function Analysis with time steps. In: *TRIZ Review: Journal of the International TRIZ Association - MATRIZ* 2(1), pp. 33–43. MATRIZ (2020).
14. Efimov-Soini, N.K., Chechurin, L.S.: Method of Ranking in the Function Model. In: *Procedia CIRP* 39, pp. 22–26. Elsevier (2016). doi:10.1016/j.procir.2016.01.160.