Design of Machines and Structures, Vol. 2, No. 1 (2012), pp. 83–93.
https://doi.org/10.32972/dms.2012.008

THREE DIMENSIONAL SOLUTION SPACE IN DESIGN METHODOLOGY

ÁGNES TAKÁCS
University of Miskolc, Department of Machine and Product Design
3515 Miskolc-Egyetemváros
takacs.agnes@uni-miskolc.hu

Theoretical basis of a method will be introduced in this paper. With the combination of the tools of design methodology this method can be used with great profit during the generation process of the optimal product variant, besides it also takes care for costumers' requirements.

1. Criticising traditional design methodology

A significant disadvantage of existing methods is that their adaptability to computers is limited. Traditional methods suggest several possibilities for the sequential steps realising the given logical step that implies their multiple character. Firstly a fix sequential design method has to be created instead of traditional methods, though utilizing their positive features.

Keeping physical principles in view is the second great disadvantage of traditional methods. It is disadvantageous because the constructor may limit the solution possibilities in a very early phase of the design process. This way he may unwittingly eliminate solutions that could be really valuable in the period of selection, because they are novel or absolutely new concepts. This way the second task is to generate the solution variants, without paying any attention to physical principles.

The third –and maybe the greatest– deficiency of existing methods is that they do not even try to generate all the possible solutions that could be made on the basis of the given functions, because they pay attention to the human capacity and not to the capacity of computers. So it can happen that some –or significantly a great number of – principally correct and absolutely new solutions escape the designer's eyes. That is why the most important task is to make it possible to generate and handle all the solutions that can be generated on the basis of the given functions.

2. Computer Aided methodological Concept Building (CACB)

Analysing other technologies integrated into CAD systems –so the CAxx technologies [2] – in connection with the whole design process it can be defined that aiding of the design process by computer was developed counter direction to the advancement of it (*Figure 1*.). Computer technologies appear first time in the documentation period of the design process that can be taken as the last phase of it, while the computer aid of conceptual design –that can be taken as the basis of the design – is not yet solved even today. Designers have to keep their eyes on the consumer's criteria as well, although these criteria can eliminate some originally new solutions. This way in this paper a method is suggested that combined with the tool system of the design methodology can be used with a great benefit in the period of conceptual design that is during the working out of the basis of the optimal product variant, and it also pays attention for the consumer's criteria.

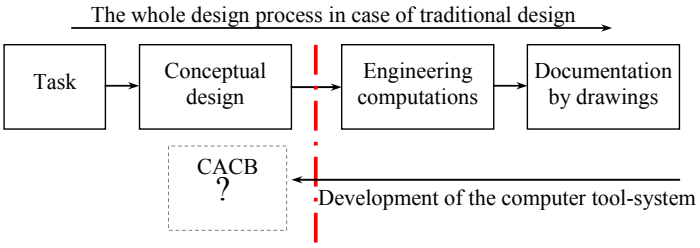


Figure 1. CACB (Computer Aided methodological Concept Building), suggestion for expanding the application of computer tool-system

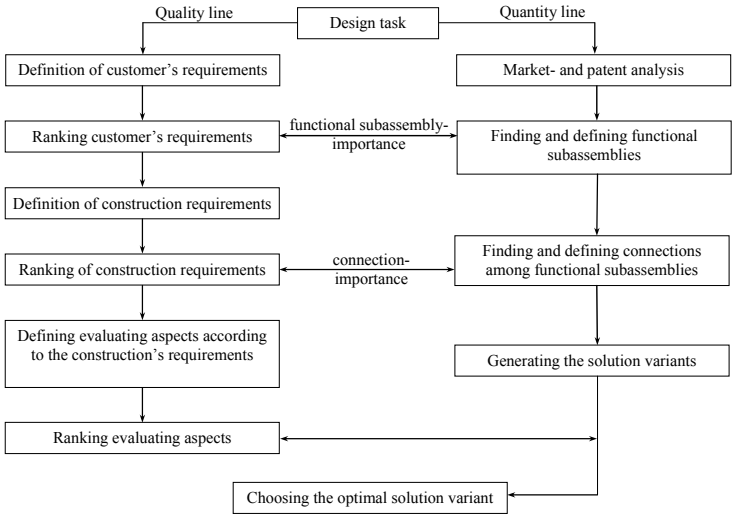


Figure 2. Logical steps of the suggested computer aided concept building

Methods that are suitable for computer application and are developed on the basis of the advantages of the traditional design methods – that are not directed by any computer application – should be created in order to get the conceptual design phase computer aided, and to find the optimal concept for the given design task. Logical steps of the suggested Computer Aided methodological Concept Building (or CACB) is shown in *Figure 2*. Before elaborating the design task it should be analysed. The tools for that are the market research and the analysis of patented solutions. In parallel with this customer's requirements should be found and defined. These requirements should be evaluated and ranked with the designer's eyes, because these requirements are the basis of the evaluation at the end of the concept building method. All the possible functional subassemblies should be defined during the market research and the analysis of patented solutions. Product structures or solution variants can be generated from these subassemblies. These variants should be evaluated by the designer. The optimal solution that is the result of the concept building is the one that fulfilled all the evaluation criteria.

2.1. Customers' requirements

Customers' requirements and claims established to the product are usually ready for the designer at the beginning of the design process in a shape of a claim-list [1]. According to this list criteria can be collected and ranked through the designer's eyes. The basis of this ranking is the fuzzy logic, but it also applies the couple comparison suggested by the traditional design methodology.

Ranking the criteria can be described by the equation (1) that also can be determined by equation (2) that is the **K** criteria matrix. Criteria should be compared in couples. It should be analysed if the criteria in the columns of the matrix are more important than those can be find in the rows of the matrix. The k_{ij} elements of the matrix according to (1) and (2) can get the value 0 and 1. The lower triangular of the matrix is the complement of the upper triangular.

$$\begin{array}{c} \\ K_1 \\ K_2 \\ K_3 \\ \dots \\ K_n \end{array} \begin{array}{c} \begin{array}{ccccccc} K_1 & K_2 & K_3 & \dots & K_n & \Sigma \end{array} \\ \begin{bmatrix} 0 & k_{12} & k_{13} & \dots & k_{1n} \\ k_{21} & 0 & k_{23} & \dots & k_{2n} \\ k_{31} & k_{32} & 0 & \dots & k_{3n} \\ \dots & \dots & \dots & 0 & \dots \\ k_{n1} & k_{n2} & k_{n3} & \dots & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ k_{1f} \\ k_{2f} \\ k_{3f} \\ \dots \\ k_{nf} \end{array} \tag{1}$$

$$K := \left(k_{ij}\right)_{n \times n} \tag{2}$$

$$1 - k_{ij} = k_{ji} \tag{3}$$

Important features of the criteria matrix are:

- the major diagonal of the criteria matrix can contain only zeros,
- if the elements above the major diagonal is summarised their pairs under the major diagonal, the result is always 1 (3).

According to that it is also enough to have the elements above the major diagonal; the value of the elements under the major diagonal can be calculated. So the number of the elements that is enough to determine a similar square matrix can be described by a mathematical row according to (4).

$$\begin{aligned} z = {} & n \cdot (n-1) - [(n-1) \cdot (n-1)] + [(n-2) \cdot (n-2)] - \dots \\ & \dots + [(n-i) \cdot (n-i)] - \dots + \dots [(n-n) \cdot (n-n)] \end{aligned} \tag{4}$$

Summarising row-by-row the results of the criteria comparison according to equation (5) effects the importance of the criteria. So a (6) $\underline{k_f}$ criteria importance vector arises that should be considered when defining criteria.

$$k_{if} = \sum_{j=1}^{n} k_{ij} \tag{5}$$

$$\underline{k_f} = \begin{bmatrix} k_{1f} & k_{2f} & k_{3f} & \dots & k_{nf} \end{bmatrix} \tag{6}$$

2.2. Generating solution variants

Functional subassemblies that are important during the generation of new solution variants and can be found in the known solutions should be disclosed by the patent and market analysis. Sometimes new functional subassemblies should be determined by intuition. Gen-

erating variants can be realised in several way – e.g.: according to binary logic or random generation. When the designer would like to perform a preliminary selection with analysing the functional subassemblies ranking according to equation (1) and (2) should be repeated with the functional subassemblies:

$$\begin{array}{c} \\ F_1 \\ F_2 \\ F_3 \\ \dots \\ F_n \end{array} \begin{array}{c} \begin{array}{ccccc} F_1 & F_2 & F_3 & \dots & F_n \end{array} \Sigma \\ \left[\begin{array}{ccccc} 0 & f_{12} & f_{13} & \dots & f_{1n} \\ f_{21} & 0 & f_{23} & \dots & f_{2n} \\ f_{31} & f_{32} & 0 & \dots & f_{3n} \\ \dots & \dots & \dots & 0 & \dots \\ f_{n1} & f_{n2} & f_{n3} & \dots & 0 \end{array}\right] \begin{array}{c} f_{1f} \\ f_{2f} \\ f_{3f} \\ \dots \\ f_{nf} \end{array} \end{array} \tag{7}$$

$$F := \left(f_{ij}\right)_{n\times n} \tag{8}$$

$$1 - f_{ij} = f_{ji} \tag{9}$$

Summarising the values in each row of the matrix it eventuates the importance of the functional subassemblies (10). This can be described by the (11) importance vector of the functional subassemblies.

$$f_{if} = \sum_{j=1}^{n} f_{ij} \tag{10}$$

$$\underline{f_f} = \left\lfloor f_{1f} \quad f_{2f} \quad f_{3f} \quad \dots \quad f_{nf} \right\rfloor \tag{11}$$

Than the rank of each functional subassembly should be given on an open interval, between 0 and 1, so each of them has a fuzzy value.

After this the designer also determines the solution level with a value between 0 and 1. Determining the solution level means that we should withdraw those subassemblies that has a smaller rank than a given value from the set of those functional subassemblies that are building up solutions. This way the designer eliminates several solution-possibilities, but those functional subassemblies that are the most valuable in technical meaning assist generating significant solutions. The value of the solution level is the closer to zero, the more functional subassembly constitutes the generated solution. So giving a value to each solution level is not showing the quality of the solutions that can be generated, but results a quantity selection. This selection can be shortening the period of the generating. Raising the value of the solution level the complexity of the product can be decreased, so the costs of the product can be also decreased. But this way it is not sure that the most important costumers' criteria will be fulfilled.

Than the connections between functional subassemblies constituting variants should be defined, so it should be controlled which functional subassembly can be connected to which one. According to this the **C** connection matrix (12), (13) can be determined.

$$\begin{array}{c} \quad \mathrm{C}_1 \quad \mathrm{C}_2 \quad \mathrm{C}_3 \quad \ldots \quad \mathrm{C}_\mathrm{n} \\ \begin{array}{c} C_1 \\ C_2 \\ C_3 \\ \ldots \\ C_n \end{array} \begin{bmatrix} 0 & c_{12} & c_{13} & \ldots & c_{1n} \\ c_{21} & 0 & c_{23} & \ldots & c_{2n} \\ c_{31} & c_{32} & 0 & \ldots & c_{3n} \\ \ldots & \ldots & \ldots & 0 & \ldots \\ c_{n1} & c_{n2} & c_{n3} & \ldots & 0 \end{bmatrix} \end{array} \tag{12}$$

$$C := \left(c_{ij}\right)_{n \times n} \tag{13}$$

The components of the matrix can be 0, 1 and x letters. These characters describe if there is connection or not between the given functional subassemblies in the structure. If the connection is defined by x, than during the generation the program has the right to make a decision whether the connection takes part in the structure or not. This way during the generation each solution has another connection matrix, so each solution can be defined by an **S** structure matrix (14), (15), that has absolutely binary structure, consisting only zeros and ones.

$$\begin{array}{c} \quad \mathrm{S}_1 \quad \mathrm{S}_2 \quad \mathrm{S}_3 \quad \ldots \quad \mathrm{S}_\mathrm{n} \\ \begin{array}{c} S_1 \\ S_2 \\ S_3 \\ \ldots \\ S_n \end{array} \begin{bmatrix} 0 & s_{12} & s_{13} & \ldots & s_{1n} \\ s_{21} & 0 & s_{23} & \ldots & s_{2n} \\ s_{31} & s_{32} & 0 & \ldots & s_{3n} \\ \ldots & \ldots & \ldots & 0 & \ldots \\ s_{n1} & s_{n2} & s_{n3} & \ldots & 0 \end{bmatrix} \end{array} \tag{14}$$

$$S := \left(s_{ij}\right)_{n \times n} \tag{15}$$

Each structure can be characterised by a structure graph as well. These graphs define the connections among the different functional subassemblies of solution variants. Nodes of the graphs mean the functional subassemblies; the edges of the graphs mean the connections among the functional subassemblies. Structures can also be defined by structure equations that describe the edges of the structure graphs, or those connections that are writing down the structure of each solution variant. Synthesis of a structure equation is very simple: it contains the connections between functional subassemblies denoted by 1 in the structure matrix. For example:

$$F_1 - F_2, F_1 - F_i, F_j - F_i, F_j - F_n \cdot \tag{16}$$

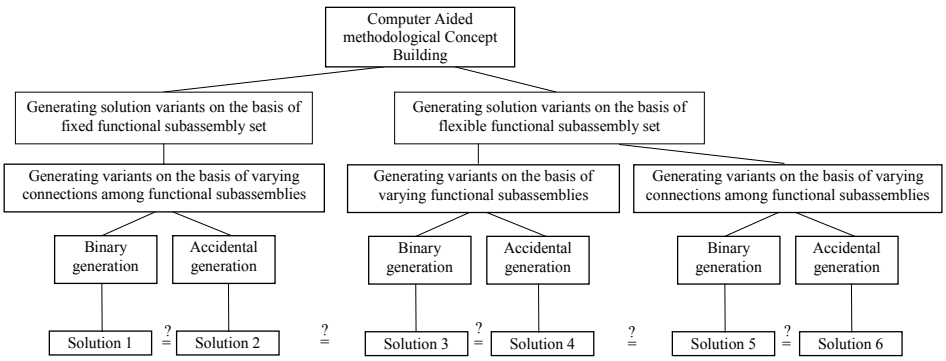


Figure 3. Generating solution variants

Generating variants can be realised basically in two different ways as it is shown in *Figure 3.;* on the basis of the fixed functional subassembly set, or the flexible functional subassembly set. There are cases when giving the importance of the functional subassemblies can be eliminated. In this case the set of functional subassemblies is not changing, so the generating happens according to the fixed functional subassembly set. When the set of functional subassemblies is decreased by giving the importance of functional subassemblies and determining the solution level, the size of the functional subassembly set changes, so the generating happens according to the flexible functional subassembly set.

Exploring the optimal field of use and analysing the efficiency of the mentioned methods needs more examination. In case of some products – *e.g.: rear-view mirror of a car*– by the definition of a low solution level all of the functional subassemblies can take part in the product structure, but at a higher solution level only those functional subassemblies are missing that has not effect on the basic operation of the product. In case of these products the hardness of the solution level does not influence the operation. But there are those kinds of products as well – *e.g.: solutions variants deriving from the connection of different kinds of planetary gears*–, where all the functional subassemblies are necessary to generate operable solutions. That is the reason for the generating solution variants according to varying connections between the functional subassemblies can be called total combination, because if the connections between the functional subassemblies are preliminary not defined, all the possible solutions can be generated by binary generation. It is not sure that according to the theory of random numbers all the possible solutions can be generated.

The method worked out basically deals with two different theories: solution variants generated by varying the functional subassemblies and solution variants generated by varying the connections among the functional subassemblies. The research shows two mathematical solutions for both theories: the binary logics, and the generation of random numbers. As it is shown in Figure 3., the different generation theories can show different results. These methods do not eliminate the possibility that the results can fall in with each other.

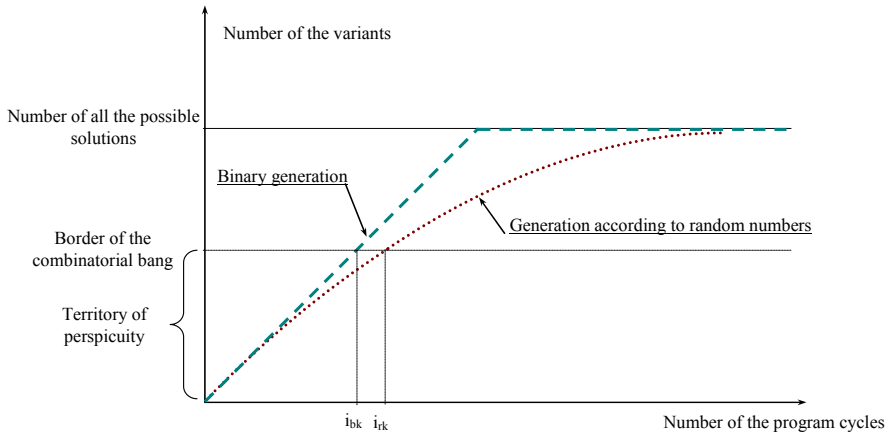


Figure 4. The maximum number of solution variants according to the binary logics and the theory of random numbers

There is no need for defining the connections between the functional subassemblies in case of generation according to random numbers, because than the main point is that the computer randomly determines solutions from all the possibilities. These solutions should be examined by a strict value analysis, if they are operable solutions. According to binary generation each functional subassembly can be connected to each of them so it is sure that all the possible solution variants can be created. But the user of the program can also fix connections, than the generation is resulted less solution. Generating according to different theories can bring different solutions. Efficiency-analysis of the methods needs more examinations.

Figure 4. shows the expectable number of solutions generated according to binary generation and the theory of random numbers depending on the number of the generating cycles, besides the computer assistance of the suggested methods. As it can be seen on the figure all the possible solution variants can be created by binary generation even besides finite cycle number. The curve of the generating according to the theory of random numbers only fitting the horizontal line of all the possible solutions in case of endless cycle number. Depending on the input data of solution generating it can happen that the border of the combinatorial bang realises at a lower number of solutions, than the number of all the possible solutions. In that case the set of solutions will be inhomogeneous, because of the lack of those solutions that were generated after reaching the border of the combinatorial bang. So if it happens it is better to use the generation according to the theory of random numbers, because solutions generated up to the borders of the combinatorial bang are from the full set of the generation that is why the set of the solutions will be homogeneous. Generating the same number of solutions i_{rk} is always higher than i_{bk}, so in the case of the generation according to the theory of random numbers more program cycle is needed than according to binary generation.

2.3. Evaluating step

In the evaluating step the generated solutions should be evaluated. During this process the **E** value matrix (17) and (18) can be created. This matrix ranks the evaluating aspects according to the consumers' criteria. Summarising row-by-row the result of the comparison of the evaluating aspects it eventuates the importance of the evaluating aspect (20). This can be described by the (21) importance vector of the evaluating aspects.

$$E := \left(e_{ij}\right)_{n\times n} \tag{17}$$

$$\begin{array}{c} \\ E_1 \\ E_2 \\ E_3 \\ \dots \\ E_n \end{array} \begin{array}{c} \begin{array}{cccccc} E_1 & E_2 & E_3 & \dots & E_n & \Sigma \end{array} \\ \begin{bmatrix} 0 & e_{12} & e_{13} & \dots & e_{1n} \\ e_{21} & 0 & e_{23} & \dots & e_{2n} \\ e_{31} & e_{32} & 0 & \dots & e_{3n} \\ \dots & \dots & \dots & 0 & \dots \\ e_{n1} & e_{n2} & e_{n3} & \dots & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ e_{1f} \\ e_{2f} \\ e_{3f} \\ \dots \\ e_{nf} \end{array} \tag{18}$$

$$1 - e_{ij} = e_{ji} \tag{19}$$

$$e_{if} = \sum_{j=1}^{n} e_{ij} \tag{20}$$

$$\underline{e_f} = \begin{bmatrix} e_{1f} & e_{2f} & e_{3f} & \dots & e_{nf} \end{bmatrix} \tag{21}$$

The ranked evaluating aspects should be characterised by a weighting factor. After that using the weighted product characters method a table according to *Table 1*. should be defined. Each variants should be evaluated with points between 1 and 5 (*the worst is 1–means* p_{min}*, the best is 5- means* p_{max}), than should be multiplied with the given weighting factor. Than the weighted values should be summarised in each row, so the rank of the evaluated solutions denoted by $\underline{q}$ vector is resulted. This vector can be found in the last column of the table. The q_i element of the $\underline{q}$ vector that has the highest value gives the optimal solution (22).

Table 1.

The method of weighted product characters

	A_1	A_2	A_3	…	A_m	Σ	q
V_1	$p_{min} \div p_{max}$	$p_{min} \div p_{max}$	$p_{min} \div p_{max}$		$p_{min} \div p_{max}$		q_1
wV_1	w_1V_1	w_2V_1	w_3V_1		w_nV_1	ΣwV_1	
V_2	$p_{min} \div p_{max}$	$p_{min} \div p_{max}$	$p_{min} \div p_{max}$		$p_{min} \div p_{max}$		q_2
wV_2	w_1V_2	w_2V_2	w_3V_2		w_nV_2	ΣwV_2	
V_3	$p_{min} \div p_{max}$	$p_{min} \div p_{max}$	$p_{min} \div p_{max}$		$p_{min} \div p_{max}$		q_3
wV_3	w_1V_3	w_2V_3	w_3V_3		w_nV_3	ΣwV_3	
…							
V_n	$p_{min} \div p_{max}$	$p_{min} \div p_{max}$	$p_{min} \div p_{max}$		$p_{min} \div p_{max}$		q_n
wV_n	w_1V_n	w_2V_n	w_3V_n		w_nV_n	ΣwV_n	

$$V_{opt} = \max \underline{q} \tag{22}$$

3. Representing the fuzzy solution-family in three-dimensional space

The fuzzy solution-family consists of those solutions that were correct solutions of the generating according to the flexible subassembly set. In Descartes coordinate-system the space determined by the V elements of the solution-family, the F functional subassemblies used during the generation and the importance of the functional subassemblies is the Ω fuzzy solution space (*Figure 5*.). The Ω fuzzy solution space can be described by a **B** product complexity matrix and an $\underline{f}_f$ functional subassembly importance vector.

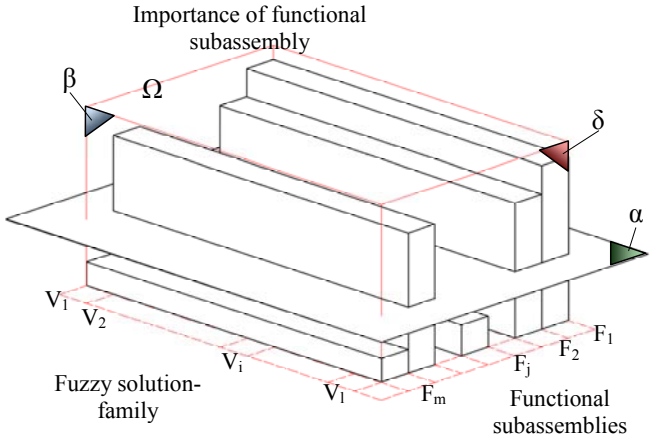


Figure 5. The importance of functional subassemblies, the complexity of solutions

$$B := \left(b_{ij}\right)_{n\times m} \tag{23}$$

$$\begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ \dots \\ V_n \end{array} \begin{array}{c} \begin{array}{ccccc} F_1 & F_2 & F_3 & \dots & F_m \end{array} \\ \begin{bmatrix} b_{11} & b_{12} & b_{13} & \dots & b_{1m} \\ b_{21} & b_{22} & b_{23} & \dots & b_{2m} \\ b_{31} & b_{32} & b_{33} & \dots & b_{3m} \\ \dots & \dots & \dots & \dots & \dots \\ b_{n1} & b_{n2} & b_{n3} & \dots & b_{nm} \end{bmatrix} \end{array} \tag{24}$$

The Ω solution space can be sectioned by 3 excellence planes. Either of the planes parallel with the plane determined by the solutions and the functional subassemblies are the α fuzzy product complexity planes. It shows which functional subassemblies take part in a solution at different level of the importance of functional subassemblies. Those functional subassemblies that has a higher importance than the α product complexity level, takes part in the solutions. Moving the plane in vertical direction the complexity of products can be controlled. Different (α) levels of product complexity can be featured by the product complexity matrix.

The Ω solution space sectioned by a plane that is perpendicular to the axis of solutions the β fuzzy functional subassembly importance plane arisen. This plane shows how impor-

tant the functional subassemblies that are building up the solutions in certain solutions. The Ω solution space sectioned by the β plane shows a "*prismatic*" nature, because it shows the same frame in case of each solution. Differences between the solutions come from the different connections between the functional subassemblies written by the **S** structure matrixes. The β plane of the functional subassembly importance can be characterised by the (11) $\underline{f}_f$ functional subassembly importance vector.

The Ω solution space sectioned by a plane that is perpendicular to the axis of functional subassemblies the δ fuzzy functional subassembly incidence plane is offered. It shows whether the certain functional subassemblies are in the set of those elements that are building up the fuzzy solution family and if it is so, how important they are. The δ plane can be featured by any element of the $\underline{f}_f$ functional subassembly importance vector according to (11).

The relationship of the introduced planes can be defined according to the followings:

- The intersections defined by the planes α and β determine, which functional subassemblies are building up a given solution of the solution-family.
- The intersection can be defined by a vector that has a value-range of 0 and 1.
- Intersections determined by the planes that are parallel with α–δ planes show the importance of those functional subassemblies that are the members of the solutions in a solution-family. The intersection-line is a certain element of the $\underline{f}_f$ functional subassembly importance vector according to (11).
- Intersections determined by the planes that are parallel with β–δ planes show the incidence of those functional subassemblies that are the members of the solutions in a solution-family at a certain level of importance, so at the level of α solution level. The intersection-line can be described by a vector with each element the same: 0 or 1.

The point of intersections determined by the above introduced three planes (*fuzzy product complexity plane, fuzzy functional subassembly importance plane and fuzzy functional subassembly incidence plane*) is the importance point of a functional subassembly, as it defines the importance of the given functional subassembly. The Ω solution space has a constant vertical size, and its maximum value is 1. But the size of the Ω solution space can be modified in the direction of V and F that can be influenced by the following factors:

- Modifying the fuzzy product complexity level the number of the functional subassemblies taking part in the solution generation can be influenced. This has an effect on the number of the solution variants.
- The number of the solution variants also depends on the values of the connection matrix.

4. Adaptability of the suggested methods for computer

It must be noticed that the introduced methods generating solution variant are able to operate effectively only then, when the application of a computer is solved. These methods make a suggestion for how to generate the more possible solution built up from the functional subassemblies, determined at beginning of the process. It is only the binary method of generating variants on the basis of varying connections among functional subassemblies capable for introducing all the possible solution variants. It was interpreted that in case of a big amount of variable connections it is better to use the method of accidental generation, because this way an overall view of the whole solution-space is given. Binary generation is

eligible to use, when the number of all the possible solution probably not overruns the Territory of perspicuity. As a summary it can be established that the introduced methods show a possibility for an algorithm of the design methodology, beat the human limits, shorten the design period, proving the probable quality of the technical plan.

Acknowledgement

"This research was carried out as part of the TAMOP-4.2.1.B-10/2/KONV-2010-0001 project with support by the European Union, co-financed by the European Social Fund."

REFERENCES

[1] E. R. Deciu–E. Ostrosi–M. Ferney–M. Gheorghe: *Configurable product design using multiple fuzzy models.* Journal of Engineering Design, Vol. 16., No. 2., Taylor and Francis, 2005.
[2] I. Horváth–I. Juhász: *Számítógéppel segített gépészeti tervezés*. Műszaki Könyvkiadó, Budapest, 1996.