UNIWERSYTET TECHNOLOGICZNO-PRZYRODNICZY
IM. JANA I JĘDRZEJA ŚNIADECKICH W BYDGOSZCZY
ZESZYTY NAUKOWE NR 262
ELEKTRONIKA I TELEKOMUNIKACJA 17 (2013), 37-73

REFERENTIAL GRAPHS

Damian Ledziński, Beata Marciniak, Mścisław Śrutek

University of Technology and Life Sciences in Bydgoszcz
[damian.ledzinski][beata.marciniak][mscislaw.srutek]@utp.edu.pl

Summary: The authors of this paper have defined the notion of referential graphs which allow to model data and telecommunication networks in order to optimize them. Parameters of this type of graphs can be compared with parameters of modified chordal rings three and four degree. A description of the program developed for graphs of this type and its effects have been presented.

Keywords: graphs, chordal rings, data communication networks.

1. INTRODUCTION

The research subject of this paper is to present a method for optimization of network structures built on the basis of chordal rings (multi top networks) which connect nodes forming data communication distributed systems. The term 'distributed system' is understood as a data communication structure consisting of a certain number of identical, intelligent nodes who are responsible for providing the system users with specific services ensuring adequate quality, speed and reliability [1,2,3,4,5]. Acceptance of such a solution positively contributes to reduction of investment costs, facilitates operation and maintenance of data communication systems as well as multi-process systems whose operation involves the concept of parallel computing processes carried out by a set of processes connected by a network with adequate configuration. The right choice of the topology of connections between its components is of key importance for the design and analysis of distributed computer systems. It determines effectiveness of the whole system operation.

In case of distributed computing systems, the main purpose is to reach the maximum computing power, whereas for telecommunication system it is of key importance to reach [4]:

- Minimal Connection Costs – expressed by a summary number of used links;
- Minimal Communication Delay – it is measured by diameter length and average path length;
- High fault tolerance which is characterized by the number of independent paths between two nodes – connectivity or the minimal number of nodes and edges after removal of which the network ceases to be coherent Node and Edge Connectivity;
- Regularity and Symmetry;
- Easy of Routing [6,7];
- Extensibility.

The main elements of distributed systems are: the type of transmission medium and configuration of the network connecting modules used for collecting, transmission and distribution of data [8].

The discussed networks can be modeled using undirected graphs, and more specifically, using Symmetric Digraphs [9]. A characteristic feature of undirected graph G with a set of vertices $V(G)$ and a set of edges $E(G)$, is that if edge $[v_i, v_j]$ belongs to set $E(G)$, then also edge $[v_j, v_i]$ belongs to the same set. Thus, the diagraph edge connecting nodes v_i and v_j can be replaced with two directed edges $[v_i, v_j]$ and $[v_j, v_i]$, which, in implementation of an optical network, corresponds to a connection of two random nodes by two optical fibers, one of which to be used for transmitting signals between v_i and v_j, whereas the other – between nodes v_j and v_i.

It is obvious that the network described by complete graph has the best transmission properties. Such a network is the highly reliable (in case of node or link damage the most bypass paths are available), it has the shortest graph diameter (equal to 1), the lowest average connection path length (equal to 1) and the lowest connectivity delay.

The main fault of a complete graph involves a necessity of physical or logical connection of all the nodes with each other which is specified by the formula:

$$L = \frac{p(p-1)}{2} \tag{1}$$

where:

L – denotes the number of links,
p – the number of nodes.

For this reason, the most desirable is such a topology which will provide the nodes with connectivity in the most satisfying way, ensuring reliability, speed and quality of provided services with economically justified outlays and implementable technical solutions.

Conclusions drawn on the basis of the studied papers indicate that the parameters which have the largest influence on the network operation properties are the graph diameter and the average path length.

Definition 1

The diameter of graph $D(G)$ is the longest path among the minimal length paths connecting two arbitrary nodes of a graph:

$$D(G) = \max_{v_i v_j} \{d(v_i, v_j)\} \tag{2}$$

where v_i, v_j denote the number of nodes, $d(v_i,v_j)$ – length of the path (number of nodes) connecting nodes v_i, v_j.

Definition 1

The average path length in a graph is defined as follows:

$$d_{av} = \frac{1}{p(p-1)} \sum_{i=0}^{p-1} \sum_{j=0}^{p-1} d_{\min}(v_i, v_j) \tag{3}$$

where d_{min} denotes the minimal number of edges between nodes v_i, v_j, whereas $i \neq j$, p – is the number of nodes forming a graph.

If a graph is symmetric, the average path length is expressed by the formula:

$$d_{av} = \frac{1}{(p-1)} \sum_{j=0}^{p-1} d_{\min}(v_0, v_j) \text{ [10].} \tag{4}$$

A ring is a base structure of most data communication systems. Commutation modules or computers can be the system nodes, whereas two-direction transmission channels are the edges. This kind of network is described by the Hamilton one cycle digraph [4]. Thus, there are always two straight routes connecting two random nodes of the network. In practice, the links between nodes are provided using a two optical fiber, in one of which the transmission is clockwise and in the other it is counter clockwise. Acceptance of such a solution increases reliability of data transmission as damage to one of the nodes or links does not involve a collapse of the entire system, although it does impair its operation. Another advantage of such a structure is standardization of nodes, small number of inter-node links and good extensibility, the disadvantage though is poor transmission parameters of this type of topology (diameter, average path length). Numerous publications contain instructions how to improve the above properties of networks built on the basis of rings proposing introduction of additional links, called chords. The structures developed in this way are referred to as chordal rings.

The notion "chordal ring" was introduced by Arden and Lee in their work [11], where they describe application of this type of topology for building a network of multi-computers. They have accepted the assumption that this structure will be described by a regular degree three graph.

Definition of degree three chordal rings is as follows:

Definition 2

A degree three chordal ring (Fig. 1) is a ring in which each odd number node v_i, $i \in \{1, 3, 5, \ldots, p-1\}$ is additionally connected with node $v_{(i+q) \bmod p}$, and each even number node v_j, $j \in \{0, 2, 4, \ldots, p-2\}$ is connected with node $v_{(j-q) \bmod p}$. Value p denotes the number of nodes forming a ring and needs to be of even number (which results from the fact that each pair of nodes is connected by a chord), whereas $q \leq p/2$ is the chord length and is an odd value equal to the multitude of the edge forming a ring. It has been accepted that this kind of structure is described by notation CHR3(p; q).

A more general definition is included in paper [12].

Definition 3

A chordal ring is a special case of a Circulant Graph defined by pair (p, Q), where p denotes the number of nodes and Q a set of chords, $Q \subseteq \{1, 2, \ldots, \lfloor p/2 \rfloor\}$. Each chord $qi \in Q$ connects each pair of nodes forming a ring, whereas a chordal ring is described by notation $G(p; q_1, \ldots, q_i)$, and $q_1 = 1 < q_2 < \ldots < q_i$. According to the rule the degree of nodes is equal to $d(V) = 2i$, except for the case when the chord length is $p/2$ when p is even and the node degree is $2i - 1$.

The above given definition refers to graphs of chordal type in which each chord is an element forming the Hamilton cycle.

Chordal rings of this type can be found in a series of publications and on this basis the concepts of optimal and ideal graphs have been introduced [9,13,14].

Definition 4

An ideal chordal ring of $d(V)$ degree is a regular graph with pi nodes which is described by the formula:

$$p_i = 1 + \sum_{d=1}^{D(G)-1} |p_d| + |p_{D(G)}| \tag{5}$$

where p_d denotes the number of nodes belonging to the d-th layer (layer is a subset of nodes equally distant from a randomly chosen source node with d edges), $p_{D(G)}$ denotes the number of the remaining nodes which belong to the last layer and $D(G)$ is the diameter of the analyzed graph. For each n and $m < D(G)$ $p_n \cap p_m = \emptyset$. If subset $p_{D(G)}$ reaches the maximum value possible for the last layer, then such a graph is considered to be optimal.

Average path length d_{avi} for an ideal chordal ring is defined by the expression:

$$d_{avi} = \frac{\sum_{d=1}^{d(G)-1} d|p_d| + D(G)|p_{d(G)}|}{p_i - 1} \tag{6}$$

whereas in an optimal graph d_{avo} is equal to:

$$d_{avo} = \frac{\sum_{d=1}^{d(G)} d\ p_d}{p_o - 1} \tag{7}$$

where:

d – the layer number,
p_d – number of nodes in the d-th layer
p_o – number of the optimal graph nodes.

Optimal and ideal nodes are characteristic of a selected type of graphs and their theoretically computed basic parameters will be used for evaluation of parameters of graphs obtained in real world systems.

According to the analyzed papers and the authors' own experiments, it has been found that it is possible to build modified structures with better properties than commonly known chordal rings on the basis of rings [15,16,17,18,19,20]. In order to find out whether the proposed structures are characterized by the best possible parameters it is necessary to define referential graphs whose parameter values will provide the lowest achievable limit. For this purpose the notion of Referential Graph has been introduced.

The minimal spanning tree comprising all p nodes of a regular graph was used as the base for construction of Referential Graphs structures and determination of their parameters (Fig. 1), and its parameters – radius and the average path length, computed from a randomly chosen source node, were accepted as values of the Referential Graph parameters – the diameter and the average path length.

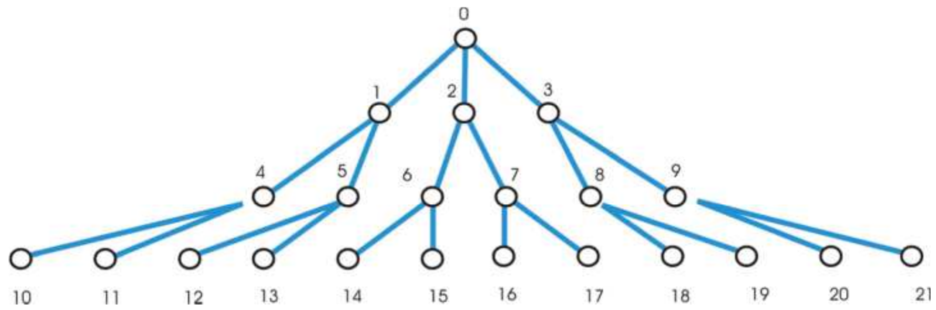

Fig. 1. A sample tree used for construction of an optimal Referential Graph third degree

The notion of Referential Graph has been defined on the basis of a carried out theoretical analysis, accounting for the degree of its nodes.

Definition 5

Referential graph of $d(V)$ degree is called a regular structure with the following properties:

– Node number p_{doGR} in an optimal graph d-th layer is defined by the formula:

$$p_{doGR} = d(V)\cdot(d(V)-1)^{d-1} \tag{8}$$

– In an ideal graph this number refers to all layers $d < D(G)$, whereas in layer $d = D(G)$ this number is :

$$p_{diGR} = p_i - d(V)\cdot(d(V)-1)^{d-2} \tag{9}$$

– Number of nodes $p_{D(G)GR}$ in an optimal graph in the function of its graph diameter:

$$p_{D(G)_{GR}} = \frac{d(V)\cdot(d(V)-1)^{D(G)}-2}{d(V)-2} \tag{10}$$

– The graph diameter in the function of the number of nodes forming a graph of this type is expressed by the formula:

$$D(G) = \log_{d(V)-1}\frac{p_{GR}\left(d(V)-2\right)+2}{d(V)} \tag{1}$$

– In an ideal graph the diameter is defined by the following expression:

$$D(G) = \left\lceil \log_{d(V)-1}\frac{p_{GR}\left(d(V)-2\right)+2}{d(V)} \right\rceil \tag{2}$$

– The mean value of path length d_{avGR} in the function of the optimal graph diameter is:

$$d_{avo} = \frac{d(V)\dfrac{\left(d(V)-1\right)^{D(G)}\cdot\left(\left(d(V)-2\right)\cdot D(G)\cdot-1\right)+1}{\left(d(V)-2\right)^2}}{\dfrac{d(V)\cdot(d(V)-1)^{D(G)}-2}{d(V)-2}-1} = \frac{\left(d(V)-1\right)^{D(G)}\cdot\left(\left(d(V)-2\right)\cdot D(G)\cdot-1\right)+1}{\left(d(V)-2\right)\cdot(d(V)-1)^{D(G)}-1} \tag{3}$$

The average path length of an ideal graph is defined by the expression:

$$d_{avi} = \frac{d(V)\dfrac{\left(d(V)-1\right)^{D(G)-1}\cdot\left(\left(d(V)-2\right)\cdot\left(D(G)-1\right)\cdot-1\right)+1}{\left(d(V)-2\right)^2}+\left(p_{iGR}-p_{(D(G)-1)GR}\right)\cdot D(G)}{p_{iGR}-1} \qquad (4)$$

where p_{iGR} denotes the number of the deal graph nodes.

– A graph is symmetric, meaning that its parameters remain identical no matter which node they are computed from.

2. COMPARISON OF PARAMETERS OF MODIFIED CHORDAL RINGS AND REFERENTIAL GRAPHS

In this section third and fourth degree Referential Graphs will be presented including comparison of their parameters with parameters of the earlier studied best modified structures.

2.1. Third degree referential graphs

The basis for determination of the main parameters for graphs of each type is provided by a description of the distribution of nodes that occur in their successive layers. For degree three Referential Graphs, theoretical distributions of the maximal number of nodes have been determined, on the basis of a minimum spanning tree comprising all the nodes that form the structure which is shown in Table 1.

Table 1. Distribution of the maximal number of nodes occurring in successive layers of Referential Graphs

d	1	2	3	4	5	6	7	8	9	10
p_{dRF}	3	6	12	24	48	96	192	384	768	1536

On the basis of the above presented table, the summary number of nodes that occur in optimal graphs in the function of their diameter, has been calculated which is presented in table 2.

Table 1. Summary number of nodes occurring in an optimal Referential Graph third degree

$D(G)$	1	2	3	4	5	6	7	8	9	10
p_{dRF}	4	10	22	46	94	190	382	766	1534	3070

A third degree optimal Referential Graph has been defined on the basis of the obtained values presented in the tables.

Definition 6

A three degree optimal Referential Graph (CHR_{RG3}) is a regular structure with the following properties:

– Number of nodes p_{dGR3} in the d-th layer is defined by:

$$p_{d_{GR3}} = 3 \cdot 2^{(d-1)} \tag{5}$$

– Number of nodes $p_{D(G)GR3}$ in the graph diameter function is:

$$p_{D(G)_{GR}3} = 3 \cdot 2^{D(G)_{GR3}} - 2 \tag{6}$$

– The graph diameter in the function of the number of nodes forming its structure is expressed by:

$$D(G) = \log_2 \frac{p_{D(G)_{GR3}} + 2}{3} \tag{7}$$

– The mean path length d_{avGR3} in the function of its diameter is:

$$d_{avGR3} = 3 \frac{\left(D(G)_{GR3} - 1\right) \cdot 2^{D(G)_{GR}3} + 1}{3 \cdot \left(2^{D(G)_{GR3}} - 1\right)} . \tag{8}$$

Charts in Fig. show a comparison between the parameters of modified structures CR3m and CR3n having better properties than the discussed in work [21] standard chordal rings, and the parameters of Referential Graphs.

Chordal rings CR3m i CR3n shown in Fig. 2 and 3 are defined in the following way:

Definition 7

Modified chordal ring CR3m is a ring whose each node with number v_i (i = 0 *mod* 4) is connected with node $v_{(i+q1) \bmod p}$, and each node v_j (j = 2 mod 4) is connected with node $v_{(i+q2) \bmod p}$; where p denotes the number of nodes in a ring that needs to be divisible by 4, whereas q_1 i q_2 are chord lengths which need to be even and satisfy condition $3 \le q_i \le p - 3$. Values p, q_1 and q_2 define the structure (graph) CR3m (p; q_1,q_2).

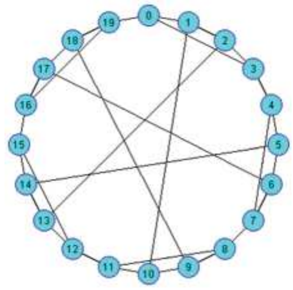

Fig. 2. An example of structure CR3m(20; 3,9)

Definition 8

Chordal ring CR3n is a ring where each node with number v_i (i = 0 mod 4) is connected with node $v_{(i+q1) \bmod p}$, node v_j (j = 1 mod 4) is connected with node $v_{(j-q1) \bmod p}$, v_k (k = 2 mod 4) with node $v_{(k+q2) \bmod p}$, a v_l (l = 3 mod 4) with node $v_{(l-q2) \bmod p}$. Values p denoting the number of nodes forming a ring is divisible by 4, q_1 and q_2 are chord lengths which fulfill condition $3 < q_i < p/2$. This graph is defined as CR3n (p; q_1,q_2).

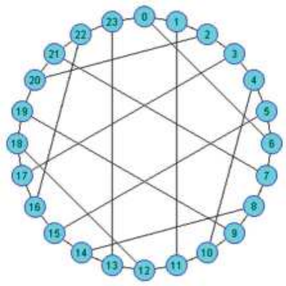

Fig. 3. An example of graph CR3n(24; 6,10)

In Figure 4 there are comparative charts for parameters of the modified graphs and Referential Graphs.

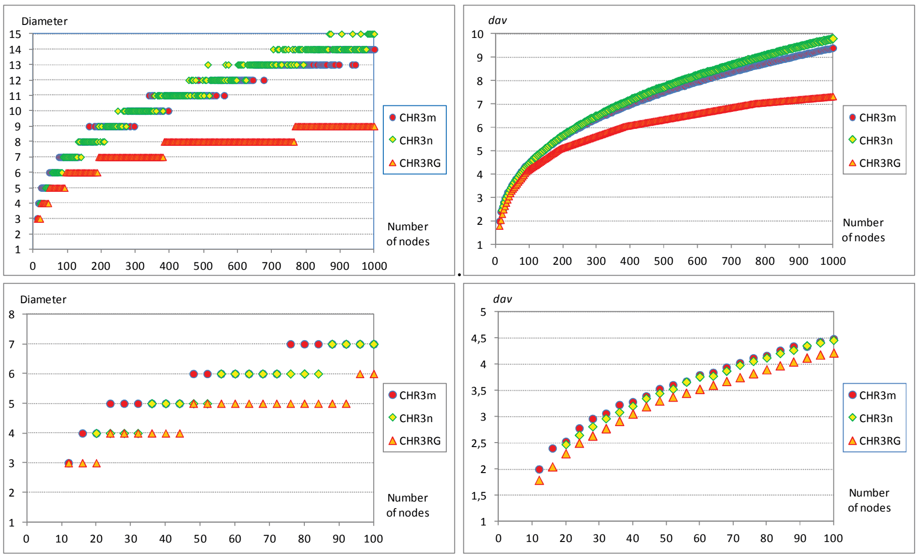

Fig. 4. Comparison of the studied structures with Referential Graphs in the function of the number of nodes

According to the charts, in case of a higher number of nodes, the parameters of modified structures vary considerably from the parameters of a Referential Graph, whereas if the number of nodes does not exceed 100, the differences are insignificant. Moreover, it has been found that there is no ideal graph possessing features of a Referential Graph among the studied structures CHR3m and CHR3n (optimal graphs are assumed not to be existing as according to the carried out studies , the number of nodes of optimal node GR3 is not divisible by 4).

Theoretically, there should exist an optimal Referential Graph CHR3 having 10 nodes as the computed sum of the determined nodes distribution equals 9 for the first two layers, however, if we use the below presented mosaic (fig. 5) it is easy to prove that there is no possibility to obtain a graph with parameters of a referential graph.

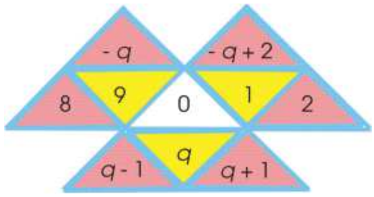

Fig. 5. Scheme of a mosaic

In the mosaic row where there is a zero number node, there occur nodes with numbers 1, 2, 9 and 8 which is determined according to the construction rule for chordal rings.

In order to provide a graph with properties of a Referential Graph, the remaining nodes of the second layer need to have numbers: 3, 4, 5, 6 and 7. The length of chord q cannot assume values 3 and 7 as in the lower layer there would appear nodes with numbers, respectively 2 and 8. If the chord length was equal to 4, 5 or 6 then the nodes with numbers, respectively: 5, 4 and 6 would be doubled in the second layer. Hence, it can be said that there is no CHR3 graph with properties of a optimal Referential Graph third degree.

The only found ideal Referential Graphs belonging to the so far discussed degree three structures are standard chordal rings built of 6, 8 and 14 nodes whose chord lengths are, respectively: 3, 4 and 5 (Fig. 6).

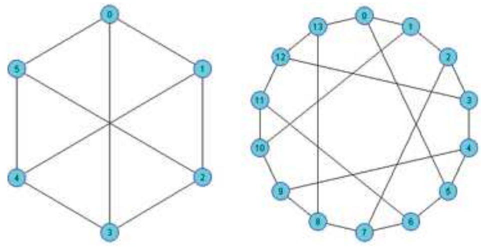

Fig. 6. Referential Graphs CHR3

The second group of the analyzed graphs of third degree, form the structures having a double ring. The definition of this type of graphs is as follows:

Definition 9

Structure NdR (fig. 7), defined as NdR(2p), is called a graph formed by two rings each consisting of p nodes:

- External ring where each node o_k is connected with two adjacent nodes $o_{k-1(\mathrm{mod}\ p)}$ i $o_{k+1(\mathrm{mod}\ p)}$,
- Internal ring where each node i_{k+p} is connected with two adjacent nodes $i_{k+p-1\ (\mathrm{mod}\ 2p)}$ and $i_{k+p+1\ (\mathrm{mod}\ 2p)}$,
- Each node of internal ring i_{k+p} is connected with a corresponding node of the external ring o_k.,
- Each node is a degree three node.

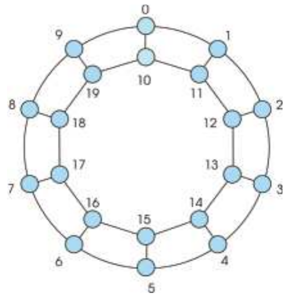

Fig. 7. An example of structure NdR(20)

Structure NdR($2p$) can be considered as 'not flexible' – its parameters are invariable and they depend merely on the number of nodes. Its modification has been proposed to remove this disadvantage. Graphs NdRm, described in papers [22,23,24] having the best properties of all the analyzed graphs, are a good example of these structures modification.

Definition 10

Two rings, each containing p nodes, form the described NdRm($2p;q$) structure:

- External ring where each node o_k is connected with two adjacent nodes $o_{k\text{-}1(\mathrm{mod}\ p)}$ i $o_{k+1(\mathrm{mod}\ p)}$;
- Internal ring where each node $i_{kwo\ adjacent\ nodes\ +p}$ is connected with two adjacent nodes $i_{k+p\text{-}q\ (\mathrm{mod}\ 2p)}$ i $i_{k+p+q\ (\mathrm{mod}\ 2p)}$ through a chord of q length being a multitude of the external ring length;
- Each node of the internal ring i_{k+p} is connected with its corresponding node of external node *ok.*;
- Degree of all the nodes equals 3.

NdRm structures can be divided into two classes:

- The first, in which chords of both rings form the Hamilton cycles. In this case the number of nodes and the chord length of inner need to be relatively prime (Fig. 8A).
- The second, in which the external ring chord forms the Hamilton cycle with property $j = i \oplus 1\ mod\ p$, whereas the internal ring consists of a certain number of disjoint cycles of the same length, smaller than p (Fig. 8B).

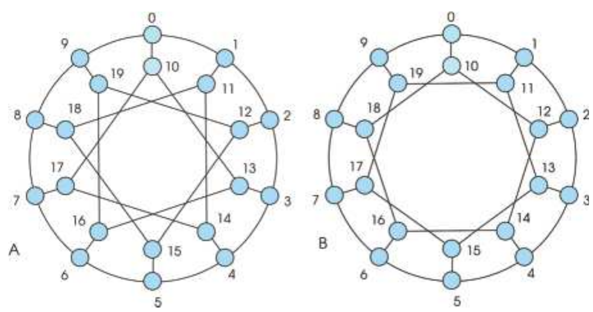

Fig. 8. Examples of two kinds of structures –NdRm(20;3) and NdRm(20;2)

Definition 11

NdRc (fig. 9) is a structure described as NdRc($2p$; q_1,q_2), formed by two rings:

- External ring consisting of p nodes (p must be divisible by 2), in which each node o_k is connected with two adjacent nodes $o_{k\text{-}1(\text{mod } p)}$ i $o_{k+1(\text{mod } p)}$;
- Internal ring also consisting of p nodes. Each even number node i_{2k+p} is connected with two adjacent nodes $i_{2k+p\text{-}q1\ (\text{mod } 2p)}$ and $i_{2k+p+q1\ (\text{mod } 2p)}$, and each odd number node i_{2k+1+p} is connected with two nodes $i_{2k+1+p+q2\ (\text{mod } 2p)}$ and $i_{2k+1+p\text{-}q2\ (\text{mod } 2p)}$;
- *Each* node of internal ring i_{k+p} is connected with its corresponding node o_k of the external ring;
- *Parameters* q_1 and q_2 denoting chord lengths must be of even number;
- All *nodes* are of three degree.

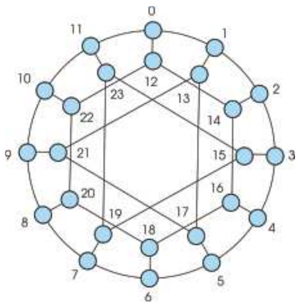

Fig. 9. Example of structure NdRc(24; 2,4)

In figure 10 there is a comparison of diameters and the average path length of the best structures, built on the basis of NdRc graphs, with Referential Graphs. The presented charts prove that parameters NdRc are closer to the parameters of Referential Graphs, though their values are far from being ideal.

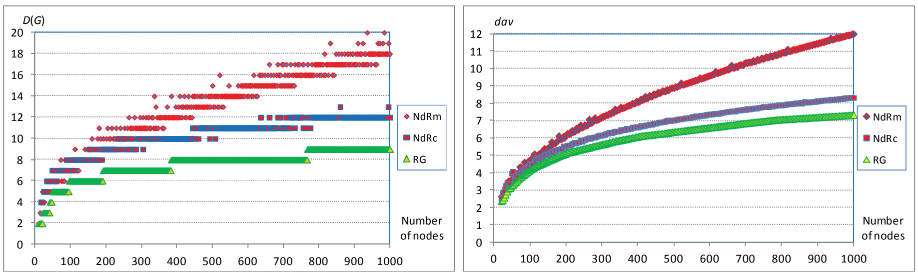

Fig. 10. Comparison of parameters of NdRm, NdRc real structures and Referential Graphs in the function of the node number.

The charts also account for parameters of NdRm structures. It results from the fact that although these parameters significantly differ from values characteristic of Referential Graphs, they form the only optimal graph NdRm (5;2) (Petersen graph) that has been identified, and two ideal graphs NdRm(7;2) and NdRm(13; 5) (Fig. 11).

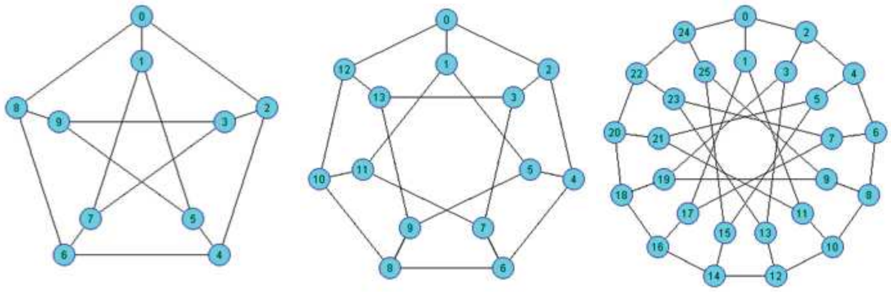

Fig. 11. Structures NdRm forming Referential Graphs

A comparison of parameters of the analyzed structures CHR3, NdR with Referential Graph has been shown in Figure 12 as a summary of this section of the paper.

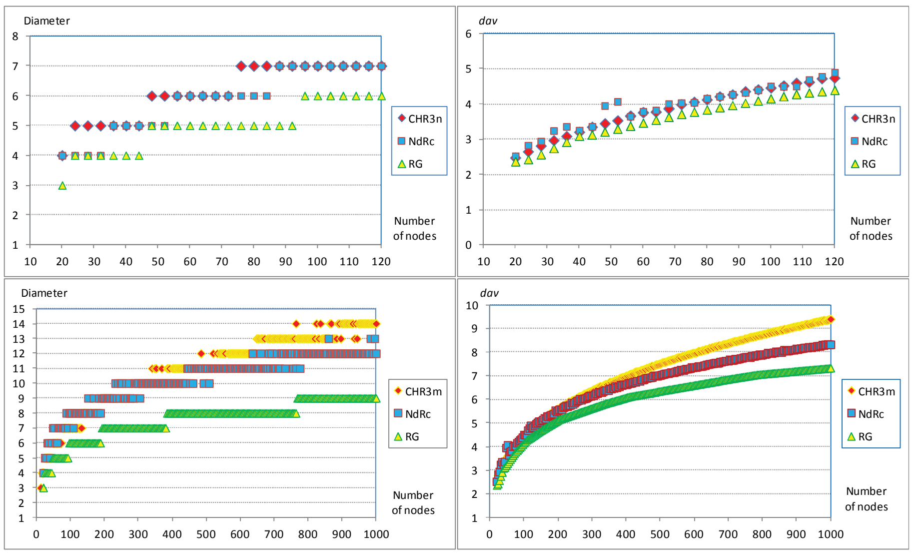

Fig.12. Comparison of real CHR3m, NdRc structures with Referential Graphs in the function of the node number

The charts show that parameters of the NdR type structure are more closer to the parameters of a Referential Graph though if the number of nodes connecting comparable modified structures does not exceed one hundred they do differ insignificantly.

This conclusion confirms that, despite being more expensive, the use of the NdR type structures for building a network increases its reliability.

2.1. Fourth degree referential graphs

In table 3 there is a distribution of nodes occurring in the successive layers of an optimal degree four Referential Graphs.

Table 2. Distribution of the maximal number of nodes occurring in the successive layers of a referential graph

d	1	2	3	4	5	6	7	8
p_{dRF}	4	12	36	108	324	972	2916	8748

The total number of nodes in the function of the graph diameter, computed using the above given number of nodes that occur in the successive layers, has been presented in table 4.

Table 3. The summary number of nodes in the function of the graph diameter, computed using the above given number of nodes that occur in the successive layers of four degree Referential Graph

$D(G)$	1	2	3	4	5	6	7	8
p_{dRF}	5	17	53	161	485	1457	4373	13121

Below there is the definition of a four degree Reference Graph.

Definition 12

Referential Graph is a graph whose parameters are equal to parameters of a minimum spanning tree defined in the following way:

– The number of nodes in the d-th layer of an optimal Referential Graph is defined by the formula:

$$p_{dRG} = 4 \cdot 3^{(d-1)} \tag{9}$$

– The summary number of nodes $p_{D(G)RG}$ in the function of diameter is described by the formula:

$$p_{D(G)RG} = \frac{4 \cdot 3^{D(G)RG} - 2}{2} = 2 \cdot 3^{D(G)RG} - 1 \tag{10}$$

– The graph diameter in the function of the number of nodes can be calculated with the use of the formula:

$$D(G) = \left\lceil \log_3 \frac{p_{D(G)RG} + 1}{2} \right\rceil \tag{11}$$

– The average path length in optimal Referential Graph d_{avRG} in the function of diameter is given by the formula:

$$d_{avRG} = \frac{\left(2D(G) - 1\right) \cdot 3^{D(G)_{RG}} + 1}{2 \cdot \left(3^{D(G)_{RG}} - 1\right)} \tag{12}$$

Like in case of third degree graphs, definitions of chordal rings which have been compared with Referential Graphs will be presented [26].

The definition of a standard chordal ring sounds as follows:

Definition 13

The described by CHR4(*p;Q)* chordal ring fourth degree, is a graph consisting of p nodes. Each node v_i is connected with four nodes: $v_{i-1(mod\ p)}$, $v_{i+1(mod\ p)}$ forming a ring and additionally with two adjacent nodes $v_{i-q(mod\ p)}$ and $v_{i+q(mod\ p)}$, where q denotes the length of the additional link (chord) corresponding to the number of nodes on the ring when the numbers corresponding to values p and q must be relatively prime. Figure 13 shows a graph of this type.

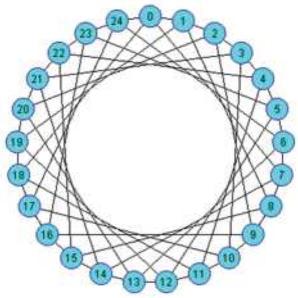

Fig.13. An example of a chordal ring CHR4(25; 7)

This base structure has been modified in order to find structures with better base parameters.

It has been found that of all the analyzed, modified, degree four chordal rings [26] the best parameters are characteristic of graphs called CHR4b and CHR4d which have been defined in the following way:

Definition 14

Chordal ring CHR4b is a structure defined as CHR4b(p; q_1,q_2), consisting of p nodes, whereas p, if it is not a full graph with five nodes, has to be of an even number. Each node v_i is connected with nodes $v_{i-1(\text{mod } p)}$ and $v_{i+1(\text{mod } p)}$. Moreover, each even number node v_{2i} is connected with nodes $v_{2i-q1(\text{mod } p)}$ i $v_{2i+q1(\text{mod } p)}$, and an odd number node $v_{2i+1(\text{mod } p)}$ with nodes $v_{2i-q2(\text{mod } p)}$ i $v_{2i+q2(\text{mod } p)}$. Values of parameters q_1 and q_2 are of even numbers, whereas $p/2$ and $q_1/2$, $q_2/2$ are relatively prime, and denote chord lengths.

An example of the discussed structure is shown in Figure 14.

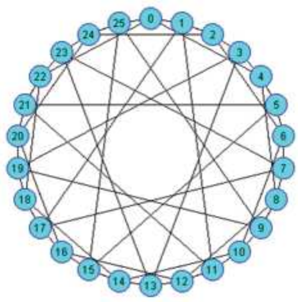

Fig.14. Graph CHR4b(26; 4,10)

Definition 15

Structure CHR4d (example Fig. 15) is a chordal ring defined as CHR4d(p; q_1,q_2,q_3), consisting of p nodes, whereas p must be divisible by 4. Each node v_i is connected with nodes $v_{i-1(\text{mod p})}$ and $v_{i+1(\text{mod p})}$. Additionally, even node v_{2i}, if $2i = 0$ (mod 4), is connected with nodes $v_{2i+q1(\text{mod p})}$ and $v_{2i+q2(\text{mod p})}$, if $2i = 2$ (*mod 4*), it is connected with nodes $v_{2i+q1(\text{mod p})}$ and $v_{2i-q2(\text{mod p})}$, whereas odd node $v_{2i+1(mod\ p)}$ if $2i+1 = 1$ (mod 4), is connected with nodes $v_{2i+1-q1(mod\ p)}$ and $v_{2i+q3(mod\ p)}$, and if $2i+1 = 3$ (mod 4) it is connected with nodes $v_{2i+1-q1(mod\ p)}$ and $v_{2i+1-q3(mod\ p)}$. The value of parameter q_1 is even, whereas parameters q_2 and q_3 are even, whereas they must satisfy condition: $q_2 - q_3 = 0$ (mod 4). Parameters $q_{1,}$ q_2 and q_3 define the chords lengths.

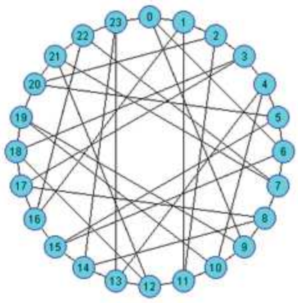

Fig. 15. An exemplary structure CHR4d(24; 9,6,10)

In Figure 16 there are results of the comparison of these structures parameters with the parameters of Referential Graphs.

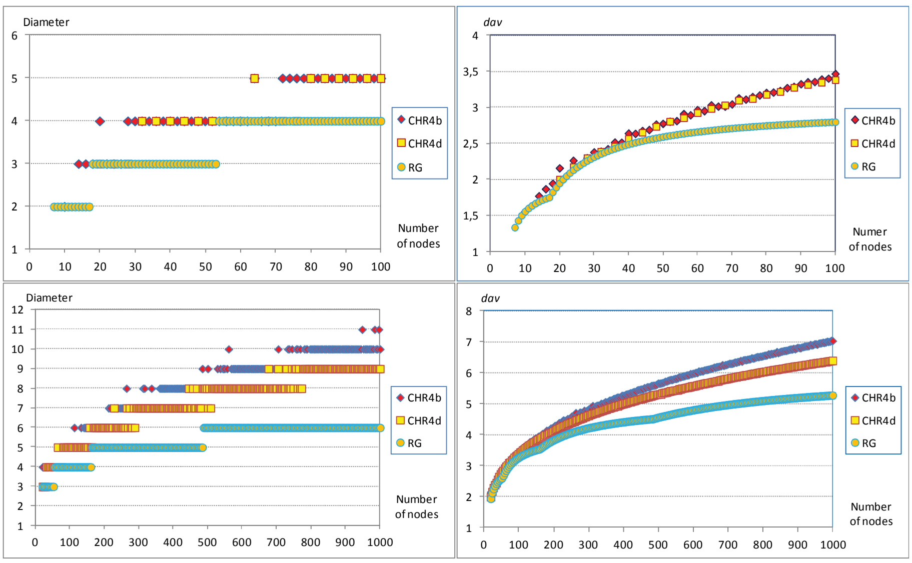

Fig. 16. Comparison of structures CHR4b i CHR4d from degree four Referential Graphs

The presented results prove that the parameters of Referential Graphs, if the number of nodes exceeds 40, differ significantly from the best proposed structures of type CHR4.

However, on the basis of the carried out tests it was found that standard chordal rings CHR4(7; 3), CHR4(8; 3), CHR4(9; 4), CHR4(11; 4), CHR4(13; 5) and graphs CHR4b(6; 2,2) CHR4b(10; 2,4), CHR4b(22; 4,8), CHR4b(26; 4,10) are ideal Referential Graphs.

These structures are shown in figures 17 and 18.

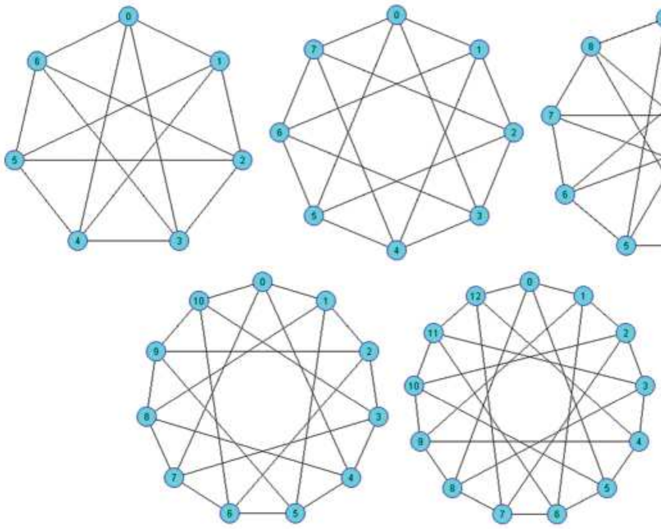

Fig. 17. Chordal rings of type CHR4

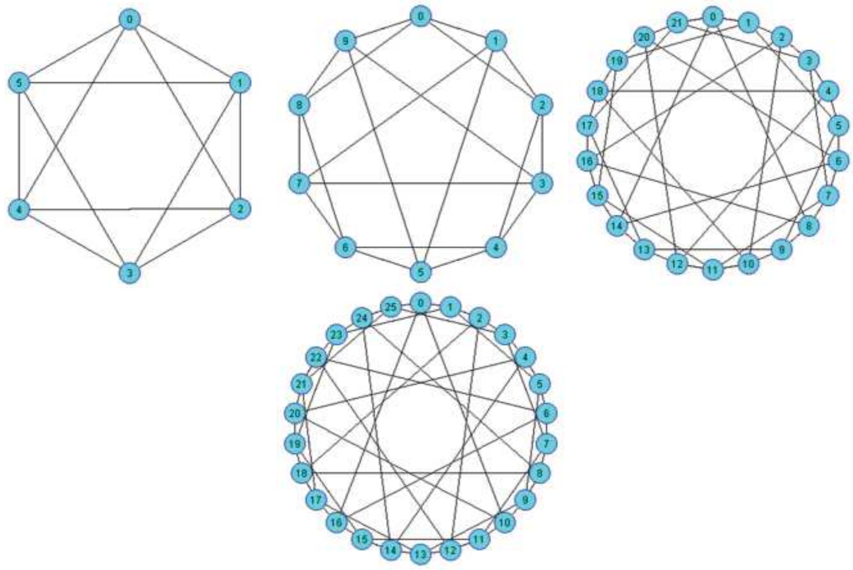

Fig. 18. Chordal Rings of CHR4b type

On the basis of the carried out analysis it can be said that:

Since Referential Graphs have been found to exist in reality it is advisable to continue searching for such structures which use rings as the base of their construction.

3. SEARCH OF REFERENTIAL GRAPHS

The authors of this paper aim at finding other graphs with the characteristics of Referential Graphs (RG) parameters. If on the basis of the conducted research it has been found that RG graphs exist in the real world, the developed program enabled to survey all the structures possible to obtain for the assumed number of nodes forming these graphs.

To begin with, a new mode of operation will be presented, that is the tool to be used for finding the considered structures, will be described.

3.1. The method used for finding referential graphs

A tool which enables to survey all possible ring structures in order to find out whether their basic parameters reach values characteristic of Referential Graphs is the program referred to as GraphFinder4.

Graph Finder4 offers a wide range of possibilities to examine graphs. For this reason, before activating the program, it is necessary to specify features of the examined structures:

- Node Number,
- Node Degree,
- Minimal cycle, function is introduced in order to limit the number of the examined connections,
- Minimal number of parts on which decomposition of the problem will follow enabling performance of computing on numerous processors.

Introduction of the function „Min. Cycle size" is the effect of the following observation. According to the definition of Referential Graph the tree constructed from an arbitrary node being a root of this graph will have an identical form. The condition that a random node of the *k-th* layer, where $k < D(G)$ can be connected with only one node of the *l-th* layer, but there cannot exist an edge connecting it with a node belonging to the *l-th* layer, where $l < k - 1$ must be satisfied. However, this node can possess a connection with a node belonging to the same layer or a layer with number $k + 1$. Thus, the minimal length of the cycle (measured by the number of nodes) from a node considered to be the root, which can appear in a degree three graph is:

$$l_{\min} = 2D(G) - 1 = 2\log_{d(V)-1} \frac{p_{GR}\left(d(V)-2\right)+2}{d(V)} - 1 \tag{13}$$

Hence, it can be concluded that the number of examined structures can be limited thanks to elimination of shorter length cycles, which reduces the examination time.

To illustrate the study, an example of a degree three referential graph has been presented in Figure 19. Node 0 was assumed to be a root of the tree. The l minimal cycle length is 5 and the length of the exemplary one covers nodes: 0, 6, 7, 8, 17, and all the remaining cycles formed with the use of chords, are longer (e.g. 0, 1, 10, 9, 8, 17) or equal to the shortest one (e.g. 1, 10, 9, 3,2).

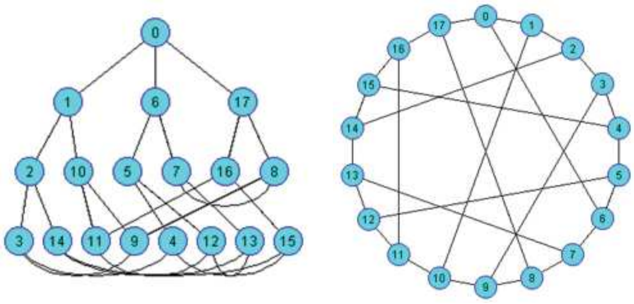

Fig. 19. An example of degree three Referential Graph

The operation principle of the program is given below. The described applications have been implemented in Java language, however, for better understanding, the program codes are presented in Phyton language.

- *Method of a graph denotation*

The program saves a graph in the form of a set of lists.
Initiation of a graph with no links looks as follows:

graph = [[-1 for x in range(degree)] for x in range(nodeNumber)]

where „degree" denotes degree of nodes , a „nodeNumber" – number of the graph nodes.

This notation can be interpreted, as a two dimensional matrix in which rows represent the successive nodes and columns connections with other nodes. Index nodes are from 0 to nodeNumber -1. Value "-1" means lack of connection.

- *Algorithm for determination of diameter and mean path length.*

The program determines a diameter and the average path length by means of a simplified Dijkstra algorithm. Such a simplification is possible thanks to acceptance of the assumption that the weight of the graph all edges are equal to 1.

Figure 20 shows the algorithm to be used for determination of the graph diameter and the average path length.

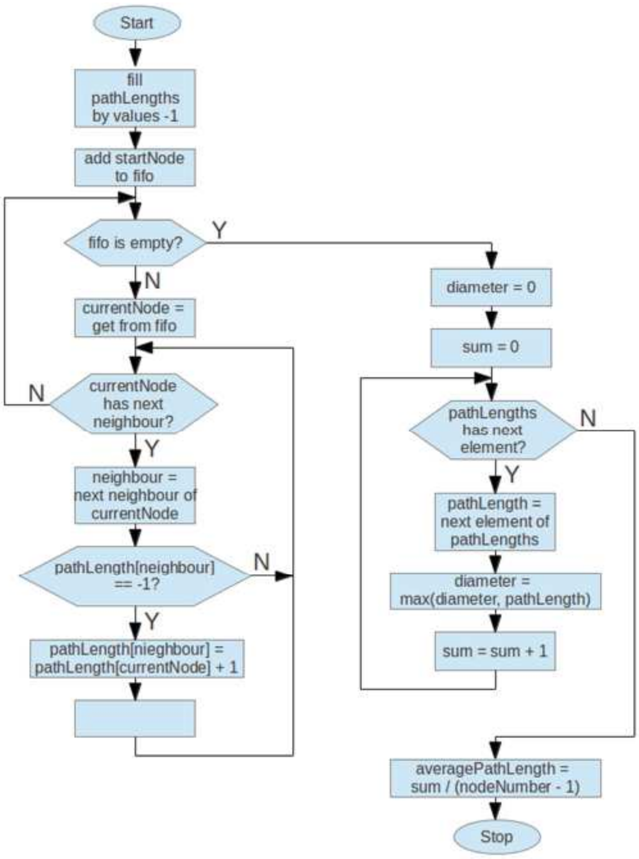

Fig. 20. Algorithm the diameter and average path length determination

Implementation of the function using the algorithm is of the following form.

```
# calculate diameter and average path length for given source node
def calculate(graph, sourceNode):
 # degree of node
 degree = len(graph[0])
 # create empty list of path lengths
 pathLengths = [-1 for x in range(degree)]
 pathLengths[sourceNode] = 0
 # create a queue of node to handle
 fifo = deque()
 fifo.append(sourceNode)
 # loop executing until the queue is empty
 while len(fifo) > 0:
 # getting and removing from queue a node to servicing
 currentNode = fifo.popleft()
 # service node neighbours
 for neighbour in graph[currentNode]:
 # checking if neighbour has been previously serviced
 if pathLengths[neighbour] == -1:
 # calculate the path length to the neighbour
```

```
pathLengths[neighbour] = pathLengths[currentNode] + 1
# adding a neighbour to the queue
fifo.append(neighbour)
# radius
diameter = 0
# sum of path lengths
pathLenhtgSum = 0.0
for pathLength in pathLengths:
# searching of the longest path
diameter = max(diameter, pathLength)
# adding to the sum
pathLenhtgSum += pathLength
# calculate of average path length
averagePathLength = pathLenhtgSum / (len(graph) - 1)
# return results
return (diameter, averagePathLength).
```

This function computes the diameter and average path length from the point of view of a given source node, thus to determine these parameters values for a graph it is necessary to provide each graph with this function. According to the definition, the graph diameter will be the longest of the shortest paths obtained with the use of the function, whereas the average path length is the mean arithmetic from all the results returned by this function.

- *Method of finding referential graphs*

The purpose of this program is to search for referential graphs. This process can be divided into two stages. In the first stag, all possible structures of a minimum spanning tree are determined for a given number of nodes of the graph chosen degree, and in the other stage – each tree is developed.

Step I

First, all possible isomorphic simple trees are constructed from number 0 node, accepted to be their root. These trees are created only up to the last but one layer, (when optimal graphs are searched for – up to the last layer), and the other layers which belong to the last layer, remain unconnected.

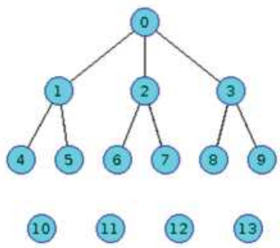

Fig. 21. A sample minimum spanning tree of the search for an ideal Referential Graph consisting of 14 nodes

In the next stage, all possible combinations of nodes of the last layer with the nodes of the last but one layer are created. Repetitions are omitted, that is, such connection topologies which after transformation would give the same effect.

In turn, a division of connections with the nodes belonging to the last layer is performed in order to build trees with 0 node root.

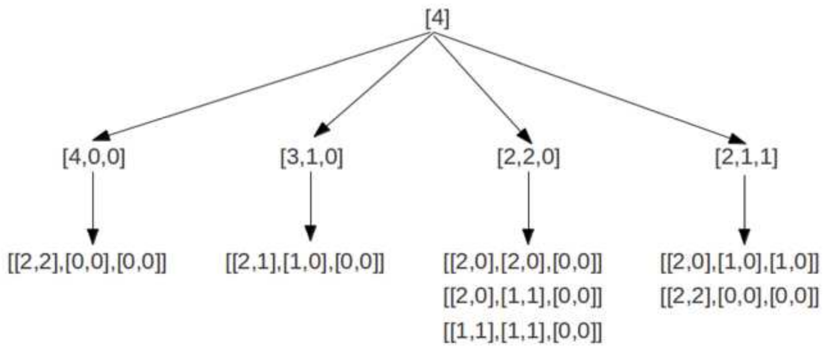

Fig. 22. Possible divisions of connections for the analyzed example

In the analyzed example four nodes are available. In Figure 22 there are presented possibilities of the second layer nodes to perform possible connections eliminating repetitions of identical connections and in Figure 23 there are trees developed for the above example which have the following form:

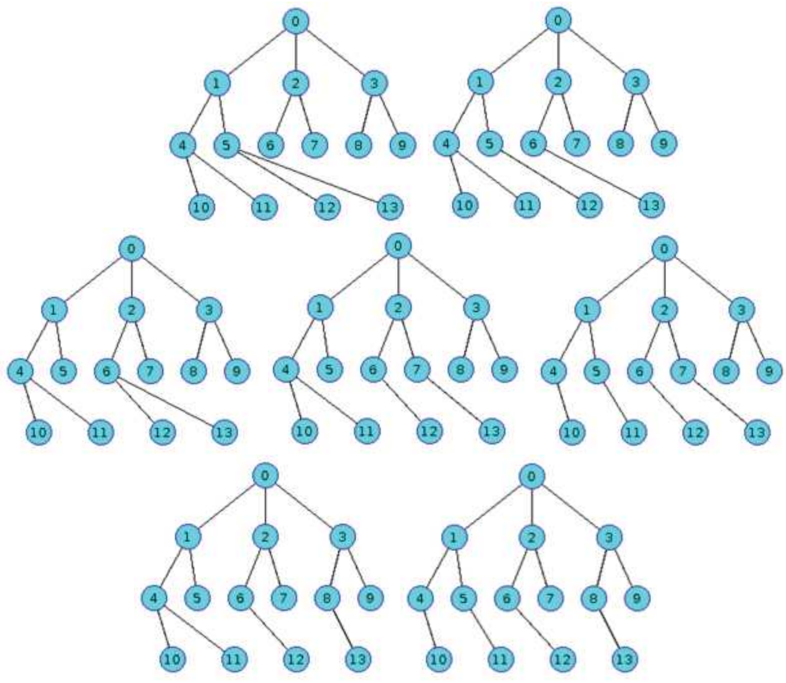

Fig. 23. Trees provided for the studied example

The obtained trees provide the basis for the next stage of Referential Graphs seeking.

Step II

The minimum spanning trees provided in the first step of Stage 1 are starting point of the algorithm. Additional connections from successively chosen nodes appear, every time the condition whether the graph radius and diameter are equal to parameters of the primary tree, is checked. I f the graph construction is finished (a set of connections is completed) and if it is a Referential Graph, the algorithm stops acting. If not – a new set of subgraphs, with one more connection, is created for the graph. This set is added to the list of graphs to be examined. The algorithm finishes its action after having examined all graphs from the list.

General algorithm for finding referential graphs looks as follows (Fig. 24):

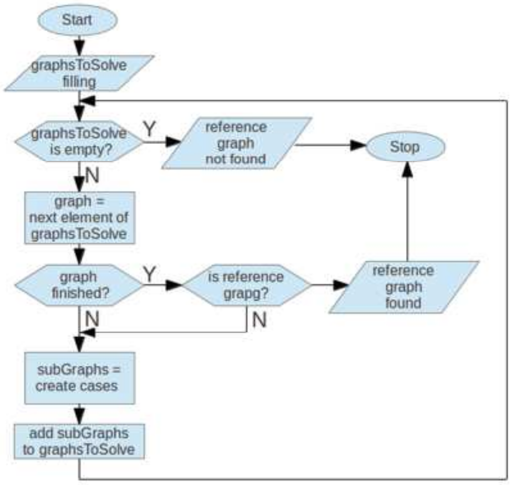

Fig. 24. General algorithm for finding Referential Graphs

Implementation of the function using recurrence for examination of graphs has the following form:

```
# decomposition function
def decompose(graphsToSolve):
 # loop on graphs to solve
 for graph in graphsToSolve:
 # checking if graph construction is finished, all nodes connected
 if isFinished(graph):
 # checking if graph is reference graph
 if isReference(graph):
 return graph
 # graph is not finished
 else:
 # sub-graphs creating
 subGraphs = createSubGraphs(graph)
 # recursive call
 result = decompose(subGraphs)
 if result != None:
 return result
 # reference graph not found
 return None.
```

This function provides the possibility to decompose a problem in order to perform calculations in a concurrent way (e.g. on many cores of the processor or in a cloud computing).

In order to study all of the possible connections of a given graph, two lists of candidate nodes are made. The first one is called a list of source nodes and the second is called a list of target nodes. Next, graphs are formed for a combination of the first source node with all the target ones. The first successive node on the list of source nodes is the node nearest the start node which has no connection with z $d(V)$ neighbors. Target nodes are nodes from the last but one or the last layer from the start node.

The start node is determined by a separate algorithm. The distance between the source node and the target one cannot be shorter than the length of the shortest cycle that can occur in this graph, reduced by 1.

The algorithm providing the connections variants are shown in Figure 25.

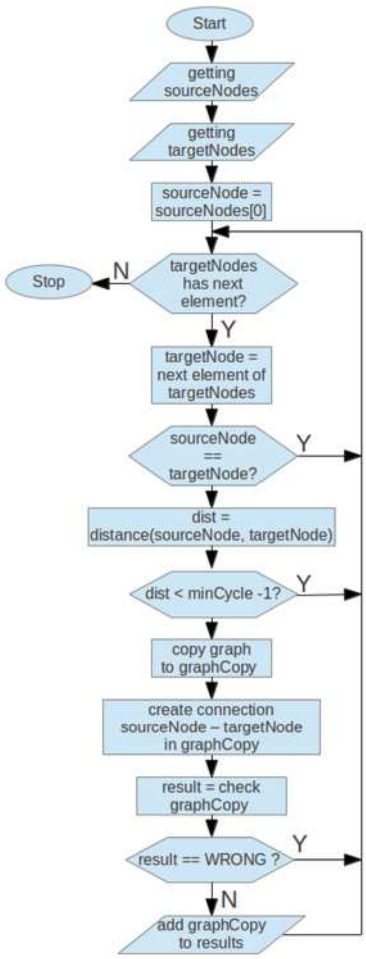

Fig. 25. Algorithm determining variants of connections

Implementation of the function searching for variants looks as follows:

```
# sub-graphs creating function
def createSubGraphs(graph):
 # marking the start node
 startNode = getStartNode(graph)
 # getting source and target nodes lists
 sourceNodes, targetNodes = findCandidates(graph, startNode)
 results = []
 # checking if any source node exists
 if len(sourcenodes) == 0:
 return result
 # getting first source node
 sourceNode = sourceNodes[0]
 # loop on target nodes
 for targetNode in targetNodes:
 # checking if source node is target node
 if sourceNode == targetNode:
 continue
 # checking if distance is lower then minimum cycle length
 if distance(sourceNode, targetNode) < minCycle - 1:
 continue
 # copy of the graph
 graphCopy = copy(graph)
 # adding connections to copy
 setConnection(graphCopy, sourceNode, targetNode)
 # examination of the graph
 if checkGraph(graphCopy) != WRONG:
 # saving result
 results.append(graphCopy)
 return results.
```

The algorithm for determination of candidate nodes is shown in Figure 26.

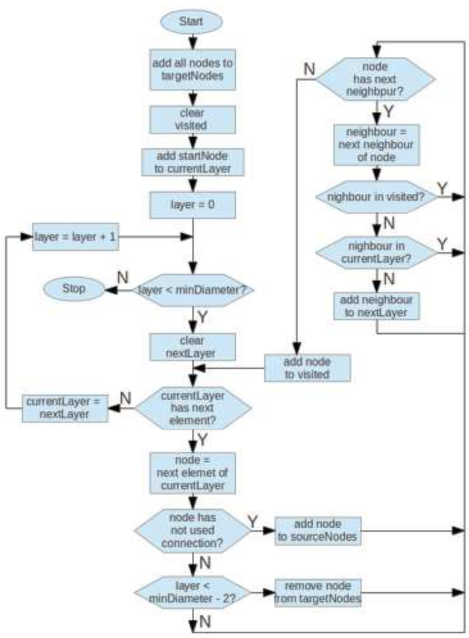

Fig. 26. The algorithm determining candidate source and target nodes

The function using shown above algorithm has following appearance:

```
# source and target nodes designate function
def findCandidates(graph, startNode):
 sourceNodes = []
 # setting a set of target nodes, which will be removed
 targetNodes = range(len(graph))
 # visited nodes
 visited = []
 # nodes in given layer
 currentLayer = [startNode]
 # loop on layers
 for layer in range(minDiameter):
 # nodes in next layer
 nextLayer = []
 # loop on nodes in layer
 for node in currentLayer:
 # checking if node has free connections
 if -1 in graph[node]:
 # adding to source nodes
 sourceNodes.append(node)
 # checking if this is last or penultimate layer
 if layer < minDiameter - 2:
 # removing the target nodes of the previous layer
 targetLayer.remove(node)
 # loop on neighbours to prepare next layer
 for neighbour in graph[node]:
 if neighbour != -1:
 # checking if neighbour has been visited
 if neighbour not in visited:
 # checking if node is not in the layer
 if neighbour not in currentLayer:
 # adding node to next layer
 nextLayer.append(neighbour)
 # marking node as visited
 visited.append(node)
 # changing layer
 currentLayer = nextLayer
 # result rerurn
 return (sourceNodes, targetNodes.
```

The next element of the method for finding Referential Graphs is an algorithm classifying graphs. It examines the graph from the point of view of each node and provides one of the following answers:

– PERFECT – means that from the point of view of this node the distribution is the best of all possible,
– WRONG – means that the algorithm detected a situation excluding the graph as a referential one,
– UNKNOWN – lack of answer, graph can be further examined.

If after having examined all the nodes, the answer WRONG appears even once, there is no point in continuing the examinations. If the answer PERFCT is obtained for all the nodes it means that it is a Referential Graph.

The algorithm examining the graph for a single node looks as follows (Fig. 27):

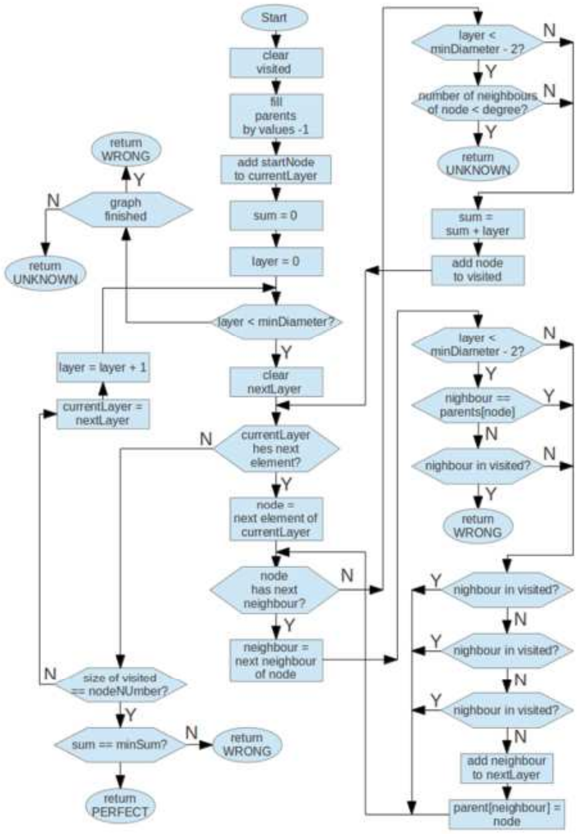

Fig. 27. Algorithm for graph examining from a single node

Implementation of this algorithm is as follows:

```
# constant definition
PERFECT = 0
WRONG = 1
UNKNOWN = 2
# graph examination function
def checkGraphFromNode(graph, startNode):
 visited = []
 # parents list
 parents = [[-1 for x in range(nodeNumber)]
 # current layer
 currentLayer = [startNode]
 # sum of path lengths
 pathLengthSum = 0
 # loop on layers
 for layer in range(minDiameter):
 # next layer nodes
 nextLayer = []
 # loop on nodes in layers
 for node in currentLayer:
```

```
# loop on neighbours
for neighbour in graph[node]:
if neighbour != -1:
# checking if this is last or penultimate layer
if layer < minDiameter - 2:
# checking if neighbour is not parent
if neighbour != parents[node]:
# checking if neighbour has been visited
if neighbour in visited:
# this combination is forbidden
return WRONG
# checking if neighbour hasn't been visited
if neighbour not in visited:
# checking if neighbour is not in current layer
if neighbour not in currentLayer:
# checking if neighbour is not in next layer
if neighbour not in nextLayer:
# adding node to the next layer
nextLayer.append(neighbour)
# marking node as parent
parent[neighbour] = node
# checking if this is last or penultimate layer
if layer < minDiameter - 2:
# checkinh if node has all connections
if getNeighbourNumber(graph, node) < len(graph[0]):
# answer cannot be determined
return UNKNOWN
# adding to path length sum
pathLengthSum += layer
# marking node as visited
visited.append(node)
# checking if all nodes were visited
if len(visited) == len(graph):
# checking if path length sum is minimal possible sum
if pathLengthSum == minPathLengthSum:
return PERFECT
else:
return WRONG
# changing layers
currentLayer = nextLayer
# checking id graph is finished
if isFinished(graph):
return WRONG
return UNKNOWN.
```

The above given operation principle of the program makes it possible to achieve the goal set by the authors of this paper, that is to find Referential Graphs.

In order to illustrate the study we will show an example depicting operation of this application.

It was assumed that the minimum spanning tree obtained from stage 1 has the form demonstrated in Figure 28.

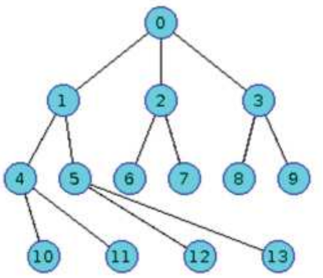

Fig. 28. Minimum spanning tree of a referential graph

The same tree seen from the point of view of exemplary nodes assumed to be start nodes is shown in Figure 29.

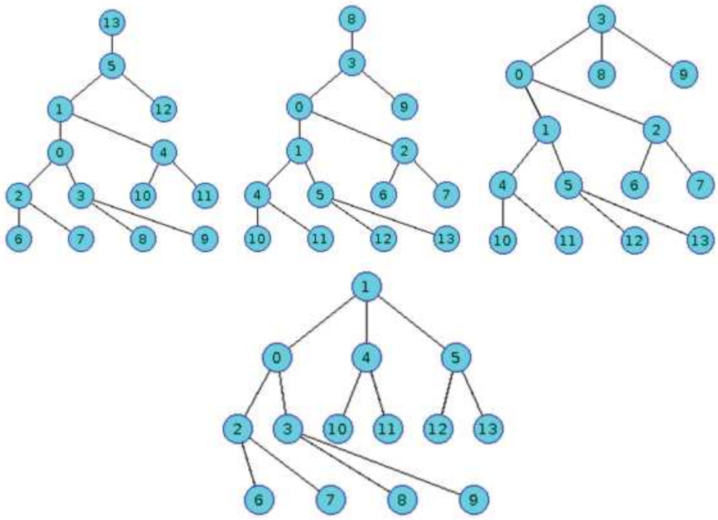

Fig. 29. Forms of the tree seen from exemplary source nodes

According to Figure 29, at the same stage of the analysis, in each case, graphs will produce the answer PERFECT.

After adding the successive four, out of eight, connections necessary to obtain the whole graph the structure has the following form (from the point of view of the chosen nodes) (Fig. 30).

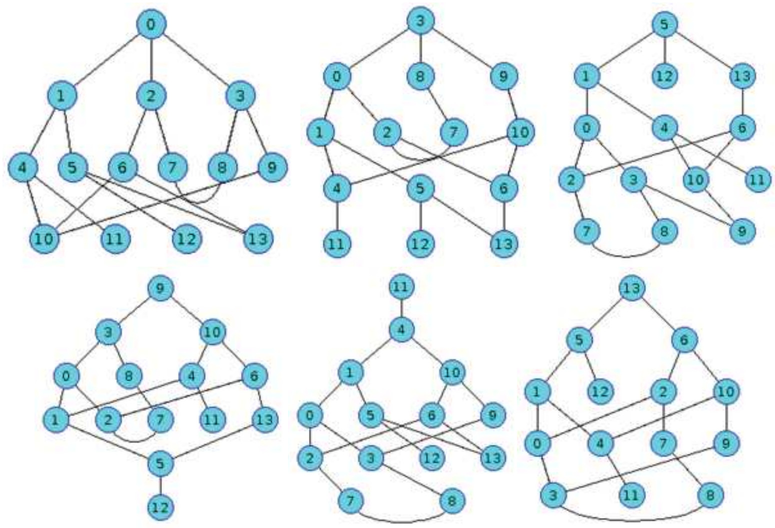

Fig. 30. Stage of searching for Referential Graph connections

After introduction of connections between nodes 12 & 8 and 9 & 11, which is definitely the most convenient stage to do it, obtained graph looks as follows (Fig. 31):

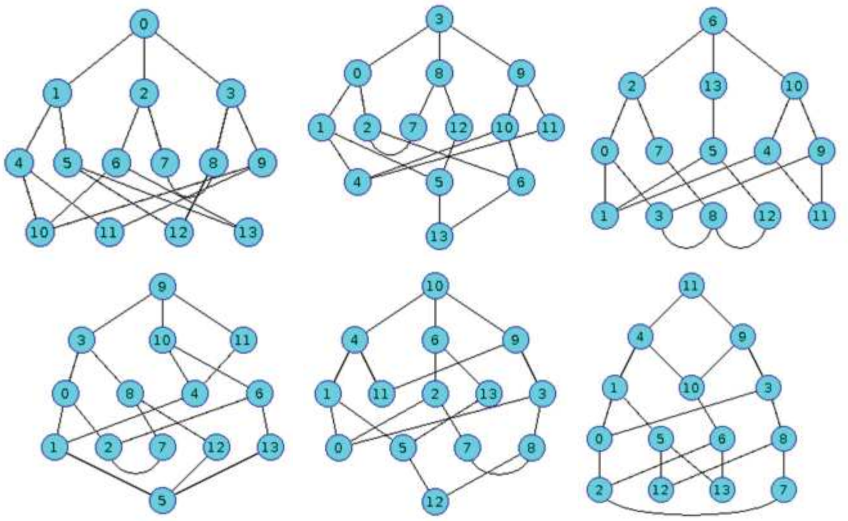

Fig. 31. Successive stage of seeking Referential Graph connections

Examination of the so obtained graph will result in the answer WRONG, that is, at this point, the process of Referential Graph seeking will be abandoned, as from the point of view of graphs no. 9, 10 and 11, there are two connections between the nodes of the first layer, and one node of the second layer, so there appeared a cycle connecting nodes 4, 9, 10 and 11.

However, the Referential Graph found thanks to the use of the discussed application is as follows (Fig. 32):

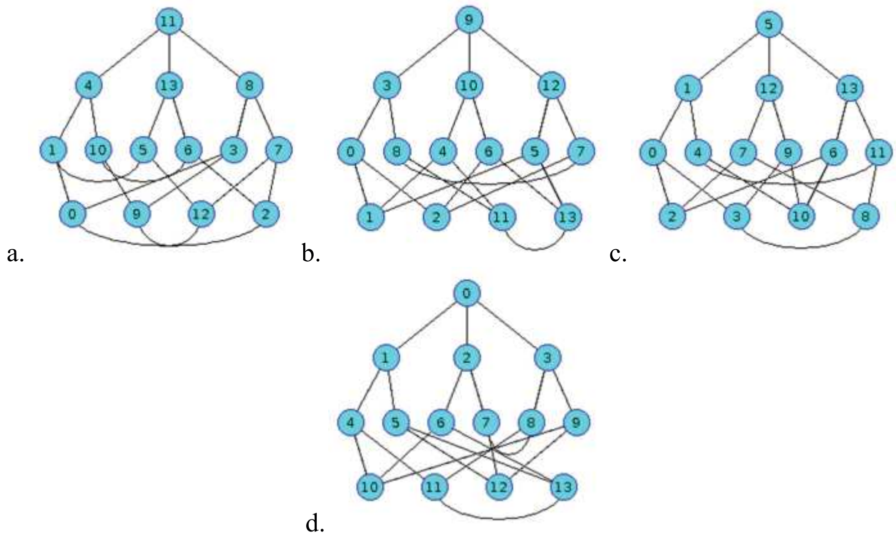

Fig. 32. Referential graph found thanks to GraphFinder4

Additional task to be performed by the discussed application is transformation of graphs shown in Figure 32 into the form of a chordal ring which is described below:

```
# function to finding Hamilton cycles in graph
def findHamilton(graph, path = [0]):
 # getting current node
 currentNode = path[-1]
 # checking that found path through all nodes
 if len(path) == len(graph):
 # checking if current node has connection to first node
 if path[0] in graph[currentNode]:
 # return found cycle
 return path
 return None
 # loop on neighbours
 for neighbour in graph[currentNode]:
 # checking if neighbour has been visited
 if neighbour not in path:
 # adding next node to path
 path.append(neighbour)
 # recursive call
 result = findHamilton(graph, path)
 # checking if cycle has been found
 if result is not None:
 return result
 # removing last element
 path.remove(neighbour)
 # cycle not found
 return None.
```

After the Hamilton cycle was found and converted, the graph shown in fig. 31d assumed the following form (Fig. 33).

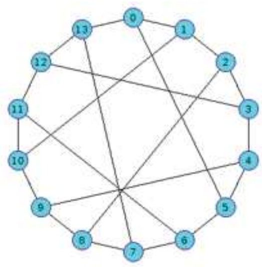

Fig. 33. An example of Referential Graph built with 14 nodes

Thanks to using the program a user can obtain the following information:
- Graph metrics, its basic features;
- Description of graph formation process;
- Topology of the obtained structure connections;
- Graph basic parameters;
- Length the above mentioned minimal cycle;
- Confirmation of appearance of the cycle covering all the graph nodes;
- Time of graph seeking.

After the searching process is finished, on the screen, apart from a description, there is an image of the obtained Referential Graph in two forms shown in Figure 34.

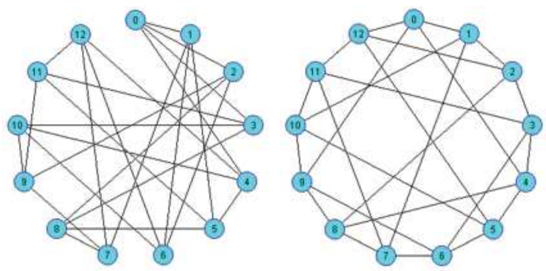

Fig. 34. Images of an exemplary Referential Graph

An additional option of this program is the possibility to turn the graphs which makes it easier to check whether the graphs using different chord lengths are isomorphic graphs.

While using the discussed program degree three and four graphs were surveyed and the effects of Referential Graphs seeking are demonstrated in figures 35 and 36.

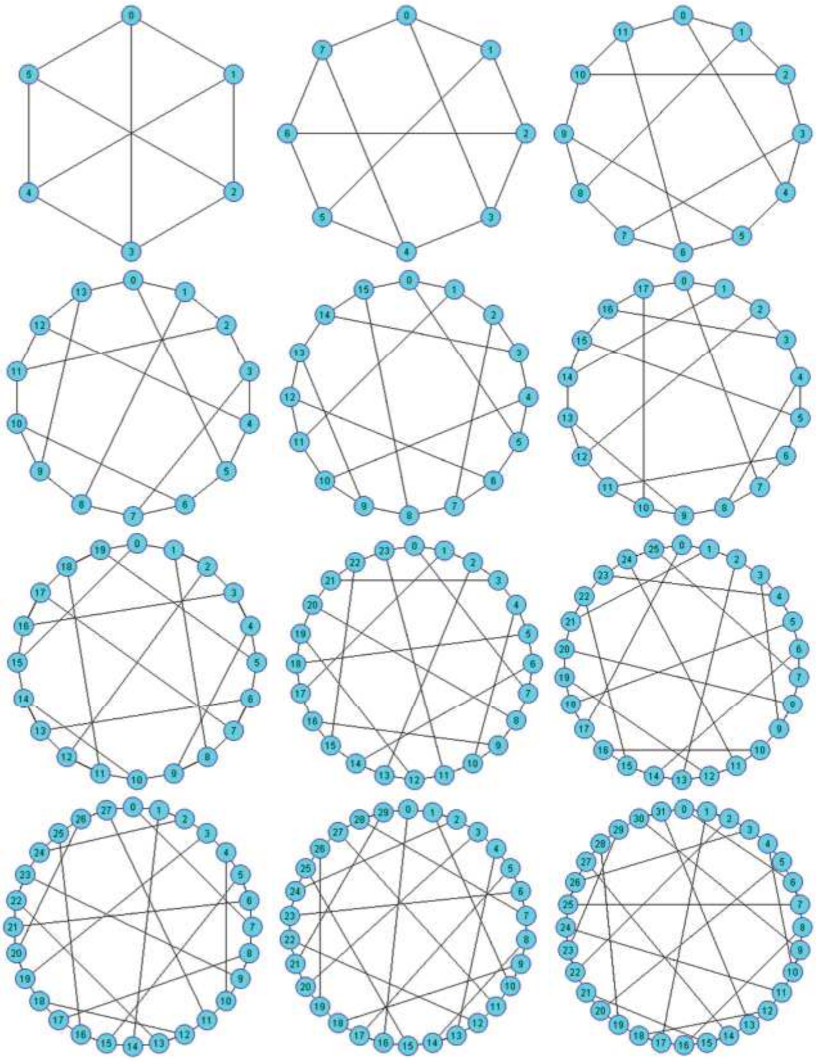

Fig. 35. Examples of degree three and four Referential Graphs

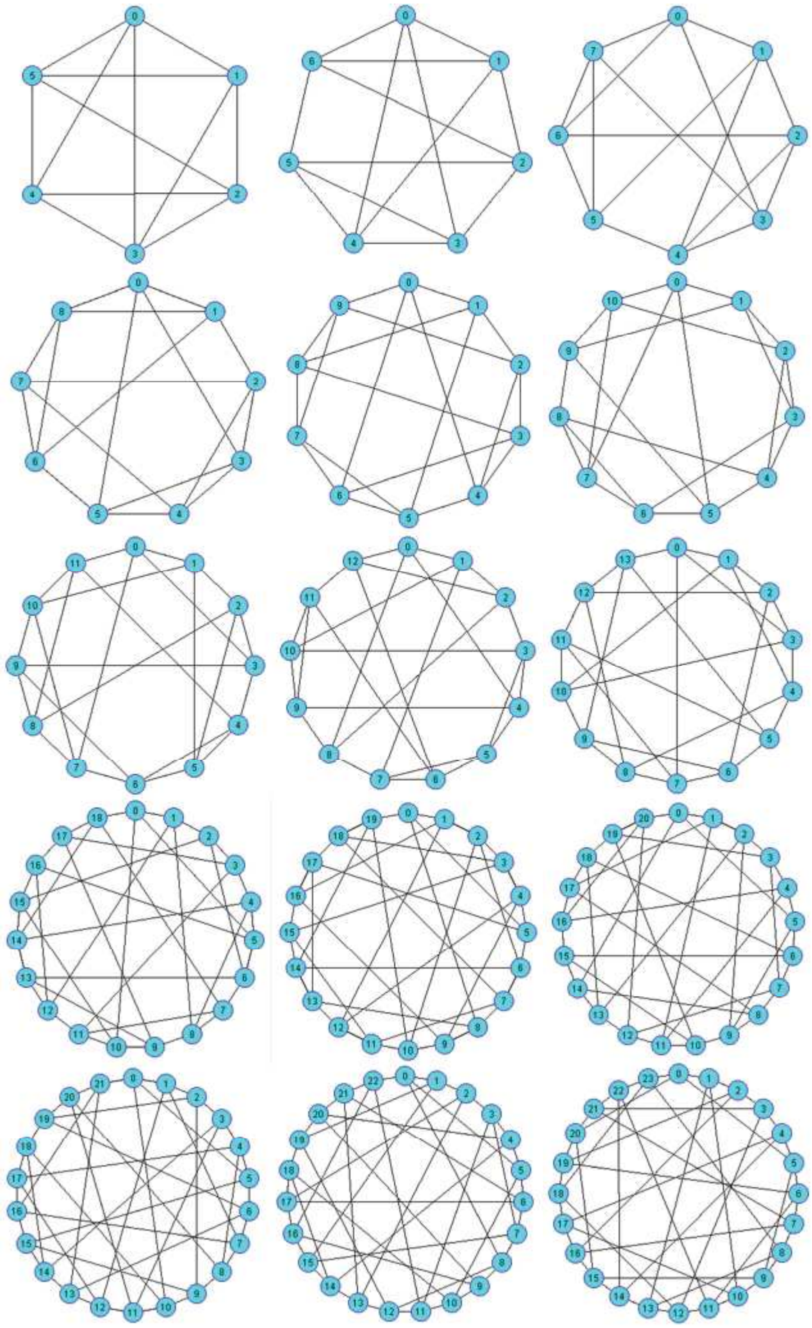

Fig. 36. Examples of found degree four Referential Graphs

4. CONCLUSIONS

In this paper, the notion of Referential Graphs has been defined in order to use it for optimization of a network connecting unified modules that can be used as telecommunication or specialized IT units. Using the studied chordal rings based networks for modeling makes it possible to reach a satisfying level of reliability, capacity, and the minimum information transmission delay thanks to minimization of the basic parameters – diameter and average path length. Parameters of Referential Graphs have been compared with the parameters of modified degree three and four structures proposed by authors. The article includes examples of Referential Graphs found thanks to using the program developed by the authors.

On the basis of obtained results they concluded that not for each number of nodes exists Referential Graphs. In case of degree three graphs, the only one optimal graph is NdR structure having 10 nodes (Pedersen graph), whereas there is no possibility to create a graph based on a single ring. There have been found no optimal chordal rings consisting of 22 nodes, which is of significance as it can indicate that it is not possible to build optimal Referential Graphs with more nodes than 10. Additionally, it would be advisable to check if it is possible to find an optimal graph for topologies with 46 nodes.

In case of degree four graphs, did not find any optimal graph, apart from complete graph constructed of 5 nodes. Despite long lasting tests, there have been found no structure with 16, 17 and 18 nodes.

Due to the time required to perform calculations, examination of degree three graphs could be carried out for structures consisting of 32 nodes, whereas for degree four graph – to 24. Thus, the main problem to be solved at the present stage is a successive modification of the program or use of higher performance computers (Cloud Computing) which should result in shortening the time of searching for nodes with higher numbers of nodes, thus enabling optimization of connections for more developed network.

Such an application provides the possibility of operation in a network and this will allow to use significantly larger computing potential for seeking graphs. The proposed architecture of this application is shown in fig. 37.

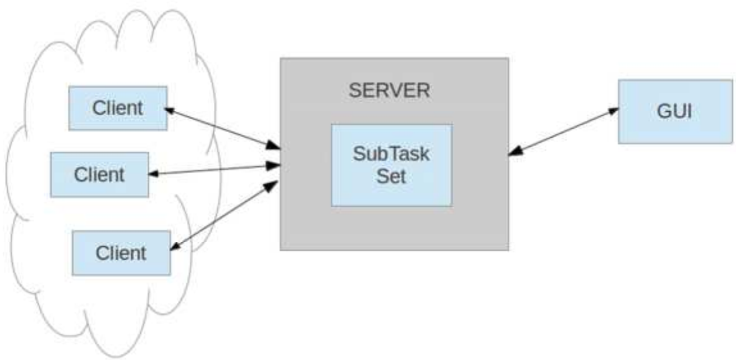

Fig. 37. A scheme of the proposed application: Server – application running on a machine available for the system remaining elements; SubTaskSet – a set of subsets which are assigned to clients. Client-application operating on a machine designed for computing; GUI – application of the user's interface

The server observes calls coming from clients and GUI, it accepts tasks from GUI, decomposes them, and assigns the tasks to particular clients and saves the results. The content of SubTaskSet is obtained from decomposition of the task which is performed by Server (e.g. search of Referential Graph with defined degree and number of nodes). A client receives tasks from the server and returns the obtained results, using all available cores of the processor. GUI enables reception and assignment of new tasks as well as collection and survey of the results gathered by the server. Thanks to implementation of such a solution it is possible to find Referential Graphs with simultaneous use, in our case, more than 80 cores. This is expected to enable disperse many doubts, and solve problems that occur while examining chordal rings which, though being important for modeling of real-world data communications network, have not been accounted for in this paper.

BIBLIOGRAPHY

[1] Bhuyan L.N., 1987. Interconnection Networks for Parallel and Distributed Processing. IEEE Computer 20(6), 9-12.

[2] Bermond J.C., Comellas F., Hsu D., 1995. Distributed loop computer networks: A survey. Journal of Parallel and Distributed Computing 24, 2-10.

[3] Blazewicz J., Ecker J.K., Plateau B., Trystram D., 2000. Handbook on parallel and distributed processing. Springer Berlin. Heidelberg.

[4] Kotsis G., 1992. Interconnection Topologies and Routing for Parallel Processing Systems. ACPC, Technical Report Series, ACPC/TR92-19.

[5] Mans B., 1999. On the Interval Routing of Chordal Rings. ISPAN '99 IEEE – International Symposium on Parallel Architectures, Algorithms and Networks, Fremantle, Australia, 16-21.

[6] Kiedrowski P., 2009. Easy Applicable Algorithm for Accelelerate Reading Process in AMR Systems based on WSN Solutions, Image Processing & Communications Challenges (ed. R.S. Choraś, A. Zabłudowski), Academy Publishing House EXIT, 482-487.

[7] Kiedrowski P., 2012. Media Independent Protocol Suite for Energy Management Systems Based on Short Range Devices. Prace Instytutu Elektrotechniki LIX, z. 258, 189-199.

[8] Kiedrowski P., Boryna B., Marciniak T., 2013. Last-mile smart grid communications based on hybrid technology as a reliable method of data acquisition and distribution, Rynek Energii 1(104), 127-132.

[9] Narayanan L., Opatrny J., Sotteau D., 2001. All-to-All Optical Routing in Chordal Rings of Degree 4. Algorithmica 31, 155-178.

[10] West D.B., 2000. Introduction to Graph Theory, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall.

[11] Arden W., Lee H., 1981. Analysis of Chordal Ring Network. IEEE Transactions on Computers 30(4), 291-295.

[12] Gavoille C. (n.d). A Survey on Internal Routing. http://deptinfo.labri.ubordeaux.fr/~gavoille/article/ survey /node28.html

[13] Bujnowski S., 2003. Analysis & Synthesis Homogeneous Structure Networks Connecting Communications Modules. PhD Thesis, UTP Bydgoszcz.

[14] Pedersen J. M., Gutierrez J.M., Marciniak T., Dubalski, B. Zabłudowski A., 2009. Describing N2R Properties Using Ideal Graphs. Advances in Mesh Networks, MESH 2009. The Second International Conference on Advances in Mesh Networks. ISBN: 978-0-7695-3667-5, 150-154.

[15] Azura R.N.F., Othman M., Selamat M.H., Hock P.Y., 2008. Modified Degree Six Chordal Rings Network Topology. Proceedings of Simposium Kebangsaan Sains Matematik Ke-16, 3-5, 515-522.

[16] Azura R.N.F., Othman M., Selamat M.H., Hock P.Y., 2010. On properties of modified degree six chordal rings networks. Malaysian Journal of Mathematical Sciences 4(2), 147-157.

[17] Farah R.N., Othman M., Selemat H., Hock P.Y., 2008. Analysis of Modified Degree Six Chordal Rings and Traditional Chordal Rings Degree Six Interconnection Network. International Conference of Electronics Design. Penang, Malaysia.

[18] Farah R.N., Othman M.., 2010. In Modified Chordal Rings Degree Six Geometrical Representation Properties. Proceedings of Fundamental Science Congress, Kuala Lumpur, Malaysia.

[19] Farah R.N., Othman M., Selamat M.H., 2010. Combinatorial properties of modified chordal rings degree four networks. Journal of Computer Science 6(3), 279--284.

[20] Farah R. N., Othman M., Selamat M. H., 2010. An optimum free-table routing algorithms of modified and traditional chordal rings networks of degree four. Journal of Material Science and Engineering 4(10), 78-89.

[21] Bujnowski S., Dubalski B., Pedersen J.M., Zabłudowski A., 2009. Struktury topologiczne CR3m oraz NdRm. Przegląd Telekomunikacyjny LXXXI, 8/9, 1133--1141.

[22] Pedersen J.M., Patel A., Knudsen T.P., Madsen O.B., 2004. Generalized Double Ring Network Structures. Proc. of SCI 2004. The 8th World Multi-Conference On Systemics. Cybernetics and Informatics 8. Orlando, USA, 47-51.

[23] Pedersen J.M., Knudsen T.P., Madsen O.B., 2004. Comparing and Selecting Generalized Double Ring Network Structures. Proc. of IASTED CCN 2004. The Second IASTED International Conference Communication and Computer Networks. Cambridge, USA, 375-380.

[24] Zabłudowski Ł., Dubalski B., Kiedrowski P., Ledziński D., Marciniak T., 2012. Modified NDR structures. Image Processing and Communications 17(3), ISSN 1425-140X, 29-47.

[25] Pedersen J. M., Riaz M.T., Madsen O.B., 2005. Distances in Generalized Double Rings and Degree Three Chordal Rings. Proc. of IASTED PDCN 2005. IASTED International Conference on Parallel and Distributed Computing and Networks. Innsbruck, Austria, 153-158.

[26] Ledziński D., Bujnowski S., Marciniak T., Pedersen J.M., Gutierrez Lopez J., 2013. Network Structures Constructed on Basis of Chordal Rings 4th Degree. Image Processing and Communications Challenges 5. Advances in Intelligent Systems and Computing 233, 281-299.

GRAFY REFERENCYJNE

Streszczenie

W artykule zdefiniowano pojęcie Grafu Referencyjnego, za pomocą którego można modelować sieci teleinformatyczne w celu ich optymalizacji. Zdefiniowano parametry takiego typu grafów, które zostały porównane z parametrami modyfikowanych pierścieni cięciwowych trzeciego i czwartego stopnia. Do modelowania poszczególnych przypadków opracowano program symulacyjny. Zaprezentowano opis tego programu oraz wyniki otrzymane przy wyszukiwaniu grafów referencyjnych dla szerokiego spektrum danych wejściowych.

Słowa kluczowe: grafy, grafy referencyjne, projektowanie sieci telekomunikacyjnych