THE METHOD OF PARAMETRIC DECISION TREES IN THE ANALYSIS OF AUTOMATIC TRANSMISSION

Adam Deptuła*

Department of Management and Production Engineering, Faculty of Production Engineering and Logistics , Opole University of Technology, 76 Prószkowska St., 45-758 Opole, Poland

ARTICLE INFO filled by the publisher

***Article history*:**
Received: 24.02.2021
Received in revised form: 24.03.2021.
Accepted: 13.07.2022.

Keywords:
Computational model
Mechanical engineering
Optimization
Automatic gearboxes
Graph theory

DOI: https://doi.org/10.30765/er.1802

Abstract:

Current applications of graph theory involve gear modeling for dynamic analysis, kinematic analysis, synthesis, structural analysis, optimization of gear chains, and automatic design based on the so-called graph grammars. Some tasks can be solved only with methods from graph theory, for example, the enumeration of design solutions.
The purpose of modelling an automatic transmission with graphs can be versatile, namely: determining the ratio of individual gears, analyzing the speed and acceleration of individual rotational elements. At a later stage, the methods of decision logical trees can be used to analyze the functional schemes of selected gears. However, for graphs that are transmission models, tree structures that play parametrically can be used. This allows for the generalization and extension of the algorithmic approach.

1 Introduction

The optimization problem can be solved by reviewing all possibilities (all elements of the state space). The state space reflects the model of a given system, e.g. a machine system is described by the state space. Diagrams are created that represent a certain class of phenomena in a given problem area in order to create a basis for research and (or) communication. In a general sense, it is a mental (internal) or formal description (diagrams, mathematical formulas, relations, etc.), generally reduced to the most essential symbolic features. In technical issues, there are real models that describe the actual construction of a specific object. One can quote Regnier's definitions [1]: Nothing, on the one hand, that an abstract object is completely described by its definitions, and a concrete object is never exhaustively described, one can say: an abstract object is a model of a concrete object, if the definition the former is taken to represent the latter.

The real model, then, is one that describes the actual construction of a specific object“. In the case of systems built of a large number of subsystems, analytical solving of differential equations is usually labor-intensive or may be significantly difficult. In such cases, network methods are used, called in the literature non-classical methods. Due to the high degree of algorithmization of network methods, their implementation in computer computing systems is facilitated. On the other side, graphically - in the form of graphs, they present the structure of the model system. Graphs and structural numbers have long played a role as models of mechanical systems [2] and are still systematically developed [3, 4, 5, 6, 7]. Power flow graphs (bond graphs) in system modeling are presented, among others, by [8] while graphs used in hydraulic systems by [9]. Graph classes such as polar graphs, flow graphs, hybrid graphs and structural numbers are used. In addition, there are special stream graphs, e.g. in chemical and process engineering. Trees are a specific type of connected graphs without cycles. Many examples of trees are provided by logical structures, for example: multivalued logical trees, polls, multi-income game dendrites, (...).

The property of trees is that they start at the root from which the branches are built up. The use of trees in the optimization of machine systems is fully useful in the sphere of concepts, because it allows the selection

*Corresponding author
E-mail address:a.deptula@po.edu.pl

(change) of the arithmetic values of the appropriate design and / or operational parameters of a given machine system and the assessment of the system operation under new conditions [10, 11, 12]. At each stage of optimization, it is possible to create a tree by selecting the optimal decisions. Then it is possible to attach vertices to the tree to represent the optimal respones to changes in arithmetic construction parameters. If the number of vertices at each decision level is limited to those that represent true (realizable) construction guidelines, then a graph can be constructed, extended 'deep' in a certain direction. In particular, there are numerous practical and implemented applications of machine system optimization using logical trees and dependency graphs [13, 14]. Parametric graphs used in the distribution of the contour graph of planetary gears [15] can be used in the analysis of the automatic gearboxes.

2 Theoretical models of gears

The advantage of modeling gears with graphs is that the problems considered with graph models can be solved in an algorithmic manner, which allows the easy use of computer programs and widely understood integrated decision-making systems. In the sense of graph theory, a graph is associated with many other algebraic structures, such as, for example, matrices, matroids, structural numbers, cut-off linear spaces. These objects enable the coding of the gear structure, which allows the use of advanced artificial intelligence algorithms: evolutionary, genetic or immunological. The goals of modeling gears with graphs were various - including: dynamic analysis, kinematic analysis, synthesis, structure analysis and enumeration [16, 17]. Among the methods of planetary gear analysis, the following methods can be distinguished: Hsu [18, 19], Freudenstein [20, 21] and Marghitu [22]. In the case of Hsu principles, the graph is built according to the following rules: geometrical dimensions are ignored and kinematic pairs are considered: rotational, planet-yoke and meshing. It is especially useful for considering mechanisms of different types (so-called planar, cross-hair, etc.).

2.1 Graphical dependence graphs

The directed dependency graph [13, 14] defines analytical expressions representing this graph, and thus constituting its analytical model. There are studies in the literature describing the use of dependency graphs and parametric structures in the study of dynamic properties of machine systems. A graph is defined by an ordered pair of sets. The first contains the graph's vertices, and the second contains the edges of the graph, i.e. an ordered pair of vertices. Figure 1 shows an example of the directed dependency graph *G* playing parametrically.

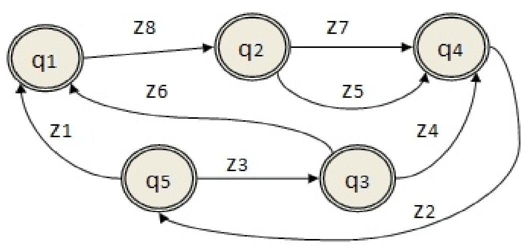

Figure 1. An orientem dependence graph.

A graph is called an ordered pair $G = (V, E)$, in which V is a finie set of elements called the vertices of the graph, and E is a set of $(v_{i,}v_j)(v_{i,}v_j \in V)$ pairs called the edges of the graph. To fully specify the graph, it must be also specify the relationship P formed by individual elements of the set of vertices $V(G)$ and edges $E(G)$. Then the graph can be called ordered three:

$$\mathrm{G} = (\mathrm{V},\ \mathrm{E},\ \mathrm{P}) \tag{1}$$

where: V- set of graph vertices, E- set of edges (graph branches), P- three -member relationship $G = (V,\ E,\ P)$ which meets the following conditions: There is such a pair of vertices $\mathrm{x, y} \in \mathrm{V}$ for each branch e, such that $\langle x,e,y \rangle \in P$.

If for branch e exist $\langle x,e,y\rangle \in P$ and $\langle w,e,z\rangle \in P$ either $x = w$ and $y= z$ or $x = y = z$. A dependency graph is an ordered pair *G=(X, R)*, in which *X* is a finie set of elements called vertices of a graph, and *R* is a set of pairs $(x_i, x_j)(x_i, x_j \in X)$ called the edges of the graph. In the case of parametric graphs, the notation introduced by [4] defines the signs: *G=(Q, Z)*, where *Z* is a set of pairs $(z_i, z_j)(z_i, z_j \in Z)$.

The oriented dependency (game) graph is shown in Figure 1 is composed of a set of vertices *Q*:

$$\mathrm{Q} = \{\mathrm{q}_1, \mathrm{q}_2, \mathrm{q}_3, \mathrm{q}_4, \mathrm{q}_5\} \tag{2}$$

and of a set of edges *Z*, that is an ordered pair of vertices:

$$\mathrm{Z} = \{\mathrm{z}_1, \mathrm{z}_2, \mathrm{z}_3, \mathrm{z}_4, \mathrm{z}_5, \mathrm{z}_6, \mathrm{z}_7, \mathrm{z}_8\} \tag{3}$$

The path in the *G=(Q, Z)* is the edge sequence $(z_{i_1}, z_{i_2}), (z_{i_2}, z_{i_3}), ..., (z_{i_{k-1}}, z_{i_k})$ in which for each $j \in \{2,3,...,k\}$ $(z_{i_{j-1}}, z_{i_j}) \in R$ and vertices $q_{i_1}, q_{i_2}, ..., q_{i_k}$ are different pairs. Vertex q_{i_1} is called the beginning of the pat, and the top q_{i_k} - the end of the road. As a result of a graph distribution from the chosen vertex, a tree structure with cycles is obtained in the first step and then, a general game tree structure is obtained. Each of them has an appropriate analytical formulation G_i^{+} and G_i^{++}. The algorithm for the analysis, structuring and distribution of the dependency graph is presented, among others in the works [13]. For example, for the start vertices q_3, q_4 and q_5 (Figure 1) the analytical expressions are obtained: Gq_3^{++}, Gq_4^{++} and Gq_5^{++} (3).

$$\begin{cases} G_{q3}^{++} = ({}^{0}q_3({}^{1}z_4q_4({}^{2}z_2q_5({}^{3}z_1q_1({}^{4}z_8q_2({}^{5}z_5\overset{1}{q_4}, z_7\overset{2}{q_4})^{5})^{4}, z_3\overset{1}{q_3})^{3})^{2}, \\ z_6q_1({}^{2}z_8q_2({}^{3}z_5\overset{3}{q_4}, z_7\overset{4}{q_4})^{3})^{2})^{1})^{0} \\ G_{q4}^{++} = ({}^{0}q_4({}^{1}z_2q_5({}^{2}z_1q_1({}^{3}z_8q_2({}^{4}z_5\overset{1}{q_4}, z_7\overset{2}{q_4})^{4})^{3}, z_3q_3({}^{3}z_4\overset{3}{q_4}, z_6\overset{1}{q_1})^{3})^{2})^{1})^{0} \\ G_{q5}^{++} = ({}^{0}q_5({}^{1}z_1q_1({}^{2}z_8q_2({}^{3}z_5q_4({}^{4}z_2\overset{1}{q_5})^{4}, z_7q_4({}^{4}z_2\overset{2}{q_5})^{4})^{3})^{2}, z_3q_3({}^{2}z_4q_4({}^{3}z_2\overset{3}{q_5})^{3}, \\ z_6q_1({}^{3}z_8q_2({}^{4}z_5q_4({}^{5}z_2\overset{4}{q_5})^{5}, z_7q_4({}^{5}z_2\overset{5}{q_5})^{5})^{4})^{3})^{2})^{1})^{0} \end{cases} \tag{4}$$

Figure 2 shows the parametric structures for the expression (4).

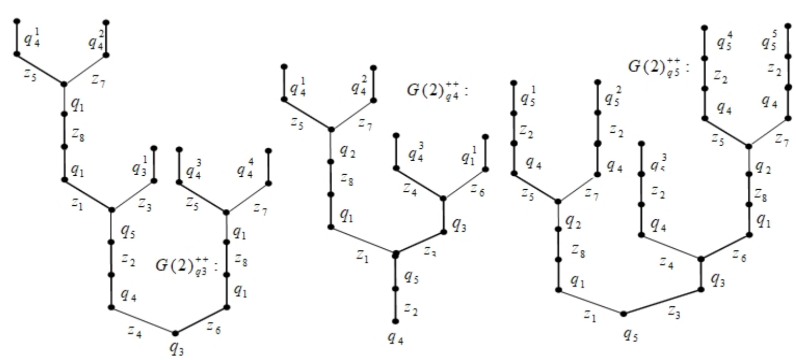

Figure 2. Game- tree structures with initial vertices: q_3, q_4, q_5.

The analysis of any *G* dependency graph comes down to determining the appropriate structure and determining the number of optimal paths. Each structure can be written as an incident matrix or an information route matrix.

3 Analysis of the automatic gearboxes with game- tree structures

The analysis of automatic gearboxes is similar to that of single planetary gears. The analysis is carried out for each run, separately introducing certain transformations of the appropriate graphs. Modern automatic transmissions are complex devices, consisting of several hundred elements of a mechanical- hydraulic system and another several hundred in the electronic module. Figure 3 shows an example of the construction scheme of a 5-speed.

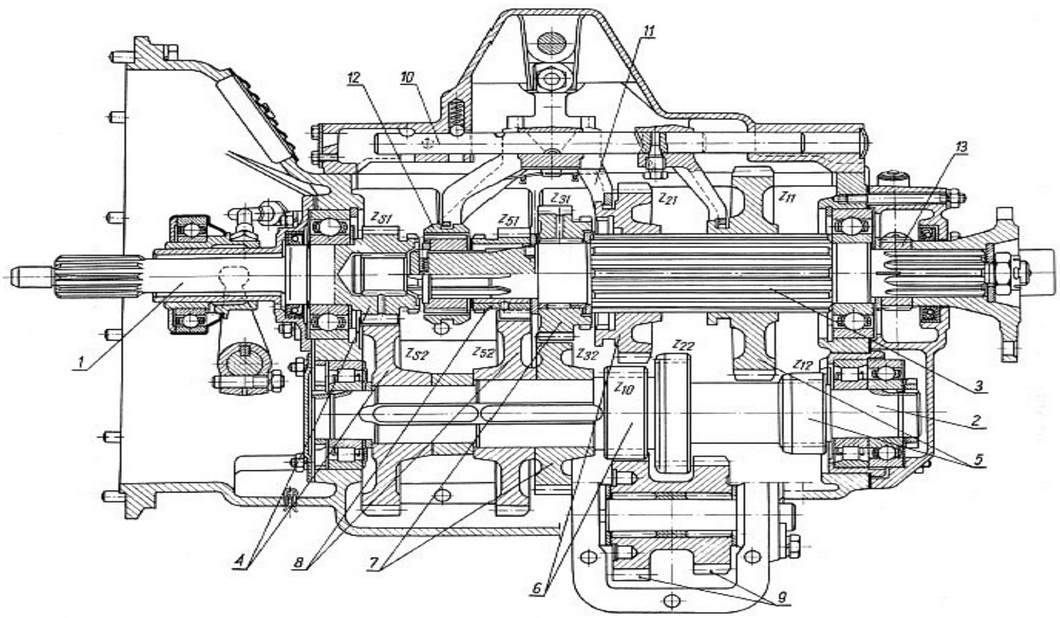

Figure 3. Construction scheme of an example 5-speed gearbox: 1- clutch shaft, 2- intermediate shaft, 3- main shaft, 4- permanent gear gears, 5- gear 1st gear, 6- gear 2nd gear, 7- gear 3rd gear, 8-gear V gear, 9- reverse gears, 10- steering mechanism guide, 11- fork, 12- gear clutch shift, 13- speedometer drive.

When enumerating kinematic structures using graph theory, identifying isomorphisms in graphs is an important and complicated problem. There are many approaches to identify isomorphic graphs, and these approaches are largely algorithmic [23]. This path is formed by the corresponding edges of the gear graph. Input and output are additionally marked.

This path allows the analysis of the sequence of transmission of rotational motion by subsequent elements of the transmission. In addition, it allows the detection of so-called redundant elements for a given gear currently under consideration. The main purpose of the kinematic analysis of the planetary gear is to determine the kinematic ratio and possibly detect oversized gears. One of the methods of analysis, based on the use of the Willis formula for any complex planetary gears, is described in detail in the textbook. In this article, the theory of contour graphs was used for kinematic analysis. In addition, redundant (oversized) wheels could be detected when plotting contour graphs.

The general algorithm for gear modeling with graphs can be described as follows (Figure 4):

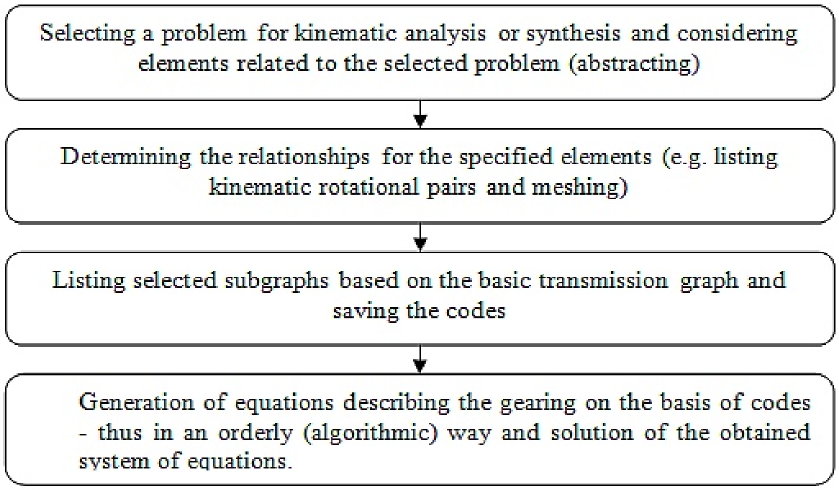

Figure 4. The general algorithm for gear modeling with graphs.

This method allowed not only to determine the kinematic ratio, i.e. the angular velocities of all wheels and gear yokes, but also to determine the angular accelerations of rotating gear elements. In addition, redundant (oversized) wheels could be detected when plotting contour graphs. The object of analysis is a complex four-speed automatic transmission with overdrive type A4LD [24]. Individual gears are implemented using brakes, clutches and backstops (free wheels) as shown in Table 1[13, 24].

Table 1. Functional matrix of the A4 ALD gearbox.

Position	DE (1 2 3 4)	1st gear	2nd gear	Reverse
cl1	( )	X	X	X
b1	(X)	-	-	-
Fw1	(X X X)	X	X	X
cl2	(X X)	-	-	X
cl3	(X X X X)	X	X	-
b2	(X)	X	-	-
b3	(X)	-	X	X
Fw2	(X)	-	-	-

The diagram of the analyzed gear is shown on Figure 5.

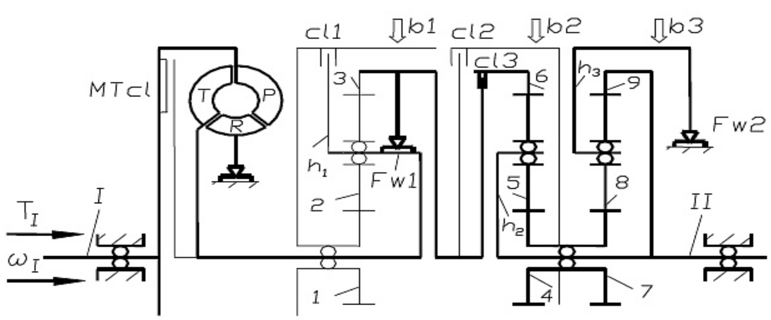

Figure 5. General diagram of the analyzed gearbox.

3.1 Generating graphs and parametric structures

Power is transmitted through the torque converter, h_1 yoke, free wheel *Fw1*, Cl_3 and Cl_2 clutches, gears *6*, *5*, *4* and yoke h_2 and in parallel through wheels *7*, *8*, *9* on output shaft II (thanks to the *Fw2* backstop activated) – Figure 5. Following the rules of building according to the contour graph methods, the dependency graph shown in Figure 6 is built for the transmission.

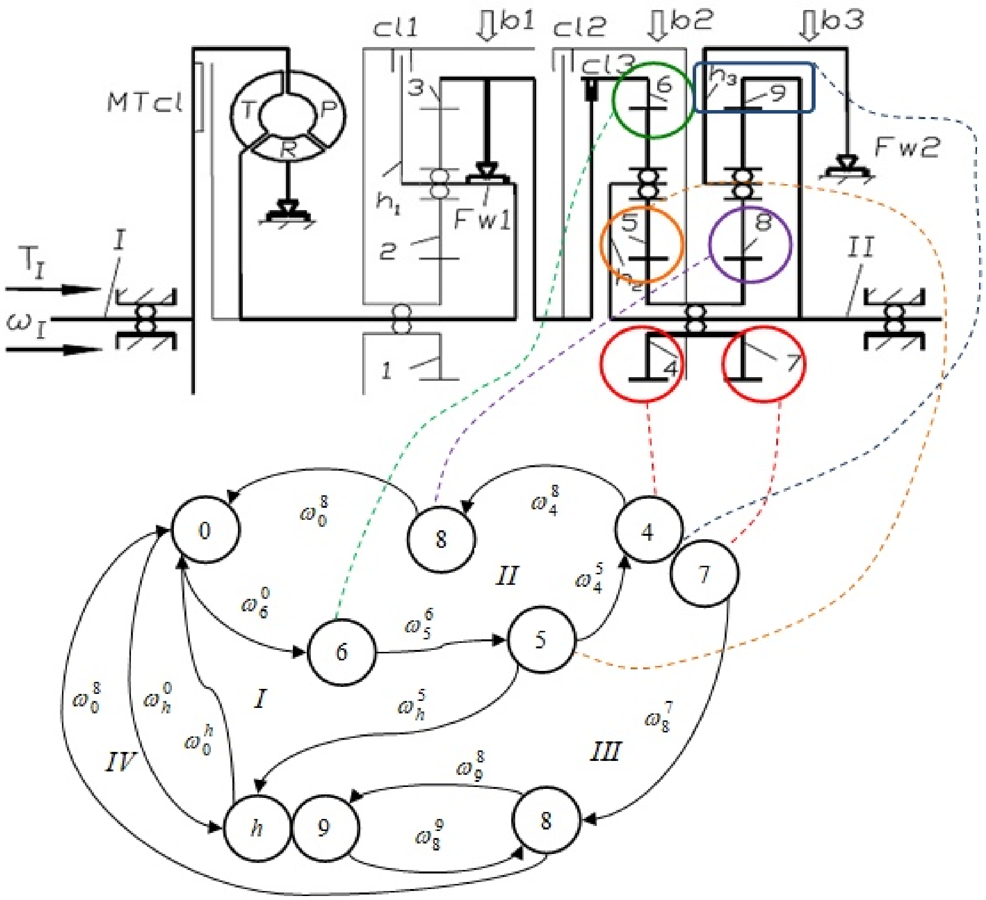

Figure 6. Signal dependency graph for DE1.

For the above graph (Figure 6), observing the appropriate algorithm, it is possible to generate a set of parametric trees:

$$\mathrm{DE1} = \left\{ G_{(q_0^{++})}, G_{(q_4^{++})}, G_{(q_5^{++})}, G_{(q_6^{++})}, G_{(q_8^{++})}, G_{(q_9^{++})} \right\} \tag{5}$$

Tree structures are shown in Figure 7.

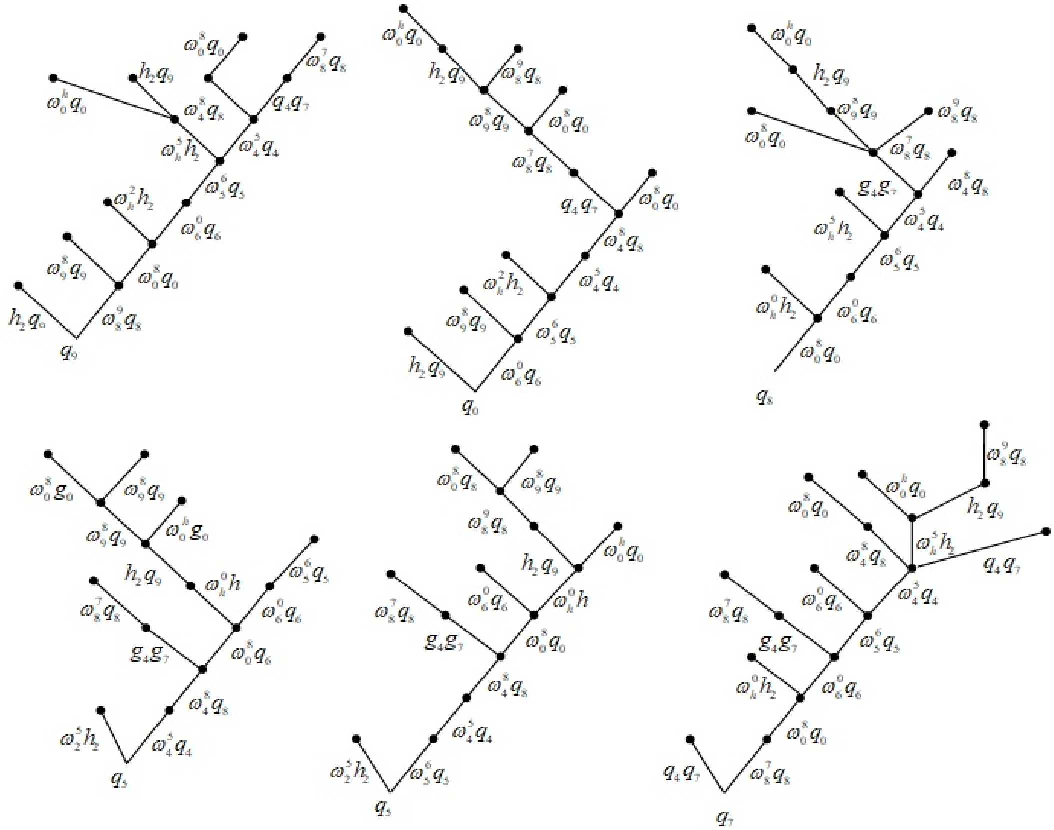

Figure 7. Game- tree structures for the dependency graph in the Figure 6.

Each of the structures has an appropriate analytical formula (G_i^{+} and G_i^{++}, where i denotes the vertex from which the decomposition of the graph was made), that clearly defines the way of transition from a dependency graph to a tree structure. The structures in Figure 7 are defined by the formulas (6- 8).

$$G_{Z0}^{++}=({}^{0}g_0({}^{1}\omega_6^0 g_6({}^{2}\omega_5^6 g_5({}^{3}\omega_4^5 g_4({}^{4}\omega_4^8 g_{8_a}({}^{5}\omega_0^8 g_0)^5, g_4 g_7({}^{5}\omega_8^7 g_{8b}({}^{6}\omega_9^8 g_9({}^{7}\omega_8^9 g_8, g_9 h_2({}^{8}\omega_0^h g_0)^8)^7, \omega_0^8 g_0)^6)^5)^4,$$
$$\omega_h^5 h_2)^3)^2, \omega_h^0 h)^1)^0 \tag{6}$$

$$G_{Z9}^{++}=({}^{0}g_9({}^{1}\omega_8^9 g_8({}^{2}\omega_0^8 g_0({}^{3}\omega_6^0 g_6({}^{4}\omega_5^6 g_5({}^{5}\omega_4^5 g_4({}^{6}g_4 g_7({}^{7}\omega_8^7 g_{8b})^7, \omega_4^8 g_{8_a}({}^{7}\omega_0^8 g_0)^7)^6, \omega_h^5 h_2({}^{6}h_2 g_9, \omega_0^h g_0)^6)^5)^4,$$
$$\omega_h^0 h_2)^3, \omega_9^8 g_9)^2, h_2 g_9)^1)^0 \tag{7}$$

$$G_{Z8}^{++}=({}^{0}g_8({}^{1}\omega_0^8 g_0({}^{2}\omega_6^0 g_6({}^{3}\omega_5^6 g_5({}^{4}\omega_4^5 g_4({}^{5}\omega_4^8 g_{8_a}, g_4 g_7({}^{6}\omega_8^7 g_{8_b}({}^{7}\omega_0^8 g_0, \omega_9^8 g_9({}^{8}h_2 g_9({}^{9}\omega_o^h g_0)^8, \omega_8^9 g_{8_b})^7)^6)^5,$$
$$\omega_h^5 h_2)^4)^3, \omega_h^0 h_2)^2)^1)^0)^2 \tag{8}$$

$$G_{Z7}^{++}=({}^{0}g_7({}^{1}\omega_8^7 g_{8_a}({}^{2}\omega_0^8 g_0({}^{3}\omega_6^0 g_6({}^{4}\omega_5^6 g_5({}^{5}\omega_4^5 g_4({}^{6}g_4 g_7, \omega_4^8 g_{8b}({}^{7}\omega_0^8 g_0)^7)^6, \omega_h^5 h_2({}^{6}\omega_0^h g_0, h_2 g_9({}^{7}\omega_8^9 g_{8_a})^7)^6)^5)^4,$$
$$\omega_h^0 h_2)^3)^2, g_4 g_7)^1)^0 \tag{9}$$

$$G^{++}_{Z6}=({}^{0}g_6({}^{1}\omega^{6}_{5}g_5({}^{2}\omega^{5}_{4}g_4({}^{3}\omega^{8}_{4}g_{8_a}({}^{4}\omega^{8}_{0}g_0({}^{5}\omega^{0}_{6}g_6,\omega^{0}_{h}h({}^{6}h_2g_9({}^{7}\omega^{9}_{8}g_{8_b}({}^{8}\omega^{8}_{0}g_0,\omega^{8}_{9}g_9)^{8})^{7},\omega^{h}_{0}g_0)^{6})^{5})^{4},a$$
$$g_4g_7({}^{4}\omega^{7}_{8}g_{8_b})^{4})^{3},\omega^{5}_{h}h_2)^{2})^{1} \tag{10}$$

$$G^{++}_{Z5}=({}^{0}g_5({}^{1}\omega^{5}_{4}g_4({}^{2}\omega^{8}_{4}g_{8_a}({}^{3}\omega^{8}_{0}g_0({}^{4}\omega^{0}_{6}g_6({}^{5}\omega^{6}_{5}g_5)^{5},\omega^{0}_{h}h({}^{5}h_2g_9({}^{6}\omega^{9}_{8}g_{8_b}({}^{7}\omega^{8}_{0}g_0,\omega^{8}_{9}g_9)^{7})^{6},\omega^{h}_{0}g_0)^{5})^{4})^{3},$$
$$g_4g_7({}^{3}\omega^{7}_{8}g_{8_b})^{3})^{2},\omega^{5}_{h}h_2)^{1})^{0} \tag{11}$$

For the graph model from the drawing, a system of contour equations of velocity $\omega_{i,i-1}$, peripheral velocities $\omega_{i,i-1}\times r_{Ai}$, angular accelerations $\varepsilon_{i,i-1}$, and tangential $\varepsilon_{i,i-1}\times r_{Ai}$ and centripetal accelerations $\omega_i^2\cdot r_{Ai,Ai+1}$ was generated (12- 15):

$$\begin{cases}\sum_{(i)}\omega_{i,i-1}=0\\ \omega_{6,0}+\omega_{5,6}+\omega_{h2,5}+\omega_{0,h2}=0\\ \omega_{6,0}+\omega_{5,6}+\omega_{4,5}+\omega_{8,7}+\omega_{0,8}=0\\ \omega_{4,5}+\omega_{8,7}+\omega_{9,8}+\omega_{5,h2}=0\\ \omega_{6,0}+\omega_{5,6}+\omega_{h2,5}+\omega_{8,9}+\omega_{0,8}=0\\ \sum_{(i)}r_{Ai}\times\omega_{i,i-1}=0\end{cases} \tag{12}$$

$$\begin{cases}\sum_{(i)}r_{Ai}\times\omega_{i,i-1}=0\\ r_6\times\omega_{5,6}+(r_4+r_5)\times\omega_{h2,5}=0\\ r_6\times\omega_{5,6}+r_4\times\omega_{4,5}+r_7\times\omega_{8,7}+r_9\times\omega_{9,8}=0\\ r_4\times\omega_{4,5}+r_7\times\omega_{8,7}+r_9\times\omega_{9,8}+(r_4+r_5)\times\omega_{5,h2}=0\\ r_6\times\omega_{5,6}+(r_4+r_5)\times\omega_{h2,5}+r_9\times\omega_{8,9}=0\end{cases} \tag{13}$$

$$\begin{cases}\sum_{(i)}\varepsilon_{i,i-1}=0\\ \varepsilon_{6,0}+\varepsilon_{5,6}+\varepsilon_{h2,5}+\varepsilon_{0,h2}=0\\ \varepsilon_{6,0}+\varepsilon_{5,6}+\varepsilon_{4,5}+\varepsilon_{8,7}+\varepsilon_{0,8}=0\\ \varepsilon_{4,5}+\varepsilon_{8,7}+\varepsilon_{9,8}+\varepsilon_{5,h2}=0\\ \varepsilon_{6,0}+\varepsilon_{5,6}+\varepsilon_{h2,5}+\varepsilon_{8,9}+\varepsilon_{0,8}=0\end{cases} \tag{14}$$

$$\begin{cases}\sum_{(i)}r_{Ai}\times\varepsilon_{i,j-1}-\omega_i^2\cdot r_{Ai,Ai+1}=0\\ r_6\times\varepsilon_{5,6}+(r_4+r_5)\times\varepsilon_{h2,5}=0\\ r_6\times\varepsilon_{5,6}+r_4\times\varepsilon_{4,5}+r_7\times\varepsilon_{8,7}+r_9\times\varepsilon_{9,8}=0\\ r_4\times\varepsilon_{4,5}+r_7\times\varepsilon_{8,7}+r_9\times\varepsilon_{9,8}+(r_4+r_5)\times\varepsilon_{5,h2}=0\\ i_{6,9}=\omega_{6,0}/\omega_{9,0}=\\ r_6\times\varepsilon_{5,6}+(r_4+r_5)\times\varepsilon_{h2,5}+r_9\times\varepsilon_{8,9}=0\end{cases} \tag{15}$$

Where, in the equations, the designations mean successively: $\omega_{i,i-1}$ - relative angular velocity vector of the element i relative to the previous element *i-1*, $\omega_{i,0}$ - vector of the absolute angular velocity vector of the element (relative to the fixed base), $r_{Ai}=r_{OAi}$ - the radius of the vector of point A_i (point *A* on element *i*), -

$r_{Ai,Ai+1} = r_{Ai+1} - r_{Ai}$, $\varepsilon_{i,j-1}$ - vector of relative angular acceleration of the element relative to the previous element *i-1*.

4 Game- tree structures in determining the optimal number of teeth

In the computer program, searching of parametric game trees takes place as a combination of graph searching. Parametric game structures represent iterative depth search. They combine effective use of space in Depth First Traversal or Search and fast Breadth First Search (for nodes closer to the root), taking account of cycles (returns) [25]. The search of parametric game structures calls a DFS (Depth First Search or Traversal) algorithm for different depths, beginning from the initial value. At each call, DFS cannot exceed a specified depth. Thus in effect we perform DFS in the style of BFT (Breadth First Traversal or Search).

Preliminary algorithm:

```
/ Returns the value of z and if the target is reachable z
// src within the bounds of max_depth
bool IDDFS(src, target, max_depth)
    for limit from 0 to max_depth //take into account the initial and design
conditions
       if DLS(src, target, limit) == true
           return true
    return false
bool DLS(src, target, limit)
    if (src == target)
        return true;
    // If reached the maximum depth,
    // stop recursing.
    if (limit <= 0)
        return false;
    foreach adjacent i of src
        if DLS(i, target, limit?1)
            return true
```

An important point is that the lowest-level nodes are visited multiple times through return cycles. The last level (or maximum depth) is visited once, the next-to-last level is visited twice, and so on. Two cases may occur:

a) When the graph has no cycle; this case is simple [26]. We can use DFS multiple times with different height limits. Most frequently at the initial levels of the structure.

b) When the graph has cycles. Most frequently at the end nodes of the structure.

The algorithm operates parametrically. The initial number of teeth is input:

The input attributes (we) are the numbers of teeth $z_1, z_2, z_3, z_4, z_5, z_6, z_7, \ldots, z_k$

The values $h_1, h_1, \ldots h_m$ are respectively hypotheses represented by the constraints. These are the output values (out) that determine the tooth values.

For searching, ranges of values for the number of teeth searched are assumed: $z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9$:

$z_1 \in (0,1,2,...,100)$, $z_2 \in (0,1,2,...,100)$, $...z_9 \in (0,1,2,...,100)$.

The values $h_1, h_1, \ldots h_m$ are determined at the output of the parametric structure. Each structure corresponds to a gear wheel and determines the optimal number of teeth. In general, there are sets of generated teeth for each of the structures:

$$\begin{cases} G_{(q_1^{++})} = \left\{ h_{1q_1}(z_{i(h1)}, z_{i+1(h1)}, \ldots, z_{j(h1)}), h_{2q_1}(z_{i(h2)}, z_{i+1(h2)}, \ldots, z_{j(h2)}), \ldots, h_{jq_1}(z_{i(hj)}, z_{i+1(hj)}, \ldots, z_{j(hj)}) \right\} \\ G_{(q_2^{++})} = \left\{ h_{1q_2}(z_{i(h1)}, z_{i+1(h1)}, \ldots, z_{j(h1)}), h_{2q_2}(z_{i(h2)}, z_{i+1(h2)}, \ldots, z_{j(h2)}), \ldots, h_{jq_2}(z_{i(hj)}, z_{i+1(hj)}, \ldots, z_{j(hj)}) \right\} \\ G_{(q_3^{++})} = \left\{ h_{1q_3}(z_{i(h1)}, z_{i+1(h1)}, \ldots, z_{j(h1)}), h_{2q_3}(z_{i(h2)}, z_{i+1(h2)}, \ldots, z_{j(h2)}), \ldots, h_{jq_3}(z_{i(hj)}, z_{i+1(hj)}, \ldots, z_{j(hj)}) \right\} \\ \vdots \\ G_{(q_8^{++})} = \left\{ h_{1q_3}(z_{i(h1)}, z_{i+1(h1)}, \ldots, z_{j(h1)}), h_{2q_3}(z_{i(h2)}, z_{i+1(h2)}, \ldots, z_{j(h2)}), \ldots, h_{jq_3}(z_{i(hj)}, z_{i+1(hj)}, \ldots, z_{j(hj)}) \right\} \end{cases} \tag{16}$$

Figure 8 shows an example of searching the optimal number of teeth for a game trees structure G_{q0}^{++}.

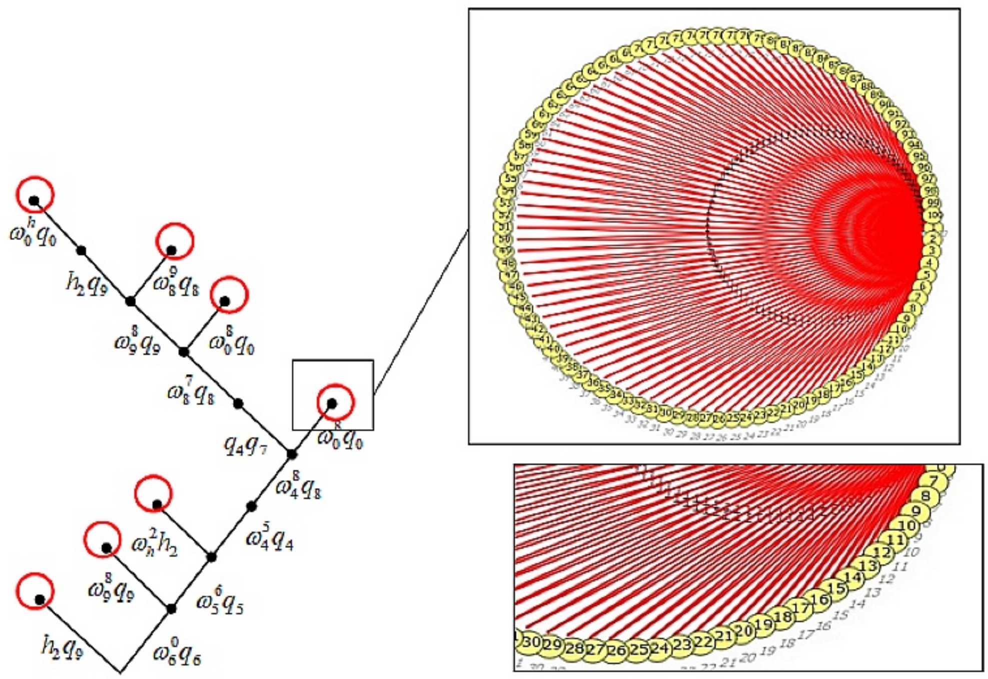

Figure 8. An example of searching of the optimal number of teeth for a game- tree structure G_i^+.

For a given gear design, there are groups of three gears with the same number of teeth: $I: z_1 = z_4 = z_7$; $II: z_2 = z_5 = z_8$; $\mathrm{III}: z_3 = z_6 = z_9$.
For group I, the optimal number of teeth: n= 18
For group I, the optimal number of teeth: n= 27
For group I, the optimal number of teeth: n= 36

For the module value: $m = 2$ and:

- input angular speed (after commissioning): $\omega_{6,0} = 377 rad/s$.
- input angular acceleration (during 8 s start-up): $\mathcal{E}_{6.0}$= 47.1 rad / s^2 the output angular velocity $\omega_{9,0} = 153,2 rad/s$ and $\mathcal{E}_{9,0}$=19,15rad/s^2.

The analysis and calculations should be carried out for the other *DE1, DE2, DE3, DE4* gears and for reverse gear. In the general notation, the algorithm for generating the optimal number of teeth is presented in Figure 9.

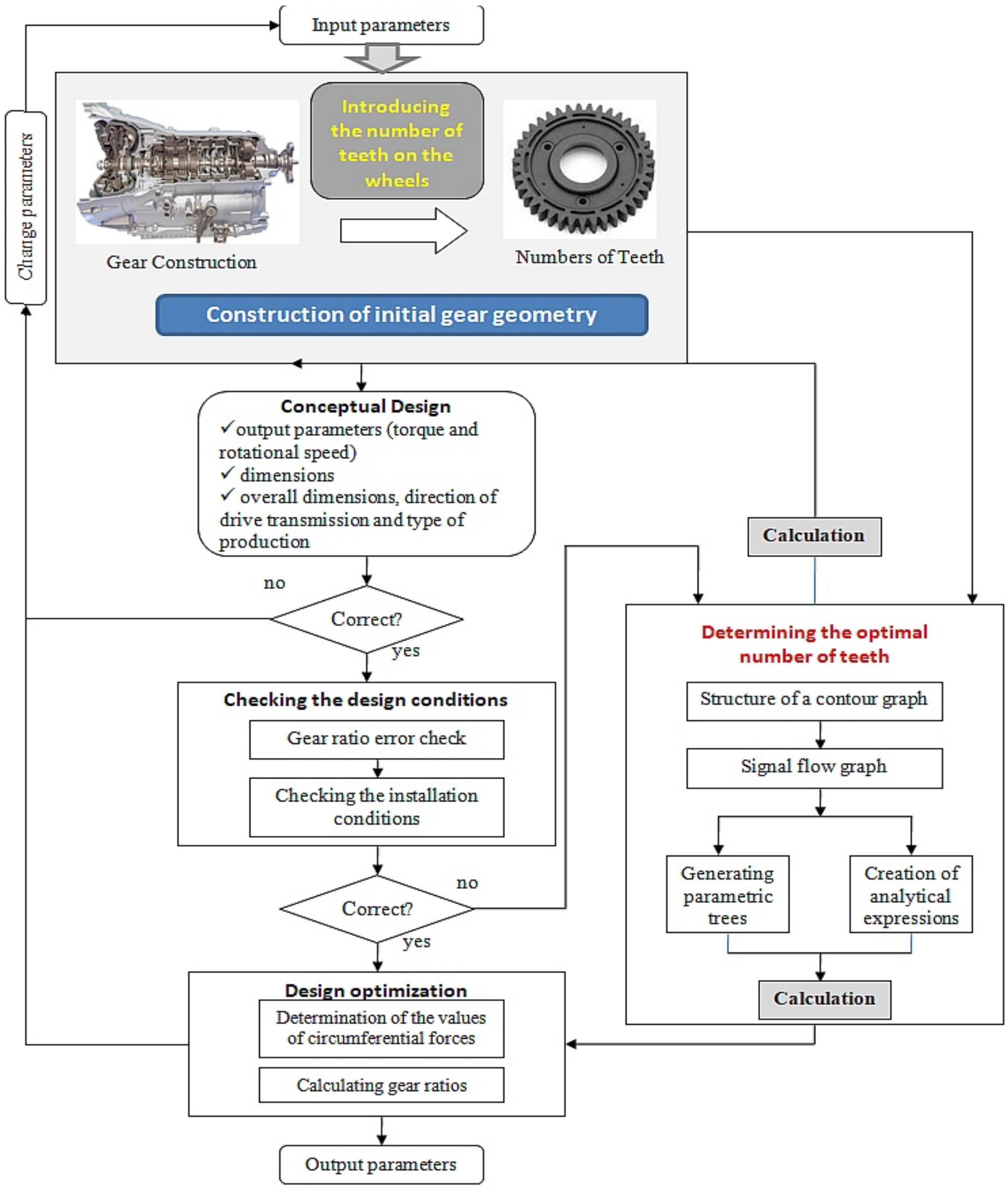

Figure 9. Concept of general algorithm for optimisation of transmission system with selection of appropriate number of teeth.

5 Results and discussion

Unlike traditional relation diagrams and tree classifiers, game trees parametrically link the importance of nodes (states) to the height of the tree structure. This approach differs from previous literature on parametric automata and their applications in terms of control systems, operating systems, natural language level knowledge representation, programming the behavior of cybernetic systems, etc. In the previous literature studies, considering graphical methods and game automata, the following parameters have been considered

- knowledge base using graphs,
- finite automaton operations based on symbolic expressions,
- dendrites representing the prognostic game,
- paths in the game dendrite describing the future system development forecast, semaphores.

The advantages of graph methods are: algorithmic approach to problems and the ability to perform other tasks, e.g. algorithmic finding of redundant wheels or enumeration of design solutions. In the above approach, parametric structures can better reflect the algorithmic capabilities of a given gear. Therefore, the overall transmission optimization process should take into account all conditions. Currently, only the optimal number of teeth has been focused. Most of these rigid parameters that characterize gears best, including planetary gears, can be virtually entirely reduced to an absolutely dimensionless, extensive form.

The number of teeth belongs to rigid, specific parameters used in computer aided design. Graph in the sense of substantive graph theory is associated with many other separate algebraic structures such as matrices, matroids and structural numbers, linear spaces of cut-offs. These objects enable coding of the transmission structure, which of course allows the use of highly advanced algorithms, the so-called artificial intelligence: evolutionary, genetic or immunological.

6 Conclusion

The graph- based methods of analysis and synthesis of planetary gears provide an alternative for the accomplishing of the tasks in question. The automatic transmission presented in the article was modeled with the use of signal flow graphs. Due to the use of the new modelling method, there is a need to analyze the technical risk of such a solution. An element necessary in the risk assessment process is to take into account the requirements of the engineering design methodology. In this regard, it is worth choosing a design methodology that meets the following criteria:

- Completeness of technical criteria,
- Adequacy of a set of parameters describing the designed object.

The criteria formulated for assessment should be synthetic because it increases the objectivity of assessment of a given solution. Designing facilities involves increasing efficiency (to some extent) while reducing the efficiency of another team. Determining effectiveness consists in comparing the effects of a given measure with its expenditure. Particularly in this respect, it should be ensured that the risk of adopted boundary standards in the scope of e.g. costs is not exceeded. The article presents the application of dependency graphs to determine the optimal number of gear teeth for an automatic transmission. Each gearbox (automatic or manual planetary gear) can be additionally modeled with Hsu graphs [19].

References

[1] A. Regnier, “Les infortunes de la raison”, Seuil, 1966

[2] B. Roy, “Multi-criteria decision support“, *Scientific and Technical Publishing House*, Warsaw, 1996

[3] A. Deptuła and M. A. Partyka, “Application of dependence graphs and game trees for decision decomposition for machine systems”, *Journal of Automation, Mobile Robotics & Intelligent Systems*,5, No.3, pp.17-26, 2011

[4] C.A, Nelson and R. J. Cipra, “Simplified kinematic analysis of bevel epicyclic gear trains with application to powerflow and efficiency analyses*”, J Mech Des*, 127, p.278–286, 2005

[5] Y. Xiaodong Yu, Z. Jian ,F. Chengyu and C. Sheng, “Stability analysis of governor-turbine-hydraulic system by state space method and graph theory”, *Energy*,Volume 114,Pages 613-622, 2016

[6] N.F. Timerbaev, A.R. Sadrtdinov and R.G. Safin, “Software Systems Application for Shafts Strength Analysis in Mechanical Engineering”, *Procedia Engineering*, Volume 206, pp. 1376- 1381, 2017

[7] J. Xu, , G. Sun, X. Ou, J. Xu, H. Lu, P. Su, P., . . .and M. Ding, “The Construction of Design Science Knowledge Graphs - Based on National Natural Science Foundation of China. Proceedings of the Design Society”, *International Conference on Engineering Design*, 1(1), 2607-2616. doi:10.1017/dsi.2019.267, 2019

[8] F. E. Cellier, F.E., “Object-oriented Modeling of Mechatronics Systems in Modelica Using Wrapped Bond Graphs“. *–Proc. IEEE Intl. Conference on Mechatronics*, Malaga, Spain 2009

[9] S.J. Lee and P.H. Chang, “Modeling of a hydraulic excavator based on bond graph method and its parameter estimation“., *J Mech Sci Technol* 26, 195–204, 2012, https://doi.org/10.1007/s12206-011-0938-2

[10] H.S. Yan and Y.T Chiu, “On the number synthesis of kinematic chains“, *Mechanism and Machine Theory*, Volume 89, 2015, Pages 128-144, ISSN 0094-114X, https://doi.org/10.1016/j.mechmachtheory.2014.08.012

[11] H.F. Ding, Z. Huang and Z. Cao, “The topological graphs creation automatically of kinematic chains and atlas database establishment“, *Chin J Mech Eng*, 42, pp. 32–36; 2006

[12] E.Pennestrì and N.P. Belfiore, “On Crossley's contribution to the development of graph based algorithms for the analysis of mechanisms and gear trains“, *Mechanism and Machine Theory*,Volume 89, 2015, Pages 92-106, ISSN 0094-114X

[13] A.Deptula, “Application of graphic decision structures in the design and management methodology. T. 3, Graphical and decision-making graphs of multivalent logical trees in the analysis of planetary gears, *Studies and Monographs, Issue No. 539, Publishing House of the Opole University of Technology*, ISBN 978-83-66033-79-5, Opole 2020, number of pages 238

[14] A. Deptuła and M. A. Partyka, “Application of dependence graphs and game trees for decision decomposition for machine systems”, *Journal of Automation, Mobile Robotics & Intelligent Systems*,5, No.3, pp.17-26, 2011

[15] A. Deptuła, J. Drewniak and M. A. Partyka M.A, “Application of dependence graphs and game trees in analysis of a planetary gear modeled with a contour graph”, *Machine Dynamics Research*, 2017, 41, pp.105-114, 2017

[16] Hsu C.H, Lam K.T. and Lin Y.L, „Automatic synthesis of displacement for planetary gear trains“*, Math., Comput., Modelling,* Vol. 19, No. 11, pp. 67- 81, 1994

[17] Uematsu S., „An application of graph theory to the kinematic analysis of planetary gear trains“, *Int. Journal of the Japan Soc. for Precis. Eng*., No. 31, pp. 141-146, 1997

[18] C.H. Hsu, “An analysis methodology for the kinematic synthesis of epicyclic gear mechanisms”, *ASME Journal of Mechanical Design*,124, pp. 574-576, 2002

[19] C.H. Hsu, “Systematic enumeration of epicyclic gear mechanisms for automotives“, *JSME Int J C* 42: 225–233, 1999

[20] F. Freudenstein, “An application of boolean algebra to the motion of epicyclic drives“. *J Eng Ind*, 93: 176–182. 52, 1971

[21] F. Freudenstein and A.T. Yang, “Kinematics and statics of a coupled epicyclic spur-gear train“, *Mech Mach Theory*, 263–275, 1972

[22] D. Marghitu, “Kinematic chains and machine components design”, *Elsevier Amsterdam*, San Diego, Academic Press, London, 2005

[23] L.P. Laus, H. Simas and D. Martins, “Efficiency of gear trains determined using graph and screw theories“, *Mech Mach Theory*, 52: 296–325, 2012

[24] J. Drewniak, S. Zawiślak and A. Wieczorek, “Planetary kinematics analysis method of an automatic gearbox”, *Scientific Notebooks of the Silesian University of Technology*, *Series: Transport*, Vol. 83, No. 1904, pp. 65-71, 2014

[25] A. C. Rao and P. B. Deshmukh, “Computer aided structural synthesis of planar kinematic chains obviating the test for isomorphism*”, Mech Mach Theory*, 36, pp. 489–506, 2003

[26] M.S. Pishave and M. Rabbani, “A graph theoretic-based heuristic algorithm for responsive supply chain network design with direct and indirect shipment”, *Advances in Engineering Software* 42, pp. 57–63, 2011