DESIGN OF AN ALGORITHM FOR GENERATING THE GRAPH OF A TECHNOLOGICAL PROCESS

LUCHIAN Mircea-Alin[1], GHEORGHIȚĂ Vlad[2]
[1]Faculty of Industrial Engineering and Robotics, Study program: IAII, Year of study: IV,
e-mail: luchianmirceaalin@gmail.com
[2]Faculty of Industrial Engineering and Robotics, TCM Department

ABSTRACT: In computer science, a graph is an abstract data type that is meant to implement the undirected graph and directed graph concepts from the field of graph theory within mathematics. A graph data structure consists of a finite (and possibly mutable) set of vertices (also called nodes or points), together with a set of unordered pairs of these vertices for an undirected graph or a set of ordered pairs for a directed graph. These pairs are known as edges (also called links or lines), and for a directed graph are also known as edges but also sometimes arrows or arcs. The vertices may be part of the graph structure, or may be external entities represented by integer indices or references. A graph data structure may also associate to each edge some edge value, such as a symbolic label or a numeric attribute (cost, capacity, length, etc.).

KEY WORDS: graph theory, graph data structures, directed graphs, abastract data types

1. Introduction

The technological process of machining a part can be designed in a very large number of variants, each of them ensuring that all the technical conditions imposed on the part are met [1]. Of all the possible machining variants, only one ensures the lowest cost.

A method that can be used to determine the optimal variant of a technological process is based on the use of graph theory, a theory that allows through the procedural model and the logical scheme associated with the problem, can simultaneously consider all the real possibilities of machining a part, indicating the most economical and fastest variant [4].

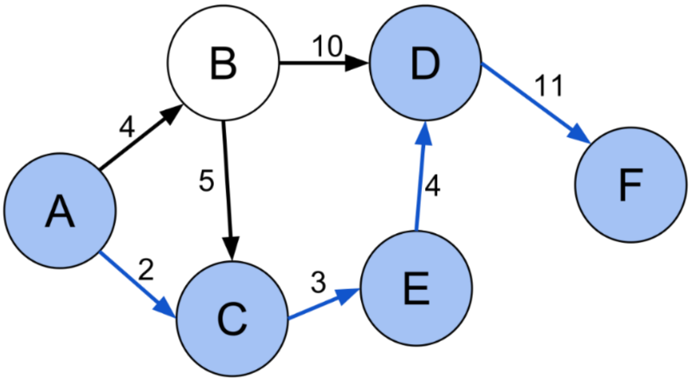


Fig. 1. Graphical example of a directed graph showing the shortest path [2]

In figure 1, the path in a graph consists of the sequence of arcs that allows one to pass from one vertex to another in the direction of their orientation, so that the final orientation of one arc coincides with the initial extremity of the next arc. In object-oriented programming, the hierarchy of objects (classes) in a program can be represented by a graph in which each node represents a class [3].

2. Current status

The development environment used to build the algorithm is MATLAB, a development environment for numerical computation and statistical analysis containing the programming language of the same name, created by MathWorks. MATLAB allows manipulation of matrices, visualisation of functions, implementation of algorithms, creation of interfaces and can interact with other applications. With its help, a directed graph can be generated in which the most efficient path is highlighted. For web-based communication with the program, the LabVIEW development environment was used, a graphical programming language that allows the development of applications using icons. Unlike textual programming languages, where instructions determine the execution of the program, LabVIEW uses instead data flow highlighted by an appropriate graphical presentation. [5]

The scheme of the principle of operation of the application is shown in Figure 2.

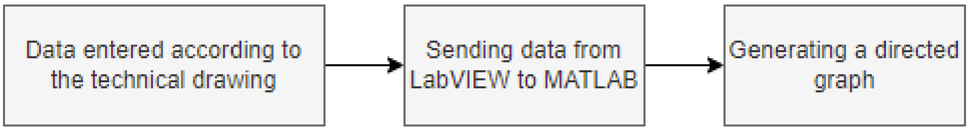

Data entered according to the technical drawing
Sending data from LabVIEW to MATLAB
Generating a directed graph

Fig. 2. Schematic diagram of the algorithm

Sending data from LabVIEW to MATLAB will be done using the OPAL-RT package. Depending on the execution drawing (Figure 3) and the machining required to make the desired part (Figure 4), the user will choose one by one all the specific details using a web interface provided by LabVIEW and the Web Services development package, until all the operations have been registered, as shown in Figure 4. It is important to keep in mind that the grouping of surfaces and operations within the program is done after the landmark is attached to the tool machine.

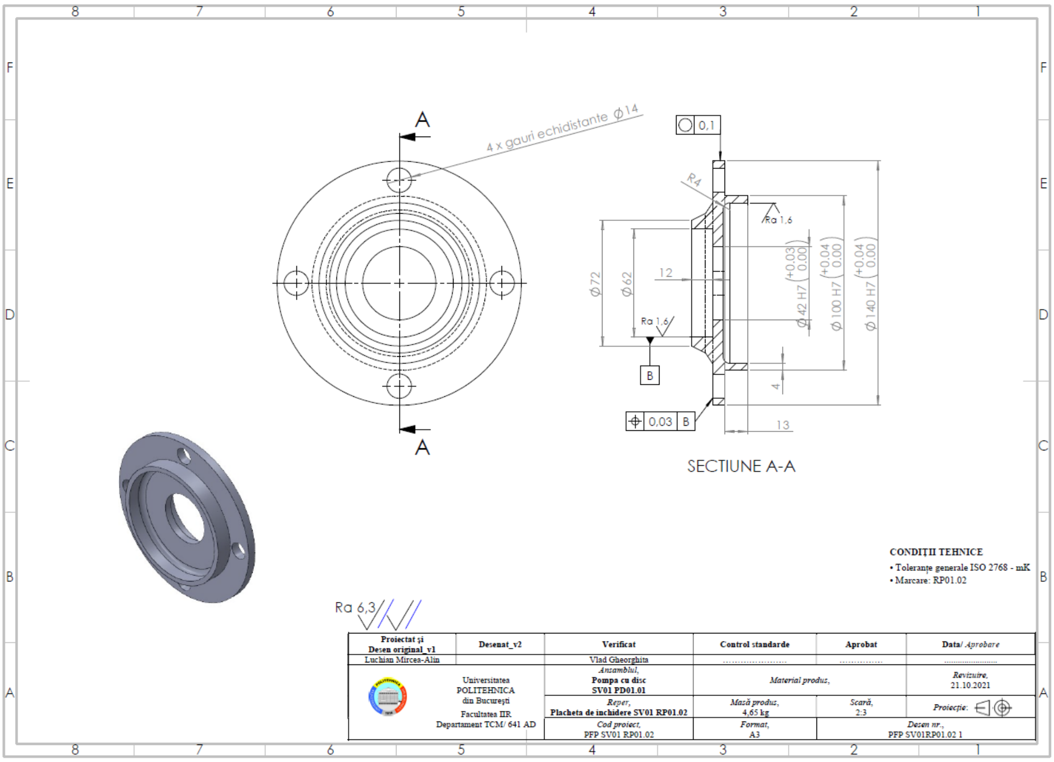
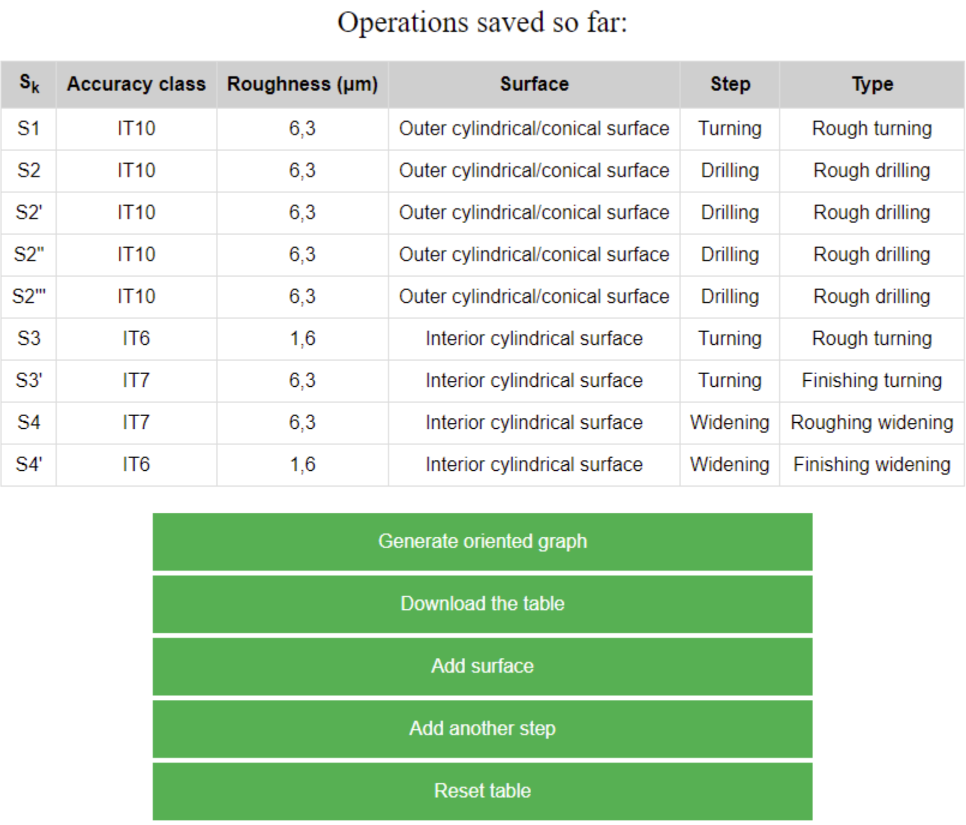


Fig. 3. Example of execution drawing

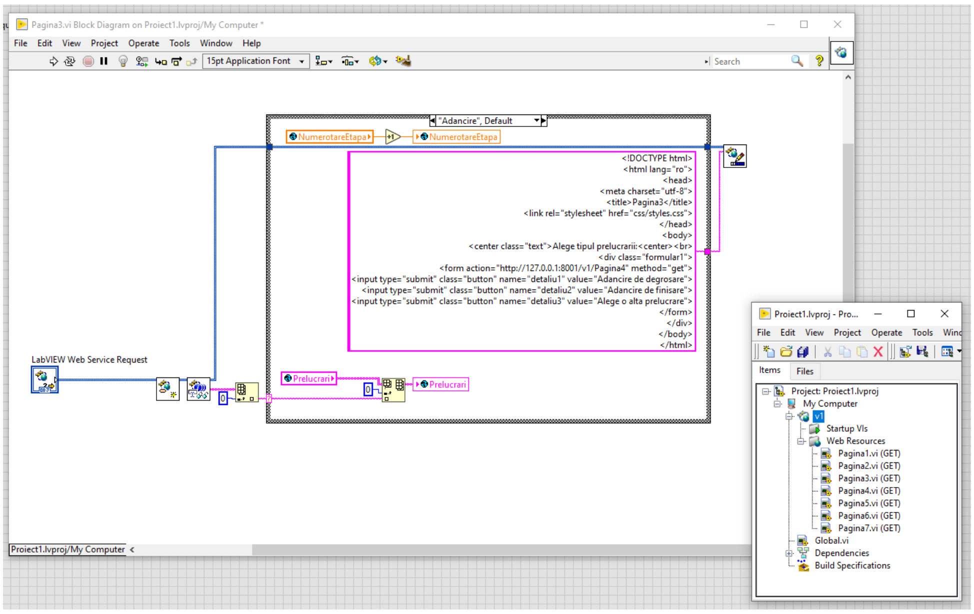


Operations saved so far:

S_k	Accuracy class	Roughness (μm)	Surface	Step	Type
S1	IT10	6,3	Outer cylindrical/conical surface	Turning	Rough turning
S2	IT10	6,3	Outer cylindrical/conical surface	Drilling	Rough drilling
S2'	IT10	6,3	Outer cylindrical/conical surface	Drilling	Rough drilling
S2"	IT10	6,3	Outer cylindrical/conical surface	Drilling	Rough drilling
S2"'	IT10	6,3	Outer cylindrical/conical surface	Drilling	Rough drilling
S3	IT6	1,6	Interior cylindrical surface	Turning	Rough turning
S3'	IT7	6,3	Interior cylindrical surface	Turning	Finishing turning
S4	IT7	6,3	Interior cylindrical surface	Widening	Roughing widening
S4'	IT6	1,6	Interior cylindrical surface	Widening	Finishing widening

Fig. 4. Current web interface with a table populated with user-made entries

Depending on the need, the user can generate the oriented graph with the choices that have been registered or download them as a table. Downloading the table will save a .csv file which can be read by Microsoft Excel. Pressing the "Reset table" button will delete all entries in the table.

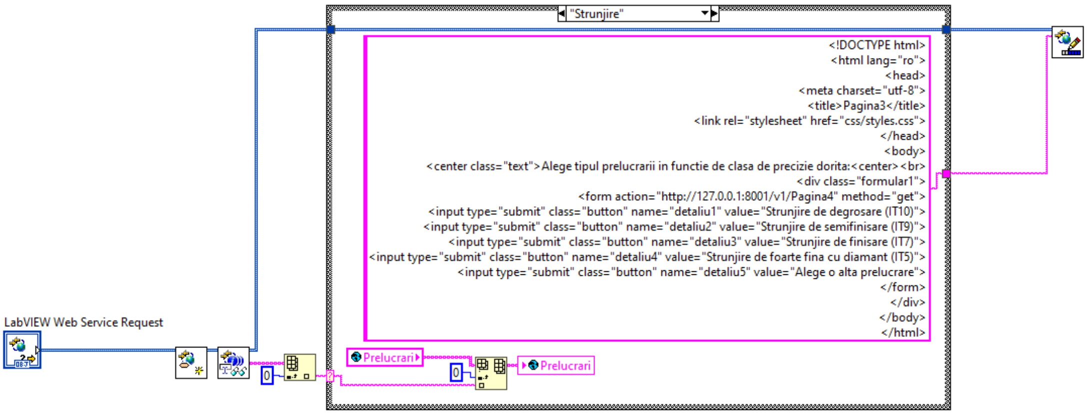


```
<!DOCTYPE html>
<html lang="ro">
<head>
<meta charset="utf-8">
<title>Pagina3</title>
<link rel="stylesheet" href="css/styles.css">
</head>
<body>
<center class="text">Alege tipul prelucrarii:<center><br>
<div class="formular1">
<form action="http://127.0.0.1:8001/v1/Pagina4" method="get">
<input type="submit" class="button" name="detaliu1" value="Adancire de degrosare">
<input type="submit" class="button" name="detaliu2" value="Adancire de finisare">
<input type="submit" class="button" name="detaliu3" value="Alege o alta prelucrare">
</form>
</div>
</body>
</html>
```

Fig. 5. LabVIEW project and one of the sub-programs that adds new web page entries to the global variable

The global variable is used for storing records within the program and using the Web Services package a processed text is sent in the form of a HTML editing language (Figure 6) to be read and interpreted by a browser.

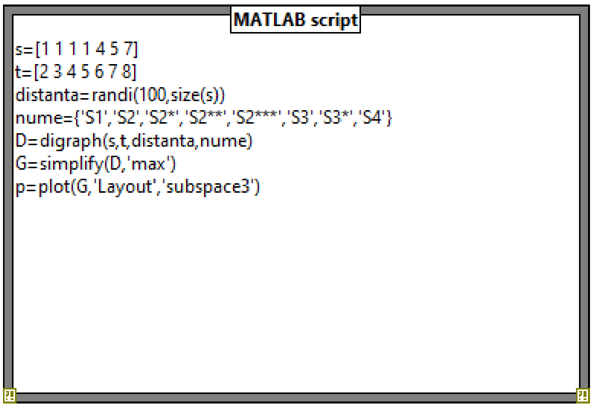
The functions in this package exchange data with the programs in the project over a local network, sending a response to HTTP requests from clients.



```
<!DOCTYPE html>
<html lang="ro">
<head>
<meta charset="utf-8">
<title>Pagina3</title>
<link rel="stylesheet" href="css/styles.css">
</head>
<body>
<center class="text">Alege tipul prelucrarii in functie de clasa de precizie dorita:<center><br>
<div class="formular1">
<form action="http://127.0.0.1:8001/v1/Pagina4" method="get">
<input type="submit" class="button" name="detaliu1" value="Strunjire de degrosare (IT10)">
<input type="submit" class="button" name="detaliu2" value="Strunjire de semifinisare (IT9)">
<input type="submit" class="button" name="detaliu3" value="Strunjire de finisare (IT7)">
<input type="submit" class="button" name="detaliu4" value="Strunjire de foarte fina cu diamant (IT5)">
<input type="submit" class="button" name="detaliu5" value="Alege o alta prelucrare">
</form>
</div>
</body>
</html>
```

Fig. 6. Example of one of the diagrams of a VI (Virtual Instrument) sub-program for displaying a web page

The incoming data is entered into a script using the OPAL-RT package (figure 7) where the entries in this structure will be replaced by the stored entries in the global variable. The references will receive nodes and edges by 2 vectors (one of sources and one of targets) used to make the directed graph after assigning a length to the edges and naming the nodes. The oriented graph is called using the *digraph* command and is displayed (figure 8) using the *plot* function with a layout layout that makes the edges as easy to read as possible.



```
s=[1 1 1 1 4 5 7]
t=[2 3 4 5 6 7 8]
distanta=randi(100,size(s))
nume={'S1','S2','S2*','S2**','S2***','S3','S3*','S4'}
D=digraph(s,t,distanta,nume)
G=simplify(D,'max')
p=plot(G,'Layout','subspace3')
```

Fig. 7. Sample script from MATLAB in LabVIEW

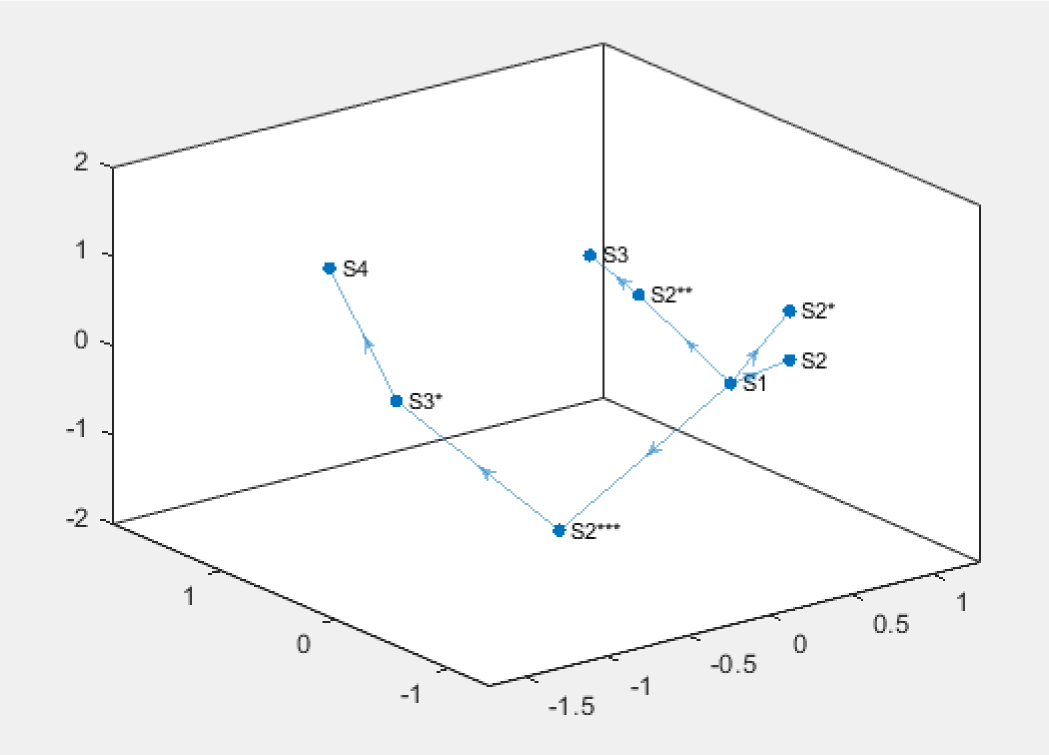


Fig. 8. Oriented graph generated at the end

3. Conclusions

Displaying and generating a directed graph can be done in MATLAB and displayed in a web service using LabVIEW. The resulting application is a tool that helps the user when it comes to assisting the planning of activities in a technological process. The role of the application is to provide and generate a graph with the chosen processes in a correct order. This leads to optimisation, standardisation of activities and increased efficiency.

Improvements are needed in the algorithm so that each operation has a value that corresponds correctly to the real manufacturing process, to respect the machining constraints (the graph will take into account the order of the machining, e.g. widening or deepening only after drilling; finishing after roughing and semi-finishing; threading after drilling) and the web page (the page should be user-friendly, to be able to display the oriented graph correctly and save it in a . png or .jpeg, be able to save the operation table in a .csv format). The application will use functions such as *shortestpath* and *highlight* that determine the shortest path, highlight it by thickening the edges, and color it in a desired color.

4. Bibliography

[1]. https://en.wikipedia.org/wiki/Graph_(abstract_data_type), date of access 07/05/2022
[2]. https://en.wikipedia.org/wiki/Shortest_path_problem, date of access 07/05/2022
[3]. https://ro.wikipedia.org/wiki/Teoria_grafurilor, date of access 07/05/2022
[4]. Jungnickel D. (2005), Graphs, Networks and Algorithms, 2nd edition, publisher Springer-Verlag Berlin Heidelberg, ISBN 3540219056
[5]. Travis J., Kring J. (2006), LabVIEW for Everyone: Graphical Programming Made Easy and Fun, 3rd edition, publisher Prentice Hall, ISBN 013185672