7th International Conference on Mass Customization
and Personalization in Central Europe (MCP-CE 2016)

Mass Customization and Open Innovation
September 21-23, 2016, Novi Sad, Serbia

FINDING A SUITABLE DEGREE OF CUSTOMIZATION FOR PRODUCT PLATFORMS

Slavomir Bednar, Vladimir Modrak
Technical University of Kosice, Faculty of Manufacturing Technologies, Presov, Slovakia

Abstract: *An important part of product variety management in mass customization environment is finding optimum variety extent. The problem appears as crucial when product configuration conflict problems occur. This paper proposes a method to decide about more suitable degree of customization for existing or newly proposed product design platforms. Finally, the case application is described in order to demonstrate applicability of the method. The newly developed method can be employed to assist product managers to independently assess competitive product variety platforms against each other and to evaluate their customization characteristics.*

Key Words: ***Customization, Platform, Complexity, Metric, Design, Product***

1. INTRODUCTION

An important part of product variety management is finding optimum variety extent based on product design architecture. Normally, variety extent is limited by production possibilities. Methods to identify and solve configuration conflicts are known also as a constraint satisfaction problems (CSPs). Constraint satisfaction problems as mathematical-based methods of operations research are quite common for their potential use also in product variety management. In principle, constraint satisfaction methods can be effectively used in many sectors. Nevertheless, their implementation in configurators for mass customization (MC) of products needs adoption of specific requirements.

Usually, extent of customizable products is perceived in a sense that the bigger the product variety, the better, and vice-versa [1-3]. Obviously, high variety extent directly relates to so called variety-induced complexity that may result in possible turbulences in the manufacturing systems, leading e.g. to higher direct production costs. But, extent of product varieties in MC environment is becoming serious problem when product configuration conflicts appear. Then, product designers have to consider also such constraints, since it can cause some serious problems. Especially, disappointments may occur when requirements of the customer are specified based on a wide portfolio of modules or components and not all configurations can be satisfied due to restrictions on selected components and their combinations.

The main scope of this paper is to explore the possibilities to solve this issue by changing a rate between infeasible product configurations and all possible product configurations when restrictions are omitted. In a simple way numbers of product configurations (NPC) are closely related to variety-induced complexity. However, numbers of product configurations, both, viable as well as unviable are not optimal indicators of variety-induced complexity to be used to solve this problem and to express the rate as NPC indicator does not reflect assembly component composition and may provide similar variety-induced complexity. Therefore instead, an entropy-based complexity metrics will be used as a tool for decision-making in variety management. Finally, in this paper, a decision-making algorithm to solve the issues related to optimal selection of product component platform will be presented on a real product platform.

2. ENTROPY-BASED COMPLEXITY METRICS FOR PRODUCT VARIETY MANAGEMENT

2.1. Theoretical background

The aim of this sub-section is to analyses relation between infeasible product configurations and all possible product configurations of any existing or intended product platform.

The very first notion of complexity was outlined in the work of Shannon [4] where information theory was originally developed. Few years later, information became a key complexity element for the description and analysis of the systems and information entropy. Its definition for the discrete case has been defined by the probability P_i of the n-state occurrence as follows [4]:

$$H_d = \sum_{i=1}^{n} P_i \; log_2 \; P_i . \tag{1}$$

Differential information entropy of the probability density function *p(x)* for continuous signals has been expressed as:

$$H_c = \int_{-\infty}^{\infty} p(x) log_2 \, (p(x)) dx . \tag{2}$$

Krus [5] adopted design information entropy for multidimensional case in the following form:

$$H_x = \int_{-\infty} p(x) log_2 (p(x)S) dx, \tag{3}$$

where D is the design space within a particular design x. Subsequently, S is the size of the design space and can be expressed as:

$$S = \int_D x dx. \tag{4}$$

In case of the general multivariable, information entropy of a design can be expressed as:

$$H = log_2 \frac{S}{s}, \tag{5}$$

where s is the region of uncertainty for the final design of a validated system architecture.

According to [5], each particular design x with regards to its design space has information entropy H_x:

$$H_x = log_2 n_s, \tag{6}$$

where n_s is a number of unique design alternatives (representing so called complete design space) that are results of a combination of product options and H_x is denoted as Entropy of complete design space.

There are many real cases, in which some product variants or configurations are impractical due to presence of constraint(s). Then, information entropy of constrained design space H_c can be enumerated as:

$$H_c = log_2 n_v, \tag{7}$$

where n_v is a number of viable design alternatives.

As higher number of all possible design variants has more positive impact on consumers than smaller constrained design space, then Entropy of constrained design space in terms of MC environment should be maximized. In this sense, Entropy of constrained design space can be considered as positive entropy.

In this context, Krus [5] proposed to express a quality of a modular design/platform through the rest of the design space that is outside the constrained design space by term "waste" information entropy of design space and to quantify it using the formula:

$$H_W = H_X - H_C. \tag{8}$$

In line with the logic used for the Entropy of constrained design space, Waste entropy can be considered as negative entropy. Once the background of the Waste entropy is outlined, we may proceed towards its application.

In order to catch the effect of product design optimization by using the concept of negative entropy, we firstly need to generate concurrent product design architectures to be mutually benchmarked. One way to do so is through a gradual execution of selected components from the original product design architecture. Subsequently, mutual relation between so called positive entropy and negative entropy can be treated. For this purpose, we will need to enumerate numbers of all possible product configurations when restrictions are omitted and all possible product configurations with component restrictions. This procedure is presented in the following sub-section.

2.2. Case I. on enumeration of waste entropy for concurrent product design architectures

To show a practicability of the approach, a realistic case is provided to motivate practitioners to solve similar problems. For this reason, an assembly model of personal computer adopted from [6] in the form of selection algorithm has been used to identify product configurations, as seen in Fig. 1.

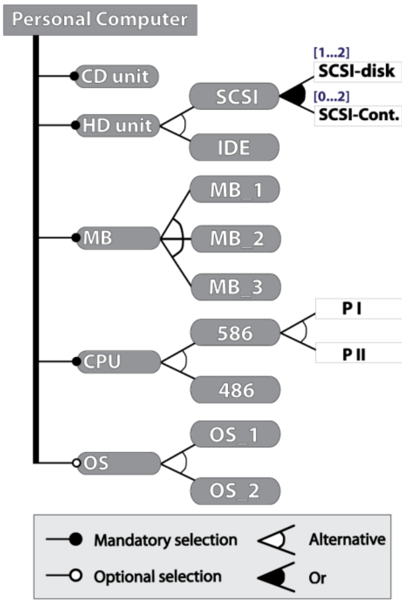


Fig. 1. Case of product structure with restrictions for constraint satisfaction problem

Once managers are in the early stage of product architecture design, they might decide about the most suitable product component (module) structure. Normally, marketing managers strive to maximize the variety offer with aim to satisfy a wide range of customers knowing also that some incompatible components can occur in possible product configurations. The problem is that they are not aware of the number of infeasible product configurations when designing a product platform. Moreover, it is not easy to identify those using amateur methods as it will be shown further. On every fall, relatively high number of such infeasible product configurations, as a rule, negatively affects customer perception and buying behaviour.

With regards to component restrictions, there are different reasons for restriction or obligation between two or more components. There may be functional, design, connectivity or other reasons for relation or for execution of the link between any two or more components. Besides the structural, hierarchy or aggregation restrictions, four types of configuration rules may arise [6]: a) require rule, b) incompatible rule, c) port-connection rule, and d) resource balancing rule.

Model depicted in Fig. 1 is a representation of MC assembly of a personal computer consisting of five basic modules: CD-unit (1 option), HD-unit (6 individual customer option), Motherboard (MB) (3 options as MB_1, MB_2 and MB_3), CPU (586_P I, 586_P II, 486), and Server Operating system (OS) (OS_1 and OS_2). The case mode 1 has various customizable options depending on the customers' choice but with predefined restrictions in the form of rules related to incompatibility of components defined as follows:

R#1 – *CPU3* must not be in the same configuration with component *MB1*.

R#2 – *MB_2* must not be in the same configuration with components *CPU1* and *CPU2*.

R#3– *CPU3* must not be in the same configuration with component *MB_3*.

R#4 – *OS1* must not be in the same configuration with component *MB_1* and *MB_3*.

R#5 – *OS2* must not be in the same configuration with component *MB_2* and *MB_3*.

R#6 – *MB_2* and *MB_3* must not be in the same configuration with components *HD4*, *HD5* and *HD6*.

R#7 – *OS2* must not be in the same configuration with components *HD2* and *HD4*.

2.3. Enumeration of product configurations with and without component restrictions

At the beginning, it is useful to transform the computer structure with constraints shown in Fig. 1 into a simplified assembly graph depicted in Fig. 2. Adopting previously developed model as in [7, 8], any such structure usually consists of number of assembly stations – nodes. These can be identified within a multi-level network. In our case, two-level network is sufficient to model final assembly operation of personal computer. Additionally, specific number of component alternatives can be identified at each node of tier t_1.

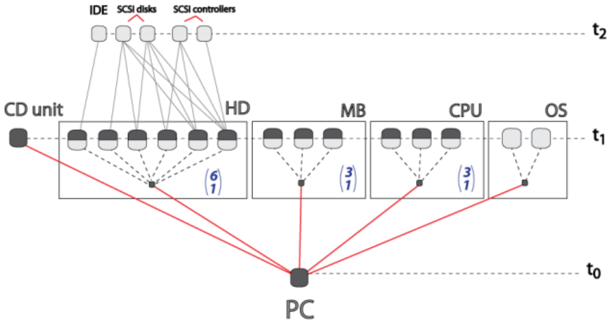


Fig. 2. Assembly graph of personal computer without component restrictions

As evident from Fig. 2, HD unit is represented by six alternatives. Number of all possible component permutations is seven but one of them is omitted, namely permutation consisting of two SCSI-Controllers with single SCSI disk, as the second controller in such combination is considered to be redundant.

Then, on the bottom tier t_0, all possible product configurations without restrictions can be identified for the original product design platform (D_0):

$$\sum Conf_{D0} = 1 * 6 * 3 * 3 * 2 = 108. \qquad (9)$$

Subsequently, it is necessary to determine the total number of configurations when restriction rules *R*#1-7 are considered. For this purpose, incidence matrix with component restrictions *R*#1-7 is constructed (see Table 1).

Table 1. *Matrix of component restrictions(R#1-7)*

		Group 1 - HD						Group 2 - CPU			Group 3 - MB			Group 4 - O	
		HD1	*HD2*	*HD3*	*HD4*	*HD5*	*HD6*	*CPU1*	*CPU2*	*CPU3*	*MB1*	*MB2*	*MB3*	*OS1*	*OS2*
Group 1 - HD	*HD1*	■	▨	▨	▨	▨	▨								
	HD2	▨	■	▨	▨	▨	▨								❼
	HD3	▨	▨	■	▨	▨	▨								
	HD4	▨	▨	▨	■	▨	▨					❻	❻		❼
	HD5	▨	▨	▨	▨	■	▨					❻	❻		
	HD6	▨	▨	▨	▨	▨	■					❻	❻		
Group 2 - CPU	*CPU1*	▨	▨	▨	▨	▨	▨	■	▨	▨		❷			
	CPU2	▨	▨	▨	▨	▨	▨	▨	■	▨		❷			
	CPU3	▨	▨	▨	▨	▨	▨	▨	▨	■	❶		❸		
Group 3 - MB	*MB1*	▨	▨	▨	▨	▨	▨	▨	▨	▨	■	▨	▨	❹	
	MB2	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	■	▨		❺
	MB3	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	■	❹	❺
Group 4 - OS	*OS1*	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	■	▨
	OS2	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	■

To enumerate number of restricted (viable) product configurations, the following procedure is proposed. In the first step, let us select e.g. group of HD units. Then, we select arbitrary configuration from the group, for example HD_2, which is one of the six HD unit options. Afterwards, we may construct an incidence sub-matrix for the HD_2 option and group of CPU components. Because there is no restriction, HD_2 as option can be combined with any CPU component (see Fig. 3).

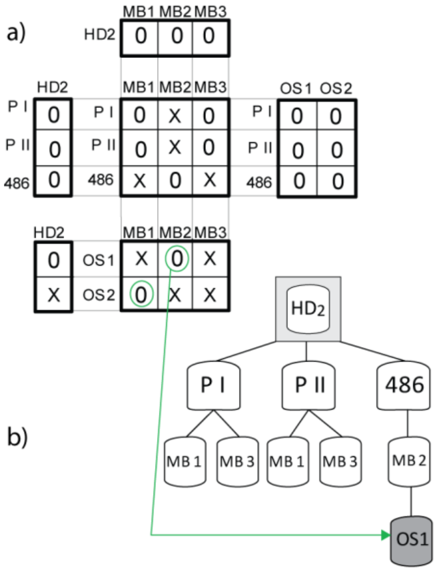


Fig. 3. Proposed approach to transform incidence matrix (a) into a product configurations model (b)

Then, a three dimensional matrix of relations between configurations *HD₂*, group of CPU components and a group of Motherboard components needs to be created. Four restrictions are identified and accordingly CPU components can be combined with compatible *MBs*. Finally, four dimensional matrix relations are constructed in Fig. 3(a) and then it is possible to exactly determine five viable product configurations where HD_2 is exclusively involved. Moreover, this procedure allows generating product component structure of all identified restricted (viable) product configurations, as can be seen in Fig. 3(b).

The sub-procedure depicted in the Fig. 3 has to be repeated for the rest of the components from the Group 1. Then, the sum of only viable configurations for individual components of the Group 1 is 21, as can be seen from Fig. 4.

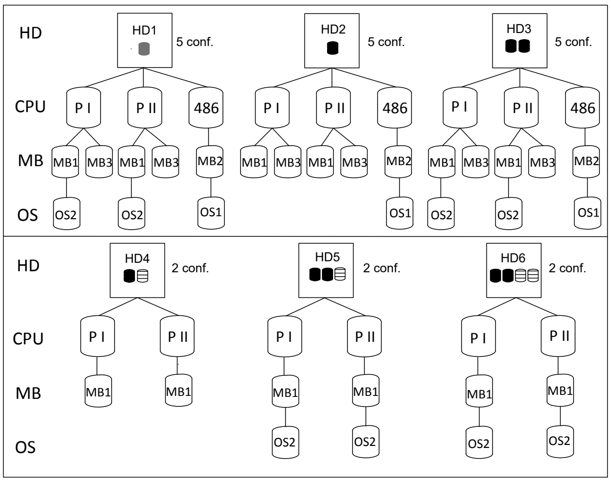


Fig. 4. Model of 21 viable product configurations respecting configuration rules of plaform D_0

2.4. Proposed procedure to reduce waste entropy

As mentioned in the introduction, the goal of the CSP solutions in terms of MC is to reduce number of infeasible configurations. One possible way to reach this goal is by changing the rate between infeasible product configurations and all possible product configurations when restrictions are omitted. This rate can be changed through an execution components linked to restriction from an original product design platform D_0.

For this reason, a new product design platform D_1 can be obtained when e.g. one of motherboards, namely MB_2 is selected for execution. From here on, configurations with MB_2 included are not counted and therefore the total number of model configurations decreased to 72 without accepting the rules and restrictions *R#1-7*. The number was reached by the following multiplications:

$$\sum Conf_{D1} = 1 * 6 * 2 * 3 * 2 = 72. \tag{10}$$

Then, applying the procedure proposed in the Fig. 3, the number of viable product configurations will decrease to 18, as enumerated by the following formula:

$$\sum Res\_Conf_{D1} = 5 + 5 + 5 + 2 + 2 + 2 = 18. \tag{11}$$

To obtain another alternative product design platform D_2 for benchmarking purposes, we eliminate another component CPU_3. Then, the number of total model configurations is calculated as follows:

$$\sum Conf_{D2} = 1 * 6 * 2 * 2 * 2 = 48. \tag{12}$$

Then, viable product configurations will also equal 18, as in Equation (11).

Obtained numbers of configurations with and without restrictions are summarily depicted in Table 2.

Table 2. *Computational results of numbers of product configurations*

Product platforms	**Number of product configurations**	
	Without restrictions n_s (complete design space)	With restrictions n_v (constrained design space)
D_0	108	21
D_1	72	18
D_2	48	18

Subsequently, waste entropy and waste entropy rates for each of the design platforms $D_{0\text{-}2}$ can be calculated. Table 2 above shows how waste configuration ratio is changed by reducing number of restricted components. Both, the reductions (from D_0 to D_1 and from D_1 to D_2) seem to be favourable in order to reduce number of waste (infeasible) product configurations. In such cases, decision-makers may have a dilemma on what design platform is optimal from the customer's perspective. For this purpose, the following decision-making tool to eliminate this dilemma is proposed.

3. DECISION-MAKING ALGORITHM

In this section, we describe the decision-making procedure to select optimal platform of product variants by using mutual relations between waste entropy H_w and constrained design space H_c.

We start by taking so called draft design platform D_0, representing an existing product design platform generating both, feasible and unfeasible product configurations for customers, where n_{s0} presents a number of unique product design configurations as results of a combination of product components and n_{v0} is a number of feasible product design configurations.

Let us further assume that we remove single component from the platform D_0, which is in conflict with other component(s). Then, D_0 can be transformed into a new state with n_{s_1} for all unique product design configurations and n_{v_1} for feasible product configurations, denoted as platform D_1.

If we would continue in such a reduction of components, the design platform D_1 is modified into D_2. Obviously, we may continue in the reduction of system component depending on specific conditions.

To compare exactly two arbitrary design platforms against each other, e.g. D_0 and D_1, the following two measures are proposed:

$$\Delta H_{w_{0,1}} = \left|\frac{H_{w_1}}{H_{w_0}} - 1\right|, \quad (13)$$

$$\Delta H_{w_{0,1}} = \left|\frac{H_{c_1}}{H_{c_0}} - 1\right|. \quad (14)$$

Then, if $\Delta H_{w0,1} > \Delta H_{c0,1}$ => design platform D_1 is more preferable for mass customization (MC) than D_0. To compare between three alternative design platforms, the following sub-procedure can be used. Let us suppose that design platforms D_1 and D_2 are more preferable for MC than D_0, based on criteria:

$\Delta H_{w0,1} > \Delta H_{c0,1}$,

$\Delta H_{w0,2} > \Delta H_{c0,1}$.

Then, one can select more preferable design platform between D_1 and D_2 using these three criteria:

I. If $\Delta H_{w0,1} - \Delta H_{c0,1} > \Delta H_{w0,2} - \Delta H_{c0,2}$ => design platform D_1, is more suitable than D_2.

II. If $\Delta H_{w0,1} - \Delta H_{c0,1} < \Delta H_{w0,2} - \Delta H_{c0,2}$ => design platform D_2, is more suitable than D_1.

III. If $\Delta H_{w0,1} - \Delta H_{c0,1} = \Delta H_{w0,2} - \Delta H_{c0,2}$ => both design platforms D_1, and D_2 are equally preferable for buyers.

Subsequently, proposed procedure for selection of optimal design platform is graphically depicted in Fig. 5 in the form of algorithm.

Analogically, a procedure to select optimal product design platform for the consideration of the three or more platforms at once can be developed.

4. PRACTICAL CASE II. - APPLICATION

In order to proof the relevance of the proposed decision-making tool to select the most optimal product design platform, the following realistic case from the Shimano [9] product compatibility catalogue is used.

The case application in this section is represented by restrictions between the two inter-operating component modules of the drive train, which can be found in every bicycle model. The starting platform D_0 consists of 12 groups (nine for gears and three for chain stay angle (CSA)). Each of the nine groups has specific number of alternative components to be combined with front drive train (FD), e.g. gear 42-32-24T can be combined with six Front Crank sets (FC): M980, M780, M670, M610, M552, M522. To construct the design platform D_0, a non-symmetric matrix consisting of 38 rows and 19 columns has been used. In Fig. 6, elements of the matrix noted with “X” stand for incompatible components.

For this design platform D_0, complete design space is defined by $n_{s_0} = 722$ product configurations and restricted design space expressed by $n_{v_0} = 239$ product configurations.

By using this matrix, it is possible to gradually remove selected components/entries with restrictions from the product platform D_0 to obtain alternative platforms.

In order to benchmark possible concurrent SHIMANO product platforms at once, gears 48-36-26T including eight crank sets (M610, T780, M670, T781, T671, T611, T551 and T521) have been selected for an execution into the platform D_1. This group of components was selected for an execution based on the criterion of the highest density of restrictions. Subsequently, we obtain compatibility table, where eight gears 48-36-26T are eliminated for D_1, as seen in Fig. 6. The number of rows in this table was reduced from 38 to 30.

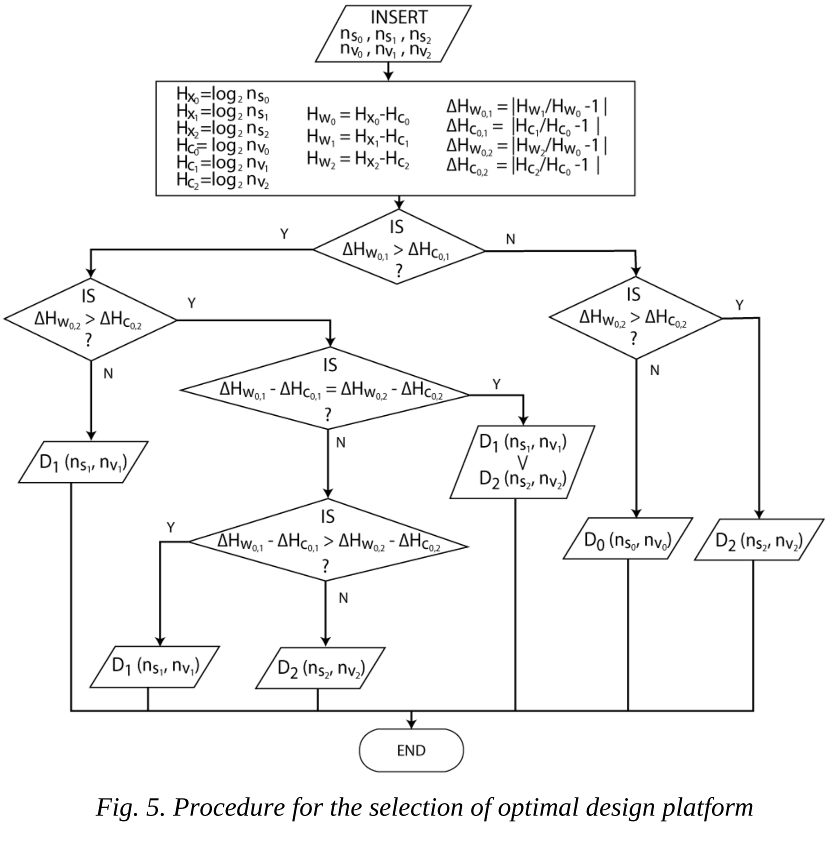
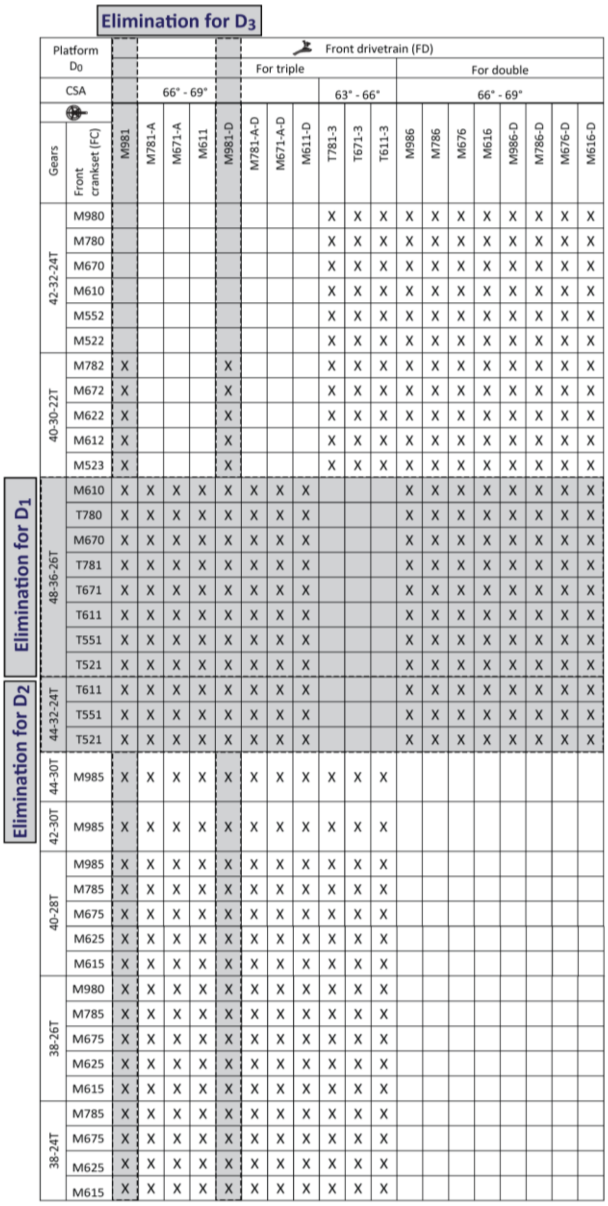


Fig. 5. Procedure for the selection of optimal design platform



Elimination for D_3

Elimination for D_1

Elimination for D_2

Platform D_0																				
		Front drivetrain (FD)																		
							For triple						For double							
CSA		66° - 69°								63° - 66°			66° - 69°							
Gears	Front crankset (FC)	M981	M781-A	M671-A	M611	M981-D	M781-A-D	M671-A-D	M611-D	T781-3	T671-3	T611-3	M986	M786	M676	M616	M986-D	M786-D	M676-D	M616-D
42-32-24T	M980									X	X	X	X	X	X	X	X	X	X	X
	M780									X	X	X	X	X	X	X	X	X	X	X
	M670									X	X	X	X	X	X	X	X	X	X	X
	M610									X	X	X	X	X	X	X	X	X	X	X
	M552									X	X	X	X	X	X	X	X	X	X	X
	M522									X	X	X	X	X	X	X	X	X	X	X
40-30-22T	M782	X				X				X	X	X	X	X	X	X	X	X	X	X
	M672	X				X				X	X	X	X	X	X	X	X	X	X	X
	M622	X				X				X	X	X	X	X	X	X	X	X	X	X
	M612	X				X				X	X	X	X	X	X	X	X	X	X	X
	M523	X				X				X	X	X	X	X	X	X	X	X	X	X
48-36-26T	M610	X	X	X	X	X	X	X	X				X	X	X	X	X	X	X	X
	T780	X	X	X	X	X	X	X	X				X	X	X	X	X	X	X	X
	M670	X	X	X	X	X	X	X	X				X	X	X	X	X	X	X	X
	T781	X	X	X	X	X	X	X	X				X	X	X	X	X	X	X	X
	T671	X	X	X	X	X	X	X	X				X	X	X	X	X	X	X	X
	T611	X	X	X	X	X	X	X	X				X	X	X	X	X	X	X	X
	T551	X	X	X	X	X	X	X	X				X	X	X	X	X	X	X	X
	T521	X	X	X	X	X	X	X	X				X	X	X	X	X	X	X	X
44-32-24T	T611	X	X	X	X	X	X	X	X				X	X	X	X	X	X	X	X
	T551	X	X	X	X	X	X	X	X				X	X	X	X	X	X	X	X
	T521	X	X	X	X	X	X	X	X				X	X	X	X	X	X	X	X
44-30T	M985	X	X	X	X	X	X	X	X	X	X	X								
42-30T	M985	X	X	X	X	X	X	X	X	X	X	X								
40-28T	M985	X	X	X	X	X	X	X	X	X	X	X								
	M785	X	X	X	X	X	X	X	X	X	X	X								
	M675	X	X	X	X	X	X	X	X	X	X	X								
	M625	X	X	X	X	X	X	X	X	X	X	X								
	M615	X	X	X	X	X	X	X	X	X	X	X								
38-26T	M980	X	X	X	X	X	X	X	X	X	X	X								
	M785	X	X	X	X	X	X	X	X	X	X	X								
	M675	X	X	X	X	X	X	X	X	X	X	X								
	M625	X	X	X	X	X	X	X	X	X	X	X								
	M615	X	X	X	X	X	X	X	X	X	X	X								
38-24T	M785	X	X	X	X	X	X	X	X	X	X	X								
	M675	X	X	X	X	X	X	X	X	X	X	X								
	M625	X	X	X	X	X	X	X	X	X	X	X								
	M615	X	X	X	X	X	X	X	X	X	X	X								

Fig. 6. SHIMANO compatibility table with Platforms $D_{0\text{-}3}$ for gears and front derailleur [9]

This way we obtained platform D_1 defined by $n_{s_1} = 570$ drivetrain configurations and $n_{v_1} = 215$ restricted (viable) product configurations.

Afterwards, for determination of the platform D_2, we proceed towards elimination of gear type 44-32-24T (including three crank sets T611, T551 and T521), as can be seen in Fig. 6. The number of rows in this table was reduced from 30 to 27.

We obtained platform D_2 defined by $n_{s_2} = 513$ drivetrain configurations and $n_{v_2} = 206$ restricted (viable) product configurations. In order to provide other alternative product platform D_3 for the benchmarking study, two FDs M981 and M981-D have been eliminated due to the high number of restrictions related to these two components. Then, we obtained platform D_3 defined by $n_{s_3} = 459$ drivetrain configurations and $n_{v_3} = 194$ restricted (viable) product configurations, while the number of columns decreased from 19 to 17.

Obtained numbers of drive train configurations and related values of waste entropy H_w and constrained design space H_c (with and without restrictions) are summarily depicted in Table 3.

As can be seen from Fig. 7(a), values of constrained (feasible) product configurations as well as related constrained design space have decreasing character starting from the Platform D_0 to D_3. Another logical objective of the methodology - decreasing waste (infeasible) configurations - has been satisfied as well, as confirmed by the Fig. 7(b).

In the next step, the decision-making algorithm to determine suitable extent of product variety for different platforms can be applied. Since algorithm in Fig. 5 is dedicated for the decision on maximum three alternative design platforms, an extension of this algorithm for maximum four design platforms can be constructed. After using this algorithm and the procedure proposed, we obtain a decision for Platform D_3 as the most suitable with respect to the amount of the waste entropy.

5. CONCLUSION

The novel method can be employed to assist product managers to independently assess competitive product variety platform against each other and to evaluate their customization characteristics quantitatively. As it was shown and proved on multiple cases, proposed approach leads to decision for optimal product platform.

Some authors, e.g. [10-15] argue that infeasible configurations might be hidden to improve “configuration experience” by using sophisticated

Table 3. *Computational results of entropy and number of product configurations*

Approach	Indicator	Platform D_0	Platform D_1	Platform D_2	Platform D_3
Design space-based approach	Entropy of design space	$E_x=\log_2 n_s$ [bit] $n_s=722$ $E_x=9.50$ bits	$E_x=\log_2 n_s$ [bit] $n_s=570$ $E_x=9.15$ bits	$E_x=\log_2 n_s$ [bit] $n_s=513$ $E_x=9.00$ bits	$E_x=\log_2 n_s$ [bit] $n_s=459$ $E_x=8.84$ bits
	Entropy of constrained design space	$E_C=\log_2 n_v$ [bit] $n_v=239$ $E_C=7.90$ bits	$E_C=\log_2 n_v$ [bit] $n_v=215$ $E_C=7.75$ bits	$E_C=\log_2 n_v$ [bit] $n_v=206$ $E_C=7.69$ bits	$E_C=\log_2 n_v$ [bit] $n_v=194$ $E_C=7.60$ bits
	Waste entropy ratio	$E_w/E_x=16.7\%$	$E_w/E_x=15.4\%$	$E_w/E_x=14.7\%$	$E_w/E_x=14.0\%$
Abosolute NPC-based approach	Number of all possible configurations	*722 conf.*	*570 conf.*	*513 conf.*	*459 conf.*
	Feasible product configurations	*239 conf.*	*215 conf.*	*206 conf.*	*194 conf.*
	Relative number of infeasible configurations	67%	62,3%	59,8%	57,7%

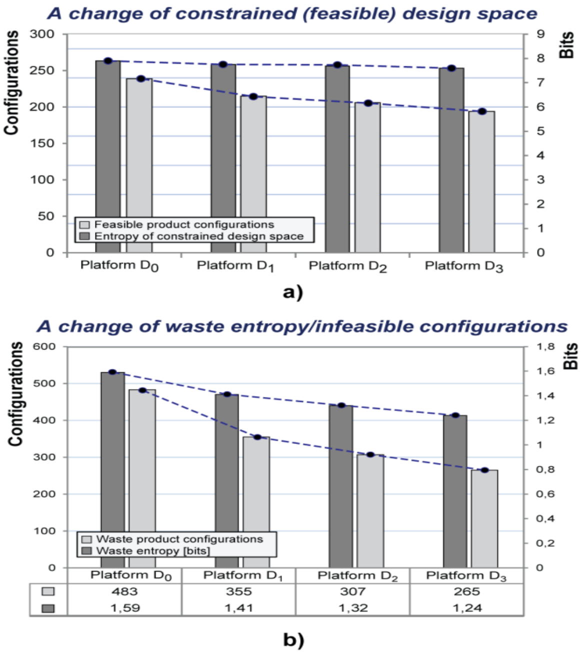


Fig. 7. Graphical interpretation of constrained (feasible) design space (a), and waste entropy/configurations (b) within Platforms $D_{0\text{-}3}$

product configurators. It was also proven in psycho-social domain (see e.g. [16]), that any changes of long-term accepted rules in human behaviour initiate disappointments or frustrations. On the other hand, it is evident that one of configurator types is developed especially for options that include also infeasible component combinations [17].

Thus, the problem treated in this paper opens new research perspectives because each different sector of mass customization requires effective approach to solve CSPs.

6. ACKNOWLEDGMENT

This paper has been supported by VEGA project no. 1/0419/16 granted by the Ministry of Education of the Slovak Republic.

7. REFERENCES

[1] M. Bonev, L. Hvam, J. Clarkson, and A. Maier, “Formal computer-aided product family architecture design for mass customization,” *Computers in Industry*, vol. 74, pp. 58-70, 2015.

[2] R. Wilson, and A. V. Hill, *The Operations Management Complete Toolbox (Collection)*. New York, USA: FT Press, 2013.

[3] Z. Anisic, and C. Krsmanovic, “Assembly initiated production as a prerequisite for mass customization and effective manufacturing,” *Strojniški vestnik*, vol. 54, no. 9, pp. 607-618, 2008.

[4] C. E. Shannon, “A mathematical theory of communication,” *Bell Systems Technical Journal*, vol. 27, no. 3, pp. 379-423, 1948.

[5] P. Krus, “Design Space Configuration for Minimizing Design Information Entropy,” in *ICoRD’15 Research into Design Across Boundaries 2015*, 2015, pp. 51-60.

[6] D. Yang, and M. Dong, “A constraint satisfaction approach to resolving product configuration conflicts,” *Advanced Engineering Informatics*, vol. 26, no. 3, pp. 592-602, 2012.

[7] V. Modrak, D. Marton, and S. Bednar, “The Influence of Mass Customization Strategy on Configuration Complexity of Assembly Systems,” *Procedia CIRP*, vol. 33, pp. 539-544, 2015.

[8] V. Modrak, S. Bednar, and D. Marton, “Generating product variations in terms of mass customization,” in *IEEE 13th International Symposium on Applied Machine Intelligence and Informatics (SAMI 2015)*, 2015, pp. 187-192.

[9] SHIMANO, “Shimano compatibility catalogue 2014/2015,” Retrieved 12 January, 2016, from http://productinfo.shimano.com/download?path=pdfs/archive/2014-2015_Compatibility_v010_en.pdf, 2014.

[10] V. Modrak, P. Krus, and S. Bednar, “Approaches to Product Variety Management Assuming Configuration Conflict Problem,” *FME Transactions*, vol. 43, pp. 271-78, 2015.

[11] P. Pitiot, M. Aldanondo, and E. Vareilles, “Concurrent product configuration and process planning: Some optimization experimental results,” *Computers in Industry*, vol. 65, no.4, pp. 610-621, 2014.

[12] P. T. Helo, Q. L. Xu, S. J. Kyllonen, and R. J. Jiao, “Integrated vehicle configuration system connecting the domains of mass customization,” *Computers in Industry*, vol. 61, no. 1, pp. 44–52, 2010.

[13] A. Trentin, E. Perin, and C. Forza, “Increasing the consumer-perceived benefits of a mass-customization experience through sales-configurator capabilities,” *Computers in Industry*, vol. 65, no. 4, pp. 693-705, 2014.

[14] A. Trentin, E. Perin, and C. Forza. “Sales configurator capabilities to avoid the product variety paradox: Construct development and validation,” *Computers in Industry*, vol. 64, no. 4, pp. 436-447, 2013.

[15] M. Aldanondo, E. Vareilles, M. Djefel, “Towards an association of product configuration with production planning,” *International Journal of Mass Customisation*, vol. 3, no. 4, pp. 316–332, 2010.

[16] J. A. Quelch, and K. E. Jocz, *Greater good: How good marketing makes for better democracy*. Harvard Business Press, 2013.

[17] K. Orsvärn, and T. Axling, “The Tacton view of configuration tasks and engines,” in *Workshop on Configuration, Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, 1999, pp. 127-130.

CORRESPONDENCE

Slavomir Bednar, PhD.
Technical University of Kosice
Faculty of Manufacturing Technologies in Presov,
Bayerova 1
08001 Presov, Slovakia
slavomir.bednar@tuke.sk

Vladimir Modrak, Prof.
Technical University of Kosice
Faculty of Manufacturing Technologies in Presov,
Bayerova 1
08001 Presov, Slovakia
vladimir.modrak@tuke.sk