

Calculation of the number of branches of multi-valued decision trees in computer aided importance rank of parameters

Agnieszka Tiszbirek^{1,*}

¹Opole University of Technology, Faculty of Production Engineering and Logistics, Sosnkowskiego 31 Str., 45-272 Opole, Poland

Abstract. An elaborated digital computer programme supporting the time-consuming process of selecting the importance rank of construction and operation parameters by means of stating optimum sets is based on the Quine – McCluskey algorithm of minimizing individual partial multi-valued logic functions. The example with real time data, calculated by means of the programme, showed that among the obtained optimum sets there were such which had a different number of real branches after being presented on the multi-valued logic decision tree. That is why an idea of elaborating another functionality of the programme – a module calculating the number of branches of real, multi-valued logic decision trees presenting optimum sets chosen by the programme was pursued. This paper presents the idea and the method for developing a module calculating the number of branches, real for each of optimum sets indicated by the programme, as well as to the calculation process.

1 Introduction

Automation and control systems which are currently formed are complex, but many of their construction and operation parameters are interdependent, which in turn affects the quality and/or operating efficiency of these devices. This is why, it is not advisable to omit the above-mentioned dependencies during the machine design and construction process. It is also not possible to ignore them in the process of stating the importance rank of parameters of the analysed device, which should cooperate with testing and improving the device in creation [1,2]. The process of ascribing the importance rank was complex and complicated as far as calculations were concerned, and thus a computer programme based on an appropriate algorithm was created to aid the process. During works on the programme, the above-mentioned dependency of parameters was additionally taken into consideration, which led to creating appropriate modules (conditionality and a proxy variable) to protect dependent variables from division.

2 Description and operating principle of the programme

2.1. The algorithm of the programme

The basis of the programme is the Quine – McCluskey algorithm of the minimization of individual partial multi-valued logic functions [3,4,5]. Its basis is a mathematical equation composed of certain elements calculated on the basis of gathered and encoded data which, for the first stage of the algorithm, can be presented in the following form:

$$x_1 = i_k - i_z * w_z + i_z + p \quad (1)$$

where:

x_1 – the number of step branches of the analysed decision tree,

i_k – the number of branches of the k -th level of the decision tree,

i_z – the number of simplifications of the z -th variable,

w_z – the valency of the z -th variable,

p – the number of branches, out of which the upper branches of the k -th level, which are not subject to simplification, were formed [6].

The main operation principle of the discussed algorithm is finding the **so-called primary implicants**, *i.e.* each combination which is not subject to further joining. The number of calculations which are made at the first stage depends on the number of analysed parameters and they are to be made this particular number of times. It is possible to proceed to the next step only after stating all possible minimums at a given stage. The next step consists in making compounds according

*Corresponding author: a.tiszbirek@po.opole.pl

to the parameter with the smallest value and then calculations for the other variables are repeated. If the minimum value was obtained at a given stage two or more times, the subsequent stages would have to be calculated collaterally, for each of the obtained minimums.

2.2 Computer programme

The window of the main programme consists of the main menu (*File* – the close the programme, *Data* – the download / save data, *Results* – save calculations and/or optimum sets, *Info* – information about the programme and the author), the area with loaded data (*KAPN*) and function buttons (*Add*, *Erase*, *Calculate* and *Check scopes*). Buttons *Calculate* and *Check scopes* are primarily off and they appear only when the data is loaded. The former causes automated calculation of each step and display of calculations together with interim results in a separate window run by clicking on the button *Results* (appearing after calculations are made); whereas the latter modifies values of these parameters, the maximum code value of which does not appear in any of the selected sets of parameters [7].

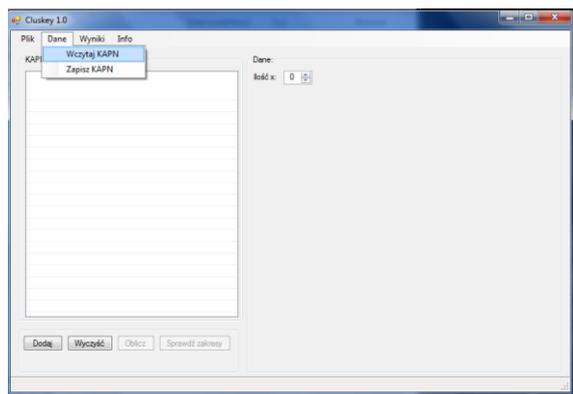


Fig. 1. Basic window of the programme with an option of downloading data.

Additional downloading of data causes the appearance of two subsequent buttons *Conditions* and *Proxy variable*, which start modules of new functionalities. The former imposes appropriate conditions of occurring parameters one after another. After it has been started, an additional window appears with a list of parameters and additional options, by means of which we may indicate which parameters and in which set should appear in subsequent optimum sets.

The algorithm of the programme, with the function of conditions, works similarly to the basic version of the programme. The only innovation is checking whether a newly calculated minimum is not included in the imposed condition (if it is at least the penultimate parameter). If this is the case, it has to be discarded and a new one has to be found. If, however, the minimum value is a parameter at the end of the imposed condition, the remaining parameters have to be set without additional calculations in an imposed sequence and then we should proceed to subsequent calculation stages. The second functionality of the programme is the possibility

of coding an appropriately chosen group of parameters in one or a few proxy variables. An additional advantage of the programme is automation of coding of proxy variables. After choosing appropriate parameters, the programme sets a new value of the proxy variable by itself and assigns appropriately set verses with values of intrinsic parameters to a given code value.

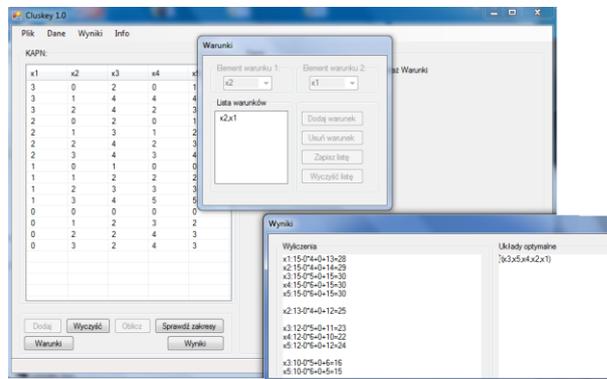


Fig. 2. The view of the programme with a condition and result window.

This process not only shortens coding time but also protects it from the mistakes resulting from the lack of due care or the plurality of variables and their significant multivalence. The process of calculation in the case of this functionality is based on the same algorithm of the programme as its basic version. After proxy variables have been coded, they are treated as “new” parameters (variables included in them are deleted from the table of parameters memory) and then a classic process of calculation with showing final results at the end takes place.

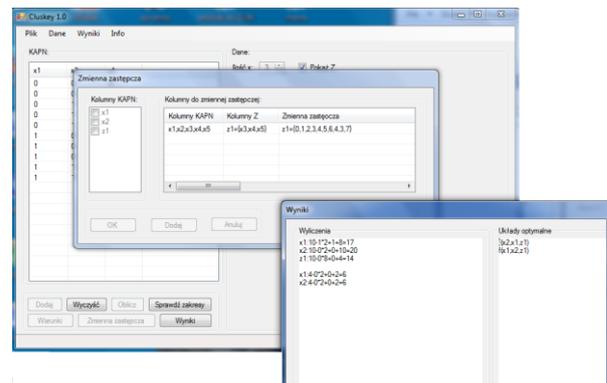


Fig. 3. The view of the programme with proxy variable and result windows.

3 Calculation complexity problems with dependent variables of similar importance

The problem described in [8] is a calculation complexity in the process of determining the importance rank of parameters of similar importance. Additionally, a dependence occurs between parameters and this aids to the difficulty of the process. An example presented in the elaboration is a set of symptoms in thyroid problems.

These symptoms were gathered during research and marked with appropriate x values. X parameters are bivalent (1 signifies the occurrence of the symptom, whereas 0 signifies its lack) and are set in eight variables $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$ written numerically in Table 1. Basic calculations were first made by means of rough sets, and then by means of bivalent decision logic trees [9]. Calculating an example by means of a computer programme was the next stage [10].

Table 1. A digital record of coding variables.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
0	1	0	0	1	0	0	1
0	0	0	1	1	0	0	1
0	1	0	1	0	1	1	0
1	1	0	1	0	1	1	0
0	1	1	1	0	1	1	0

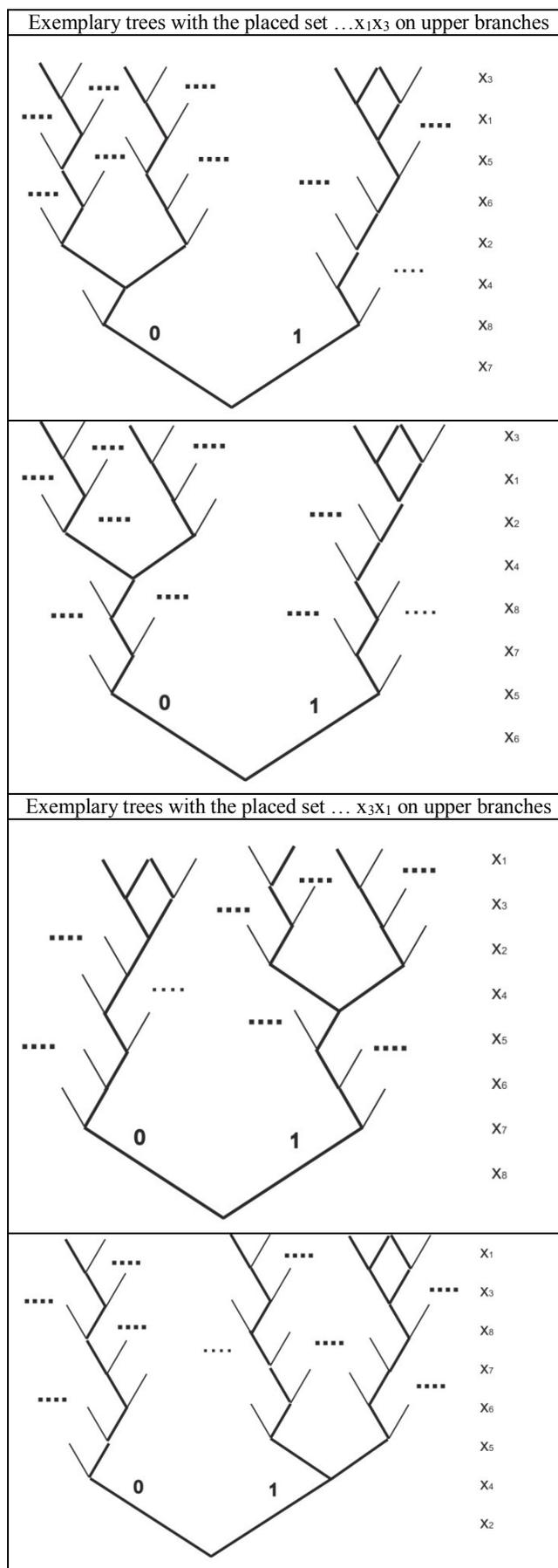
576 optimum sets we obtained, the analysis of which showed the lowest importance rank of parameters x_1 and x_3 as well as the dependency between parameters x_2 and x_4 . This is why an idea of applying such functionalities as proxy variable and conditionality in case of subsequent calculations emerged. The realization of the idea of applying a proxy variable z, in which parameters x_1 and x_3 are included, with applying the condition of the sequence of parameters x_2 and x_4 at the same time, has is described in [8]. It has been forced in calculations that parameters x_2 and x_4 always occur in the following order x_4x_2 , whilst parameters x_1 and x_3 become one variable z. Consequently the process calculation time is reduced along with the number of optimum sets to 120. Such a reduced number of optimum sets makes it possible to make further analysis in an easier way focusing only on remaining variables (x_5, x_6, x_7, x_8). A deeper analysis of obtained results would make it possible to assume that remaining parameters are of the same importance rank. It is thus possible to state that they are equivalent and independent.

4 Problem defining optimum sets of parameters on the basis of the number of real branches

4.1 The need and idea for the realisation of a new module of the computer programme

Although during calculations after drawing exemplary multi-valued logic trees for chosen optimum sets it turned out that they are different from each other in the number of real branches (highlighted on trees). For example, a tree representing a set $x_6x_5x_7x_2x_4x_8x_1x_3$ had 21 irreducible real branches, whereas with the set $x_7x_8x_4x_2x_6x_5x_1x_3$ it had 23 of them. A further analysis of the trees showed that up to a certain level, all trees are on the same level – possessing the same number of real branches, which is 13 (the fourth floor of the tree counting from the top).

Table 2. Exemplary optimum sets presented on shortened logic trees.



Therefore, the choice of the fifth variable was more important. For example, putting the variable x_7 after x_8 in the first set was beneficial (there are 15 real branches at this tree level), whereas choosing x_2 after x_6 was not (16 real branches). This problem showed how useful the method of checking irreducible real branches on multi-valued logic trees is. It enables finding the best sets of parameters among advantageous ones, even if the difference between them would be minimum.

The new functionality would be a module calculating the number of branches of real, multi-valued, logic decision trees on which optimum sets chosen by the programme will be located.

4.2 Modification of the basic algorithm

The creation of a new module requires modification of the basic algorithm and new software. The number of real branches has to be counted by summing their number on each level. In order to get to know the number of branches on each level, it is necessary to calculate the difference in the number of branches of k -th level of a decision tree and the number of simplifications of z -th variable multiplied by the valency of z -th variable for a given stage. Only the last stage should be calculated completely according to the basic algorithm. Sums obtained in this way have to be juxtaposed against each other. The smallest ones will indicate the most beneficial sets. Therefore, the pattern of the basic equation (1) has to be modified to the following form (for all stages, except the final one, it will be counted according to the basic pattern):

$$x_l = i_k - i_z * w_z \quad (2)$$

where:

- x_l – the number of step branches of the analysed decision tree,
- i_k – the number of branches of the k -th level of the decision tree,
- i_z – the number of simplifications of the z -th variable,
- w – the valency of the z -th variable.

The value of each substage (each tree level) has to be counted according to the formula above. The value for the whole set will be used for comparisons. For example, for the set $x_8x_7x_6x_5x_4x_2x_3x_1$ it is necessary to count values for particular x at a given stage as indicated below:

- Step I (on x_1): $x_1:5-1*2+1+3=7$
- Step II (on x_3): $x_3:4-0*2+0+3=7$
- Step III (on x_2): $x_2:3-0*2+0+3=6$
- Step IV (on x_4): $x_4:3-0*2+0+2=5$
- Step V (on x_5): $x_5:2-0*2+0+2=4$
- Step VI (on x_6): $x_6:2-0*2+0+2=4$
- Step VII (on x_7): $x_7:2-0*2+0+2=4$

Clearly, only the part of the equation which is included in the modified formula (the part in bold print) is to be taken into consideration in calculations. Finally, the following values will be obtained for each stage and their sum will give 21:

- Step I (on x_1): $x_1:5-1*2=3$
- Step II (on x_3): $x_3:4-0*2=4$
- Step III (on x_2): $x_2:3-0*2=3$
- Step IV (on x_4): $x_4:3-0*2=3$
- Step V (on x_5): $x_5:2-0*2=2$
- Step VI (on x_6): $x_6:2-0*2=2$
- Step VII (on x_7): $x_7:2-0*2+0+2=4$

The resulting value (21) is the number of all real branches for the calculated example. In this way, it is necessary to calculate values for each of optimum sets indicated by the programme and then compare these values. These sets which obtained the minimum value (or values) will be the most beneficial, even if their number is slightly different.

4.3 New functionality – data for a chosen device

An axial-flow vortex pump, which is a kind of a centrifugal rotary pump, was chosen for exemplary calculations. This pump is characterised by the following physical quantities (which influence its work efficiency):

- Q – describing the volume flow,
- H – describing the head,
- N – describing the power,
- η – describing the efficiency,
- n – describing the rotational speed,
- φ – describing the angle of setting the impeller blades.

Knowing the model construction we may connect the parameters describing its operation and elements responsible for its values. This, in turn, makes it possible to carry out a better analysis of the set of pump parameters from with a view to determine particular importance ranks and thus, an optimal set of construction and operation parameters.

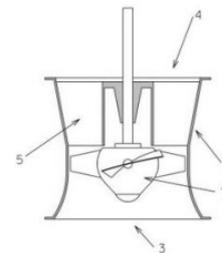


Fig. 4. Construction of the propeller pump

Pumps of this type are mainly used in industrial and energetic installations, particularly in power plants (most often in small water power plants). The research was carried out on the model turbine (designed and built with the use of construction elements of the serial axial-flow vortex pump 25P21-2 of the Warsaw Factory of Pumps and Valves - *Fabryka Pomp i Armatury WAFAPOMP SA*). The actual results had to be selected and coded in an appropriate way in order to make it possible to present it by means of a multi-valued logic function. Final values got an appropriate code number: $\varphi, n=0,1,2,3$; $Q=0,1,2,3,4,5$; $H, N=0,1,2,3,4,5,6$ [11].

Table 3. Coded results of parameter values

φ	n	Q	H	N
0	0	0	0	0
0	1	2	3	2
0	2	2	4	3
0	3	2	4	3
1	0	1	0	0
1	1	2	2	2
1	2	3	3	3
1	3	4	5	5
2	1	3	1	2
2	2	4	2	3
2	3	4	3	4
2	0	2	0	1
3	0	2	0	1
3	1	4	4	4
3	2	4	2	3

Computer-based calculations of entered values made it possible to obtain the following results :

step I:

$$\begin{aligned} x_1: 15 \cdot 0^4 + 0 + 13 &= 28 \\ x_2: 15 \cdot 0^4 + 0 + 14 &= 29 \\ x_3: 15 \cdot 0^5 + 0 + 15 &= 30 \\ x_4: 15 \cdot 0^6 + 0 + 15 &= 30 \\ x_5: 15 \cdot 0^6 + 0 + 15 &= 30 \end{aligned}$$

step II (on x_1):

$$\begin{aligned} x_2: 13 \cdot 0^4 + 0 + 12 &= 25 \\ x_3: 13 \cdot 0^5 + 0 + 12 &= 25 \\ x_4: 13 \cdot 0^6 + 0 + 12 &= 25 \\ x_5: 13 \cdot 0^6 + 0 + 13 &= 26 \end{aligned}$$

step IIIa (on x_2):

$$\begin{aligned} x_3: 12 \cdot 0^5 + 0 + 11 &= 23 \\ x_4: 12 \cdot 0^6 + 0 + 10 &= 22 \\ x_5: 12 \cdot 0^6 + 0 + 12 &= 24 \end{aligned}$$

step IIIb (on x_3):

$$\begin{aligned} x_2: 12 \cdot 0^4 + 0 + 11 &= 23 \\ x_4: 12 \cdot 0^6 + 0 + 8 &= 20 \\ x_5: 12 \cdot 0^6 + 0 + 11 &= 23 \end{aligned}$$

step IIIc (on x_4):

$$\begin{aligned} x_2: 12 \cdot 0^4 + 0 + 10 &= 22 \\ x_3: 12 \cdot 0^5 + 0 + 8 &= 20 \\ x_5: 12 \cdot 0^6 + 0 + 11 &= 23 \end{aligned}$$

step IVa (on x_4):

$$\begin{aligned} x_3: 10 \cdot 0^5 + 0 + 6 &= 16 \\ x_5: 10 \cdot 0^6 + 0 + 5 &= 15 \end{aligned}$$

step IVb (on x_4):

$$\begin{aligned} x_2: 8 \cdot 0^4 + 0 + 6 &= 14 \\ x_5: 8 \cdot 0^6 + 0 + 4 &= 12 \end{aligned}$$

step IVc (on x_3):

$$\begin{aligned} x_2: 8 \cdot 0^4 + 0 + 6 &= 14 \\ x_5: 8 \cdot 0^6 + 0 + 4 &= 12 \end{aligned}$$

Optimal systems:

$$f(x_3, x_5, x_4, x_2, x_1)$$

$$f(x_2, x_5, x_4, x_3, x_1)$$

$$f(x_2, x_5, x_3, x_4, x_1)$$

where: x_1 - μ ; x_2 - n ; x_3 - Q ; x_4 - H ; x_5 - N

As calculated, there should be three optimum (best) sets. However, further calculations (the calculation of the number of real branches for each set) show better and worse sets among them. Calculations for particular sets are presented below:

$f(x_3, x_5, x_4, x_2, x_1)$	$f(x_2, x_5, x_4, x_3, x_1)$	$f(x_2, x_5, x_3, x_4, x_1)$
$x_1: 15 \cdot 0^4 = 15$	$x_1: 15 \cdot 0^4 = 15$	$x_1: 15 \cdot 0^4 = 15$
$x_2: 13 \cdot 0^4 = 13$	$x_3: 13 \cdot 0^5 = 13$	$x_4: 13 \cdot 0^6 = 13$
$x_4: 12 \cdot 0^6 = 12$	$x_4: 12 \cdot 0^6 = 12$	$x_3: 12 \cdot 0^5 = 12$
$x_5: 10 \cdot 0^6 + 0 + 5 = 15$	$x_5: 8 \cdot 0^6 + 0 + 4 = 12$	$x_5: 8 \cdot 0^6 + 0 + 4 = 12$
sum: 55	sum: 52	sum: 52

It is thus possible to select two sets out of three, the number of real branches of which is the lowest value and these two sets ($f(x_2, x_5, x_3, x_4, x_1)$, $f(x_2, x_5, x_4, x_3, x_1)$) can be acknowledged to be the most beneficial.

Certainly, knowing the construction and the process of pump operation, it is obvious that parameters φ and n should occur one after another because of their interactivity (requires applying an appropriate condition here). However, for the needs of the analysed problem it was resolved not to in order not to overshadow the calculation results.

5 Conclusion

An example described in this article shows how a computer programme can facilitate the process of calculating and how useful it can be in determining the importance rank of construction and operation parameters. A complex and time-consuming process lasts from a few to several dozens of seconds, it does not draw the user's attention and it eliminates many potential mistakes. Additional functionalities of the programme (such as the possibility of imposing conditionality of the set of parameters and creation of proxy variables) are equally important and useful. However, as shown, sometimes even these elements of the programme are insufficient, which is why the creation of the described module determining the number of real branches is an important solution recommended for proper functioning of the process of determining the importance rank of construction and operation parameters. A new module will facilitate the calculating process and confirm the credibility of obtained results. It will also solve the problem of obtaining a huge number of beneficial sets differing in the minimum number of real branches. This is why in subsequent stages, the described idea should be programmed and the obtained solution should be tested and described.

References

1. M. Giovanni, *Synthesis and optimization of digital circuits* (Wydawnictwa Naukowo - Techniczne, 1998)
2. C. D. Kozen, *Automata and Computability* (Springer, 1997)
3. E. J. McCluskey, *Bell System Tech. Journal.* **35**, 1417-1444 (1956)

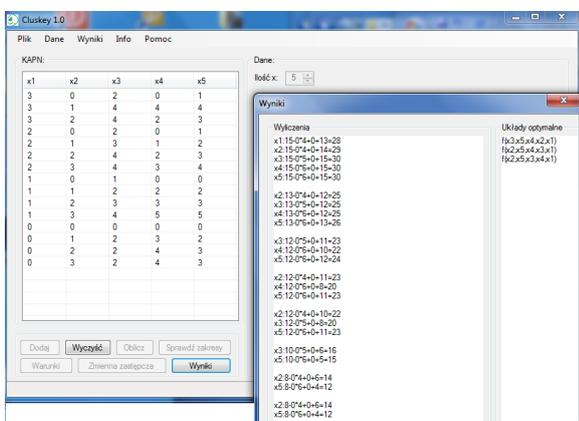


Fig. 5. Computer-based calculations for the analysed example.

4. T. K. Jain, D. S. Kushwaha, A. K. Misra, *Optimization of the Quine- McCluskey Method for the Minimization of the Boolean Expressions. International Conference on Autonomic and Autonomous Systems (ICAS, 2008)*
5. R. Mohan Ranga Rao, IJAEST **3**(1), 12-14 (2011).
6. M. A. Partyka, A. Tiszbierek, Napędy i Sterowanie. **9/2015**, 132-139 (2015).
7. S. Ford, *Microsoft Visual Studio Tips* (Microsoft Press, 2008).
8. A. Tiszbierek, Napędy i Sterowanie, *Zastosowanie komputerowego wspomaganie wyznaczania rangi ważności parametrów z funkcją zmienna zastępcza oraz warunkowość na rzeczywistym przykładzie ze zmiennymi zależnymi o podobnej ważności* (in review).
9. M. A. Partyka, *Podobieństwa i różnice przybliżonej klasyfikacji obiektów w ujęciu logiki i systemów informacyjnych dla CAD procesów decyzyjnych. XXIV Konferencja Zastosowań Matematyki* (Instytut Matematyki PAN, 1995).
10. A. Tiszbierek, M. A. Partyka, *XLV Konferencja Zastosowań Matematyki* (Instytut Matematyki PAN, 73-75, 2016).
11. M. A. Partyka, R. Łuszczyna, M. Sojka, *Konferencja Komputerowe Zintegrowane Zarządzanie* (Oficyna Wydawnicza Polskiego Towarzystwa Zarządzania Produkcją, Opole, 292-299, 2009).