ANALELE UNIVERSITĂŢII
"EFTIMIE MURGU" REŞIŢA
ANUL XII, NR. 1, 2005, ISSN 1453-7394

Constantin Bărbulescu, Marius Groza, Gabriel Limbean

Optimal Equipment Programming for Realizing an Ensemble of Electrical Overhead Lines

The main purpose of this paper is to elaborate a calculation program in Pascal language, using Delphi environment. This calculation program is designed to solve the power engineering optimization problems using the method of optimal Hamiltonian path and circuit, solved with the so called "Latin multiplication". For illustrating the use of the algorithm and the calculation program we present a problem of optimal equipment programming for realizing an ensemble of electrical overhead lines.This paper is structured in 4 parts. In the first part of the paper we present the application as a problem of optimal Hamiltonian path and circuit. In the second part of the paper, we determine the optimal Hamiltonian path (circuit) in a graph using the so called Latin multiplication method. In the third part of the paper we present a representative numerical application. In the fourth part of the paper it is described the calculation program.

1. Presenting the problem

Let us consider an enterprise specialized in constructing high voltage overhead power lines, using for this purpose a set of utensils. The transport of those utensils from one building site to another generally implies high costs. The transport expenses consist of a fix part and a variable one. The fix part is linked to removing of the utensils from the device and bringing them to the embarking station, as well as entering the new device in the arrival station. The variable part consists of the cost for the transport on the railroad.

We are required to establish the order in which we will execute the power lines in order to minimize the cost of transport of the utensils.

Additionally, we establish the following:

a) the cost of transport of the utensils on the railroad is proportional to the distance of transport, the minimization of the cost being equivalent to the minimization of the distance of transport;

b) the utensils are received on the initial point of the power lines and are shipped to the final point of the lines, which leads to the conclusion that generally the distance from the i building site to the j building site is not the same as the one from j to i.

c) to begin the approximation, we consider that in the initial moment the utensils may be found on any building site, and in the end we state the initial building site; in both alternatives the final building site may be any of the building sites, instead of the initial one.

Let us consider that the building sites of the power lines are the nodes of a graph, and that the railroads between them are the arcs of the graph (the transport lengths are the values of the arcs), then the solving of the problem is reduced to finding a Hamiltonian path (HP) of minimum value, which have the initial node corresponding to the initial building site.

The problem may be regarded from the point of view of returning with the utensils in the initial point of departure, in this case being necessary to find a Hamiltonian circuit (HC), of minimum value.

Analogous mathematical models may be obtained in other situations, leading to the necessity of knowing the Hamiltonian paths or the minimal value HP:

a) a certain system must go through a certain number of states; knowing the costs of going through states, we are required to determine the optimal succession of the states;

b) setting the order of launchings in production or repartitions of certain marks in the workshops of the mechanical and power services, knowing the costs of going from one mark to another .

2. Determining the optimal Hamiltonian path and circuit via the Latin multiplication

Via a special kind of multiplication, *the Latin multiplication*, we may successively find all the elementary paths of $\overline{1,n-1}$ length, those of n-1 length (if they exist) being the Hamiltonian paths.

To the G graph we assign the Latin matrices $[M]^{(1)}$ and $[\overline{M}]^{(1)}$ defined as follows:

$$[M]^{(1)} = \begin{cases} x_i x_j & if \quad (x_i, x_j) \in A, i \neq j \\ 0 & if \quad (x_i, x_j) \notin A, i \neq j \\ 0 & if \quad i = j \end{cases};$$

$$[\overline{M}]^{(1)} = \begin{cases} x_j & if \quad (x_i, x_j) \in A, i \neq j \\ 0 & if \quad (x_i, x_j) \notin A, i \neq j \\ 0 & if \quad i = j \end{cases}. \qquad (1)$$

We define the Latin multiplication of the matrices $[M]^{(1)}$ and $[\overline{M}]^{(1)}$, noted as follows:

$$[M]^{(2)} = [M]^{(1)} \mathcal{L} [\overline{M}]^{(1)} \qquad (2)$$

$$m_{ij}^{(2)} = \bigcup_{l} m_{il}\overline{m}_{lj} = \left\{ m_{il_1}\overline{m}_{l_1 j}, \ldots\ldots, m_{il_s}\overline{m}_{l_s j} \right\} \qquad (3)$$

where $s < n$, and the reunion does not contain any zero terms nor any terms for which the sum $\sum_{l=1}^{n} m_{il}\overline{m}_{lj}$ contains repeated letters. Then $x_i x_{l_1} x_j$, $x_i x_{l_2} x_j$, ..., $x_i x_{l_s} x_j$ are all path of elementary length 2 which unite the x_i node with the x_j node.

Analogously we define the Latin matrices $[M]^{(2)}, [M]^{(3)}, \ldots, [M]^{(k)}$

$$[M]^{(k)} = [M]^{(k-1)} \mathcal{L} [\overline{M}]^{(1)} \qquad (4)$$

$$m_{ij}^{(k)} = \bigcup_{l} m_{il}^{(k-1)}\overline{m}_{lj} = \left\{ m_{il_1}^{(k-1)}\overline{m}_{l_1 j}, \ldots\ldots, m_{il_s}^{(k-1)}\overline{m}_{l_s j} \right\} \qquad (5)$$

where $s < n$, and the reunion does not contain any zero terms, nor repeated nodes.

The elements of the $[M]^{(k)}$ matrix will give all the elementary paths of k length of the G graph. Finally, the elements of the $[M]^{(n-1)}$ matrix will give all the Hamiltonian paths of the graph (if they exist).

If we are only interested in the HP, then the volume of calculation may be considerably reduced by introducing the $[\overline{M}]^{(k)}$ matrices, which may be obtained from $[M]^{(k)}$ by cutting off the first peak of each element (elementary path of k length). The following relation may then be written, which, applied accordingly, to n, may significantly reduce the calculations:

$$[M]^{(k)} \mathcal{L} [M]^{(k-1)} = [\overline{M}]^{(1)} \qquad (6)$$

In order to apply the abovementioned method, the following steps must be observed:

a) writing the Latin matrices $[M]^{(1)}$ şi $[\overline{M}]^{(1)}$ of the *G* graph;

b) by corresponding successive Latin multiplications we will obtain the Latin matrix $[M]^{(n-1)}$, and its elements are all HP (if there are such paths in the *G* graph);

c) calculating the HP value, in the end selecting the optimal HP (of minimal value).

The Hamiltonian circuits are determined calculating the Latin matrix $[M]^{(n)*}$, defined by the following relation

$$[M]^{(n)*} = [M]^{(n-1)} \mathcal{L} [\overline{M}]^{(1)} \qquad (7)$$

In order to obtain the HC (if it exists) for every element of the $[M]^{(n-1)}$ matrix we assign as initial element the peak corresponding to the line index of the element.

3. Numerical application

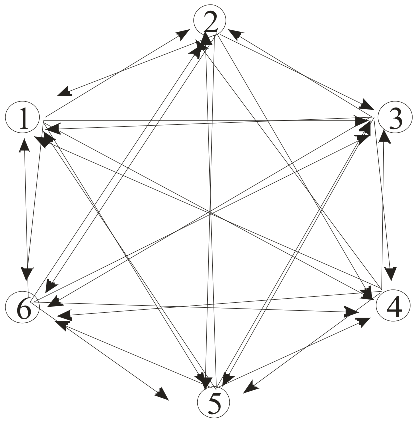
Let us consider an enterprise specialized in constructing EOL, which, during a year, has to produce 6 high voltage EOL, using a set of utensils. The distances of transport between the 6 building sites are known and presented in the $[C]$ matrix of values. The elements c_{ij} of the $[C]$ matrix represent the distances between the building sites, in km.

We are required to set the order in which the EOL are to be executed in order to minimize the cost of transport for the utensils.

	1	2	3	4	5	6
1	0	625	604	467	389	297
2	486	0	876	541	888	344
3	480	653	0	569	330	338
4	257	672	710	0	196	580
5	345	734	424	282	0	685
6	220	211	542	537	416	0

We will now present two alternatives of applying the Latin multiplication method, which differ through their way of defining the partial graph to which the Latin multiplication method is applied.

First, let us consider an asymmetrical partial graph (from the two "symmetrical" arcs, we will store the one with the lowest value as being part of the partial graph). For this graph we will write the Latin matrices $[M]^{(1)}$ and $[\overline{M}]^{(1)}$, and we will do the following operations:



$$[M]^{(2)} = [M]^{(1)} \mathcal{L} [\overline{M}]^{(1)}$$
$$[M]^{(4)} = [M]^{(2)} \mathcal{L} [\overline{M}]^{(1)}$$
$$[M]^{(5)} = [M]^{(4)} \mathcal{L} [\overline{M}]^{(1)}$$

Figure 1.

The elements of the $[M]^{(5)}$ matrix will give all the HP.

1234 54 121 1621465 33 51453164 415
5612 56 4151411 241512 2454
3416 64 521321 62451
16 $[M$ 341 64521
$[\overline{M}]^{(1)}$ 35 51
4563 12 351 361
6162 12
6465 1
5341 41
3132 62
2422 1
1151 64
6165 14
4 51
$[M]^{(5)}$ 54
2543 4
6245 51
3652 61
4345 65
2116 1
2352
2445
5224
6452
4524
5

$\left[\overline{\mathrm{M}}\right]^{(2)}$

2345
6652
6246
4552
4452
3453
2435
2561
2456
2131
2212
4

$\left[M\right]^{(1)}$

3313
2343
5364
4145
5515
2661
6264
65

3654
1
3524
1
3624
1
3245
1
3645
1
65
64
62
365
364
362

4515 1		164532	162453	153624	165324		145362

4232
1451
2545
2545
3435
3632
6123
4563
3645
2111
2345
66
$[M]^{(4)}$
2642
53
$[M]^{(2}$
2241
1123
4562
412
$[\overline{M}]^{(1)}$
4561
126
T
he
secon
d
altern
ative
requir
es
the
segm
entati
on of
the
partia
l
graph
from
the
first
altern

	145362		153264		

ative, by introducing also the arcs for which c_{ji} has a value equal to or just a bit higher than c_{ij}, but relatively low when compared to other elements of the $[C]$ matrix: 1-4 (467 compared to 257), 1-5 (389 comp						

ared to 345), 1-6 (297 compared to 220), 2-6 (344 compared to 211), 5-3 (421 compared to 330) and 5-4 (282 compared to 198). After writing the Latin matrices $[M]^{(1)}$ and $[\overline{M}]^{(1)}$ we do the Latin						

operations. The $[M]^{(5}$ matrix contains a number of 82 HP, being synthesized in the next table those which value is under 2000, noticing the fact that also exists paths between the peak 1 and the peak 2, 3, 4, 6, the

peak 2 and 1, 3, 4, 6, the peak 4 and 1, 2, 6, the peak 5 and 1, 2, 4, peak 6 and peaks 1, 2, 3. The HP marked in the table are the optimum ones for the situations when it is imposed the initial peak, the solution						

„optimum optimorum” being 3-5-4-1-6-2, having the value 1377. 365241 362451						
2	264531 265341 245361		216453 241653 261453 264153	216534 265314 261534 215364		245316 241536 214536
$[M]^{(5}$ 3	364521 352641 326451 345261 326541 365241 362451	341652 361452 364152 316452 345162 354162		352164 316524 365214 361524 352614 362154 351624 315264 321654 326154 326514	326415 326145 321645 362415 316245 362145 324165	352416 324516 345216 341526 314526
4	453621 453261	415362 416532 453162				453216 415326
5	536241 532641	534162				532416
6	645321 624531 653241	641532 614532				

Table 1.

123456 453621 165413 216453 191025 181936 214517 001124 153636 621453 178319 372319 593261 451882 926153 434624 531185 118881 145362 163514 362451 163327 453261 183621 624531 668153 541621 377284 531621 607315 362419 001636 145219 572941 536216 164264 531198 017345 162162 030536 241176	2619522 1453662 4153241 7272453 1661453 2221736 HP	Value	Nr.	HP	Value	The HC is determined based on the relation: 19393 26541 12371 94631 62451 03519 43362 415Nr.

$[M]^{(n)*}$ (8)

For *n = 6* we obtain the Latin matrix $[M]^{(6)*}$,which is a diagonal matrix.

By

852453
611718
183516
241724
315341
621755
624165
319321
936215
417063
253621
4Nr.

analyzing the $[M]^{(6)*}$ matrix, we come to the conclusion that there are 7 HC, the values of which are presented in the following table, the optimal cycle being 1-4-5-3-6-2-1, with the value 2121 (second line in the table). We notice the existence of other two HC with values close to the optimal HC (the third an the fourth line of the table)

335316 241950 826415 319482 119237 261453 165020 326154 HPValue	1645321	2416532	3264153	4536214	5326415	6453216
	1453621	2614532	3624153	4532614	5362415	6245316
$[M]^{(6)*}$ =	1624531	2164532	3261453	4165324	5326145	6532416
	1536241	2641532	3162453	4153624	5362145	6415326
	1653241	2415362	3621453	4532164	5324165	6214536
	1453261	2145362	3241653	4153264	5321645	6241536

Table 2.

Nr.	Circuit	Value	Nr crt	Circuit	Value
1	1645321	2592	5	1532641	2601
2	1453621	2121	6	1553241	2585
3	1624531	2148	7	1453261	2303
4	1536241	2157			

Both ways to apply the Latin multiplication method are approximate methods (due to the reduction of the initial graph), with great chances for the second alternative to find the exact solution of the problem. Also, the second alternative allows determining HC close to the optimal solution or even the optimal solution itself, as in the present case.

4. LATIN calculation program

4.1. Overview

The CRITIC calculation program has been elaborated in Turbo Pascal language, using the Delphi environment. It finds the CP via a program graph and it solves a series of related problems.

The following aspects related to the calculation program may be noted:

a) The program solves all aspects related to the creation and actualization of the data base related to a certain application, as well as the loading and saving of the files that contain the data bases;

b) A large number of check-ups for the compatibility and the accurateness of the elements in the data base, with warning or error messages if necessary;

c) The partial graph may be set by the program or by the user;

d) The problem of the optimal HP is solved via the Latin multiplication method, using special storing techniques for the sparse matrices, with messages according to each case that may present itself during the solving process;

e) The problems also having a didactic side, the visualization of the results may be done according to the user's wish: from the sole display of the final results to the visualization in a work window of all the intermediate results for each iteration, with the option of browsing them;

f) The option of saving or listing the result files is available.

4.2. Delphi® environment

Introduced as Object Pascal 7.0, the firm that produced it was not able to give up the codename Delphi, due to its market popularization.

Delphi introduces several concepts unknown in the Pascal language, such as: classes, properties, interfaces and objects. The classes derive from the old objects Pascal, the method concept setter-getter being introduces in an original manner.

Delphi is part of the RAD (Rapid Application Development) applications package, being a visual environment, via which we can easily build Windows interfaces.

The programming style is named OOP (Object Orientated Programming), known in the world of Borland developers as PME (Properties-Methods-Events).

There is a great similarity between the Delphi environment and Turbo Vision, both operating new objects and having a new way of treating events and objects. Due to the fact that Windows is a system based on messages, the Pascal should be modified so as to respond to messages, in order to comply with the present requirements. Delphi introduces the concept of notification procedure, which is actually a method of a class, this method is indirectly called by the system, its filling being actually a perpetual waiting after a user's action.

At this time, worldwide, for the standard of home user, the operating system is minimum Windows 98, and the industrial standard is Windows NT4. A strong tendency of migration to the NT engines is obvious, now existing the possibility for a simple user and a firm to use the same operating system (Windows XP) for different problems in order to achieve maximum efficiency. This operating system is promoted in all fields where a computer is required.

The integrated developed environment (IDE) Delphi has been chosen since Pascal is an easy language, which allows introducing easily and in a graduate manner all the Windows programming problems, without losing much of the specifics of the applications.

4.3. Using the program

LATIN program, elaborated in **Delphi**® environment, version 5, is a classical application for **Windows**®, offering a facile interface for the students, during the practical works for the ***Techniques of optimization in the power industry*** course, as well as for all those interested in such applications.

This third version is developed using the Delphi environment, for IBM-PC computers for the Windows® 98 engine. Delphi represents and integrated environment of development, offering a complete set of application for developing the "standard" Windows® application, based on the Pascal language, with the extensions offered by the Object Pascal.

The applications covered by the Delphi environment range from the office applications and / or utilitarian (Office Ready), it offers an intuitive simple base for the standards COM, DCOM, OLE, dB, RDBM, XML, HTML, MIDAS, to the servers of distributed applications.

The program package covers the following objectives:

- solves optimization problems for the power industry

- offers a didactical base for the works for the practical works for the ***Techniques of optimization in the power industry*** course.

When it is launched, the program window will have the following aspect:

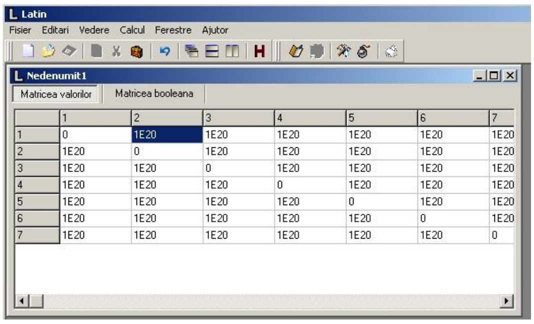


Figure 2.

A new file (empty) is loaded, and it contains a matrix which represents a graph with 7 nodes and which is to be filled in by the user with all the necessary data. This is the value matrix. The program also allows in this window the option to see the Boolean matrix corresponding to the truth values regarding the existence (value 1) or not (value 0) of a path between 2 nodes.

Before initialising the process of filling-in the elements of this matrix, the user must access the menu *Vedere – Optiuni*, or o click on the according button on the toolbar. This command will lead to another window, in which the user may specify the number of nodes of the graph, as well as other figures (the number of exact digits, the initial value of the data etc.).

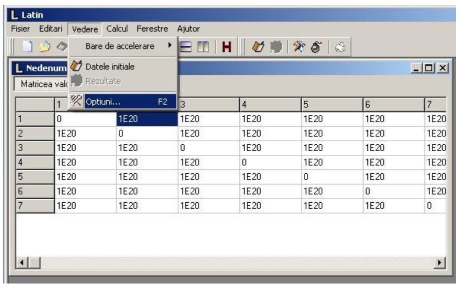


Figure 3.

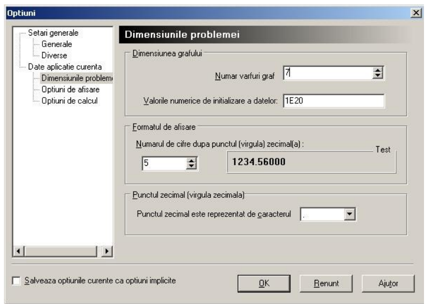
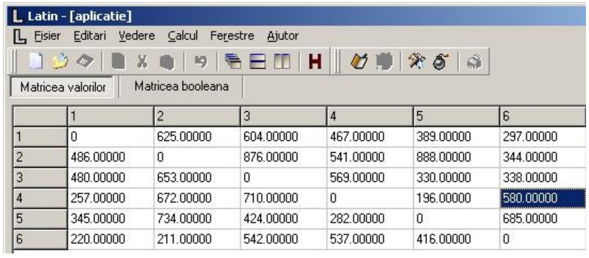


Figure 4.

After introducing the information necessary in order to fill in the matrix, the user has the option to save that file under a certain name, for a subsequent use.

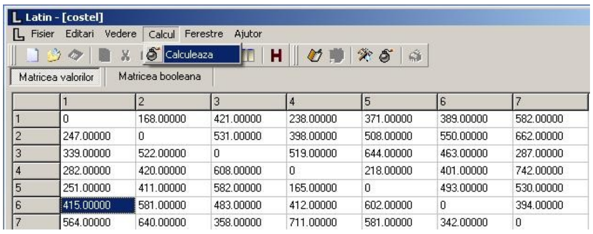
The window of the application for the matrix that contains the user data and the table with the existing arcs will have the following aspect.

After having gone through the process of filling-in the data matrix, the user accesses the menu *Calcul – Calculeaza*.



	1	2	3	4	5	6
1	0	625.00000	604.00000	467.00000	389.00000	297.00000
2	486.00000	0	876.00000	541.00000	888.00000	344.00000
3	480.00000	653.00000	0	569.00000	330.00000	338.00000
4	257.00000	672.00000	710.00000	0	196.00000	580.00000
5	345.00000	734.00000	424.00000	282.00000	0	685.00000
6	220.00000	211.00000	542.00000	537.00000	416.00000	0

Figure 5.



	1	2	3	4	5	6	7
1	0	168.00000	421.00000	238.00000	371.00000	389.00000	582.00000
2	247.00000	0	531.00000	398.00000	508.00000	550.00000	662.00000
3	339.00000	522.00000	0	519.00000	644.00000	463.00000	287.00000
4	282.00000	420.00000	608.00000	0	218.00000	401.00000	742.00000
5	251.00000	411.00000	582.00000	165.00000	0	493.00000	530.00000
6	415.00000	581.00000	483.00000	412.00000	602.00000	0	394.00000
7	564.00000	640.00000	358.00000	711.00000	581.00000	342.00000	0

Figure 6.

After selecting this option, the "Optiuni" window (seen earlier) will appear again and it will mainly allow to configure 3 parameters.

- the dimensions of the problem (tackled above);
- the display options: whether to print the initial data or not, the intermediary results, whether to clear or not the screen before displaying the results;
- the calculation options: whether the partial graph is generated by the program, or set by the user. The user is also allowed to access several options to modify the partial graph in case the latter is generated by the program.

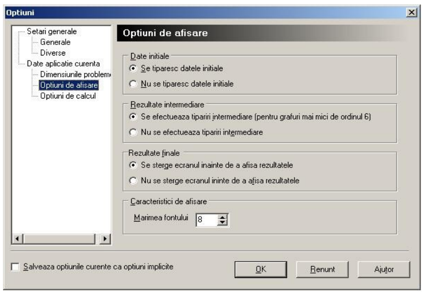


Figure 7.

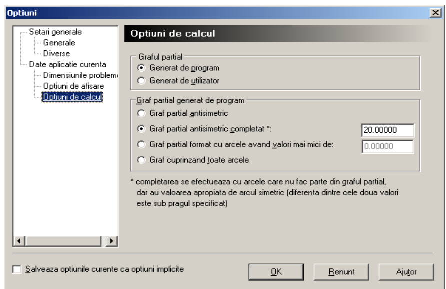


Figure 8.

By clicking the OK button, the calculation process is initiated. In the window which appears upon completion, the results of the calculation process are presented, according to the user's wishes. The application allows the export of the initial data, respectively of the results as text or rtf files (fig. 9), files that may be later opened via programs in the **Office**® package.

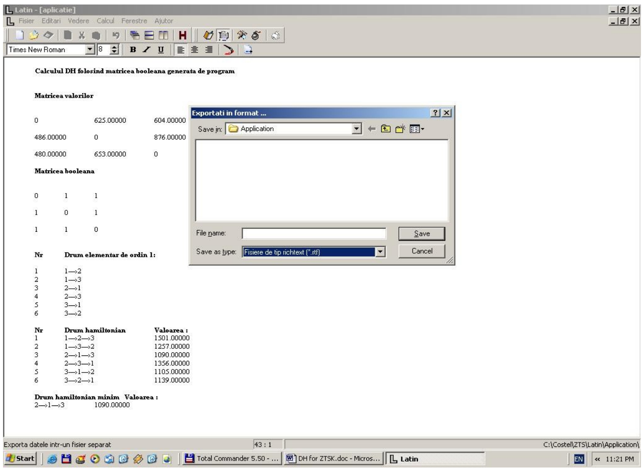


Figure 9.

References

[1] Kilyeni Şt., *Tehnici de optimizare în ingineria energetică*. Editura Orizonturi Universitare, Timişoara, 2004

[2] Kilyeni Şt., *Metode numerice. Algoritme, programe, aplicaţii în energetică, Ed. a 3-a.* Editura Orizonturi Universitare, Timişoara, 2004

[3] Kilyeni Şt., Groza M., Limbean G., *Tehnici de optimizare în ingineria energetică. Lucrări practice.* Editura Orizonturi Universitare, Timişoara, 2003

[4] Momoh J.A., *Electric Power System Applications of Optimization.* Marcel Dekker Inc., New York, 2000

[5] *Borland Delphi 3. Object Pascal Language Guide.* Borland International, 1997

[6] *Borland Delphi 3. Visual Component Library Reference.* Borland International, 1997

Addresses:

- Constantin Bărbulescu, student IVth year, "Politehnica" University Timişoara, Electrical and Power Engineering Faculty, Bd. V. Pârvan Nr. 2, Timişoara, Romania, e-mail costelbarbulescu@k.ro

- Asist. Drd. Ing. Marius Groza, "Politehnica" University Timişoara, Electrical and Power Engineering Faculty, Bd. V. Pârvan Nr. 2, Timişoara, Romania, e-mail marius.groza@et.upt.ro
- Asist. Drd. Ing. Gabriel Limbean, "Politehnica" University Timişoara, Electrical and Power Engineering Faculty, Bd. V. Pârvan Nr. 2, Timişoara, Romania, e-mail gabi.limbean@et.upt.ro