Algorithm for create network diagrams for planning maintenance of agricultural machines

Anton Popov[1,*] and *Vladimir* Lukhanin[2]

[1]Don State Technical University, pl. Gagarina 1, Rostov-on-Don, 344000, Russia.
[2]Azov-Black Sea Engineering Institute - affiliate Don State Agrarian University str. Lenina 21, Zernograd, 347740, Russia

Abstract. Reducing the maintenance man-hour is one of the ways to improve of operation efficiency of agricultural machines. The total maintenance time of the agricultural machine is determined by the critical path and other network planning parameters. The network model in the form of network diagrams is used for visualization and analysis of maintenance processes. Building network diagrams be not a trivial task and it takes a long time to build. Therefore, automating the creation of network diagrams of technological maintenance of agricultural machines is an urgent task. This work aims to develop an algorithm for creating and analyzing network diagrams for planning the processes of technological maintenance and repair of various agricultural machines. When building an algorithm for creating network diagrams, the methodological apparatus of network planning and basic algorithmic structures were used. The algorithm is realized in the Matlab programming language. The article presents the flowchart and a description of the formation of an array of all and full paths. The description of identifying fictitious operations in the network diagram is given. The present algorithm calculates the maintenance man-hour of paths, determines the critical path of the network diagram and visualizes it on the graph, and also calculates the time reserves of the specified work. The logic of the algorithm is developed in accordance with the rules for building network models and excludes dead-end and tail events, closed loops. Thus, the algorithm will increase the efficiency of solving practical tasks of network planning of maintenance of agricultural machinery, shorten working hours and time for the building of the network diagram and reduce the associated possible errors.

1 Introduction

Reducing the maintenance man-hour of maintenance operations and improving the adaptability of agricultural machines to checking and adjustment is one of the ways to improve of their efficiency of operation. Technological maintenance of an agricultural machine consists in performing a number of interrelated works and operations [1]. When studying the maintenance process, you can identify paths – a set of operations that are performed in succession. They can be organized by the type of work or by the machine nodes

---

* Corresponding author: popov_a_ju@mail.ru

being serviced. In accordance with network planning, fictitious operations can be highlighted from the all set of operations. Without them, it is impossible to perform subsequent work.

The total maintenance time of the agricultural machine is determined by the critical path with the maximum duration [2].

The critical path and the paths close to it determine the time reserves of the work carried out, the possibility of reducing the maintenance man-hour and the possibility of optimizing the maintenance of agricultural machines [2, 3]. The network model in the form of a directed graph is used for visualization and subsequent analysis of maintenance processes. The vertices of this graph are maintenance operations. The edges of the graph define the relationships between operations.

However, building network diagrams of the maintenance and setting processes of various agricultural machines be not a trivial task and it takes a long time to build [4]. With that, the complexity and maintenance man-hour of developing network diagrams increases significantly with the increase in the number of serviced nodes and operations. Also the number of possible errors in the calculations is increases.

The analysis of the software market indicated that there are few programs for network planning and process management. This is due to the narrow specialization of calculations. In most cases, building network diagrams and planning events is performed using Microsoft Excel. But the calculations in it are cumbersome and inconvenient.

Therefore, automating the creation of network diagrams of technological maintenance of agricultural machines to speed up calculations and reduce the number of errors associated with this is an urgent task.

This work aims to develop an algorithm for creating and analyzing network diagrams for planning the processes of technological maintenance and repair of various agricultural machines.

2 Materials and methods

When building an algorithm for creating network diagrams, the methodological apparatus of network planning and management, also basic algorithmic structures were used [2, 5]. The algorithm is realized using the Matlab mathematical package, which contains a large library of mathematical functions

In the general structure of the algorithm, the following sequential stages appear definable: 1) forming an array of all paths with fictitious operations; 2) calculating the maintenance man-hour of all network graph paths and determining the critical path; 3) deleting zeros at the end of an array rows of full paths; 4) excluding fictitious minor operations and defining full paths; 5) calculating time reserves of operation execution; 6) visualization of calculation results in the form of a table and a network diagram.

3 Results and discussion

Initial data in the form of three arrays, labeled *S*, *T* and *W*, are entered by the user at the initial stage of the calculation. To do this, the numbers of operations of technological maintenance of the analyzed agricultural machine are written sequentially in the input field. The list of numbers of all technological maintenance operations is written in the array *S*, for example, the setting and adjustment operations. The list of operations is written to the array *T*. Each of the given operations in the array S is based on this list of operations T [4]. The list of maintenance man-hour work, corresponding to the arrays *S* and *T*, are written in the array *W*. The flowchart of the algorithm for determining all the paths of the network graph with the formation of an array of performed *LA* operations is presented in figure 1.
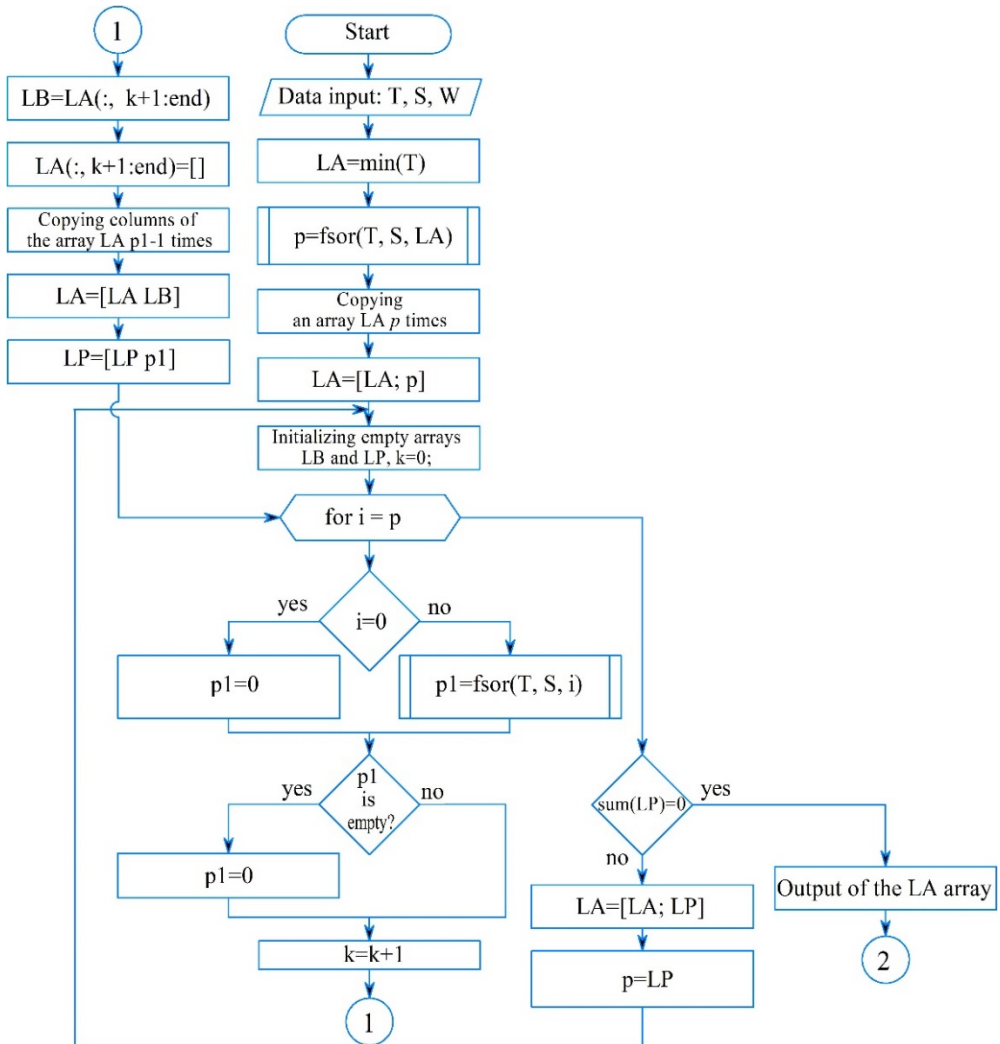
Fig. 1. Flowchart of the formation of an array of operations performed

Since one operation can to be based on several other operations, the number of n_c connections of the network graph will be more so than the entire number of works. And the operation numbers will be duplicated in the S and T arrays. With, the size of the arrays S, T, and W will be the same and equal to the number of links n_c.

The source vertex of the network graph is determined after entering the arrays T, S and W, by finding the minimum element in the array T. The source vertex number is written to the *LA* array.

Then the numbers of operations p that to be based on the source vertex must be determined. For this, the user function fsor() of Matlab, which will be used repeatedly later, is introduced into the program algorithm. The function listing is presented in figure 2.

```
function [f]=fsor(A, B, n)
    c=A==n;
    f=B(c);
```

Fig. 2. Listing of the fsor() function to define base operations

The operation numbers defined by the user function fsor() are copied to the array *p* and are associated with the array *LA* source vertex in the network graph.

Then, the operation numbers from the array p must be added to the array *LA* in forms the second line (figure 1). Since the number of elements in the array *p* can be more than one, therefore you must first increase the number of columns in the array *LA* and make it equal to the array *p* length. The source vertex number must be written in each first row of the new formed columns of the *LA* array. The repmat() function of Matlab is well suited for the performing this operation.

The *LA* array is formed after the executed actions. It contains two rows and an arbitrary number of columns. With that the number of columns depends on the number of operations and is based on the source vertex. Repeating source vertex number will be written in the first column. The operation numbers that are related with the source in the network graph will be written in the second column.

The "while" loop with the postcondition is the next operation. In this loop, the rows with the operation numbers of the network graph associated with the operations from the preceding row are added sequentially to the *LA* array.

The "while" loop begins by initializing the auxiliary empty arrays *LB* and *LP*. A variable $k=0$, which will serve as a counter in a sub-loop "for" with a certain number of iterations, is created.

In the "for" loop, the values of the elements from the array p defined earlier are assigned sequentially to the variable *i* at each iteration. With that the beginning of each iteration, the variable *i* is compared to zero. If $i=0$, then 0 is written to the new auxiliary array *p1*. Otherwise, the user function fsor() is run and identified a list of operation numbers which base on the *i* operation. Then it writes them to the array *p1*. Thereafter, the array *p1* is checked for the content of elements in it. If the array *p1* is empty, then 0 is written to it.

Further, the counter *k* is incremented by 1. Then the values of the *LA* array columns written to the auxiliary array *LB*. They are located to the right of the current number of the *k*-th column. After writing, they are removed from the *LA* array.

The *LA* array is complemented with copies of its *k*-th column, using the Matlab repmat() function. The number of copies must be equal to the length of the array *p1*. Then we append to the elements of the array *LA* on the right the columns consisting of the elements of the auxiliary array *LB*, which was written earlier. And the operation numbers from array *p1* to array *LP* are written for use in subsequent iterations of the "for" loop. Then the for loop repeats until all the elements of the array *p* are finished.

The element sum of the *LP* array is compared to zero after the "for" loop ends. If this is the case, then the "while" loop is exited. Otherwise, the operation numbers from the *LP* array are appended to a new line in the *LA* array. Also, the values of the p array are rewritten to new values from the *LP* array. Then the next iteration of determining the list of operation numbers that base on the operations written in the array p is started. With that all previously recorded operation numbers are removed from the auxiliary arrays *LB* and *LP*. This makes the *LB* and *LP* arrays empty again.

The next "for" loop with the new values of the array p is started and the process is repeated until the sum of the elements of the array *LP* becomes zero. Only in this case, the "while" loop ends. As a result of the *LA* array containing all the paths of the network graph operations being performed is formed.

Next, the maintenance man-hour of all paths of the *LA* array is calculated. The critical path of the network graph, that is, the sequence of operations for setting up the agricultural machine with the highest maintenance man-hour is determined. These operations are performed using standard Matlab mathematical functions, logical structures of the branching, and iterations. All calculated paths and maintenance man-hour are written to the array *L*.

Due to the fact that paths in a network graph can have a different number of operations, this leads to the need of storage a list of operations of all paths of different lengths in a single array. This is not possible for the "array" data type.

When forming an array L, this problem does not occur. Because as a result of the "while" loop, zeros are appended to the list of operations of all paths. As a result of all paths become equal in the number of elements and correspond to the length of the path with the largest number of operations. Also, zeros do not affect the results of calculations when calculating the maintenance man-hour of paths.

However, when the calculation results are displayed in the program window, these zeros will also be displayed. This will worsen the visualization of the results for a user who is not familiar with the calculation features. Therefore, in the algorithm of the program provides for the removal of zeros at the end of the rows of the array *L* by using the standard join() and replace() functions of Matlab.

The determination of the full paths of the network graph, that is, paths without fictitious operations, is performed after determining all the paths of the network graph. The flowchart of the algorithm is presented in figure 3.

To perform this operation, the previously formed array of full paths *L* is written to the array *M*. Further, the "for" loop is run on the *j* columns of the array *M*, aside from the first and second columns. At each iteration of this loop, the current column of the array *M* is written to the column vector *m*. Then, using the standard Matlab unique() function, unique values are defined in the vector m and written to the vector *p*. Essentially the vector *p* contains the same data as the vector *m*, but without repetitions and in sorted order.

An auxiliary empty "*Is*" array is set to store later all the fictitious path numbers.

Further, the iteration of the elements of the vector p with checking the source vertex and sink vertex in the current column j of the array *M* is performed at each iteration of the "for" loop on the columns j of the array *M*. The source vertex is denoted by zero, and the sink vertex is the maximum element in the array *S*. If an element of the vector *p* is a sours vertex or sink vertex, then the "for" loop skips the subsequent operations with using the "continue" command and starts checking the next element of the vector *p*.

Otherwise, the operation to determine the position of the current operation number of the network graph in the vector *m* begins. To do this, the standard find() function of Matlab, which creates a new vector *u*, is used. The position indexes of the previous and next column of the array M are defined and written to the vectors *u1* and *r*, using the vector *u* and the unique() command.

Further, the number of operations associated with the current operation number *i* is checked. The number of operations in vector *u1* must be greater than 1, and the number of operations in vector *r* is equal to 1. If this is not the case, the "for" sub-loop goes to the next element of the vector *p* and assigns its value to the variable *i*. If the condition is met, then the maximum maintenance man-hour in the array *M* for rows whose numbers are represented in the vector *u* is determined and it is written to the variable *k*. Then the path number with the maximum maintenance man-hour k is determined using the find() command and written to the vector *I*.
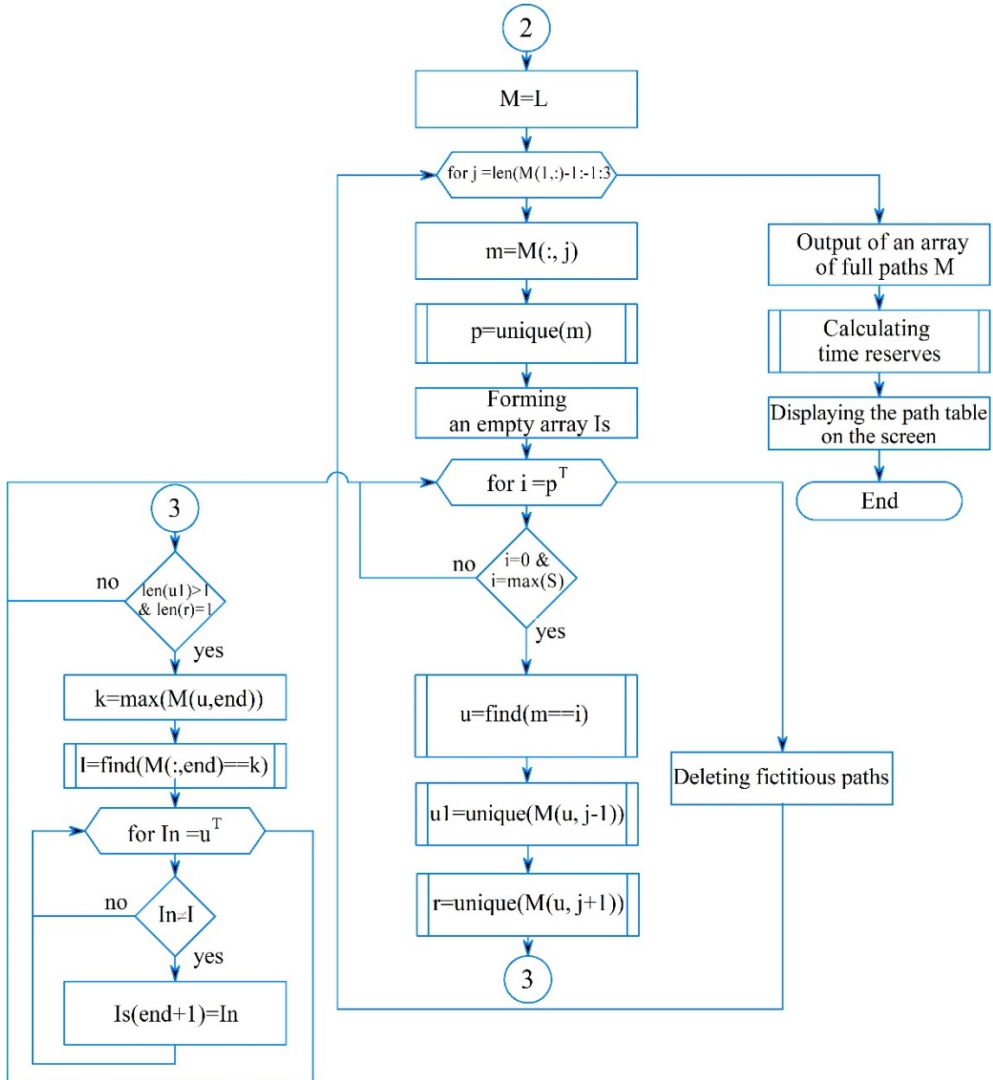
Fig. 3. Flowchart for determining full paths in a network diagram

Further, all the numbers of the fictitious paths are recorded, except for the number of the critical path. To do this, the second "for" sub-loop is started. It iterates over the values of the vector *u* and compares them with the value of *I*. If they do not match, then are written to the *Is* array.

When all the elements of the vector *p* end, the first "for" sub-cycle ends. The operation to delete rows with fictitious paths from the array *M* is started. To do this, the data in the *Is* vector is used, as the row numbers to delete.

Thus, an array with strings of full paths without fictitious operations and with maintenance man-hour remains in the array *M* at the end of all "for" loops.

Then the operation execution time reserves are calculated using the well-known network planning expressions on base of the *M*, *L* and *W* arrays and displayed on the computer screen.

Also, the network graph is constructed using the functions digraph() of Matlab. And the labeledge() function sets the weights of the network graph.

4 Conclusions

As a result of the research, an algorithm for automated building of network diagrams for planning and managing technological maintenance and repair of agricultural machinery and equipment is developed. The present algorithm determines the main and complete paths of operation of adjustment and setting, identifies fictitious operations, calculates the maintenance man-hour of paths, determines the critical path of the network diagram and visualizes it on the graph, and also calculates the time reserves of the specified work. The logic of the algorithm is developed in accordance with the rules for building network models and excludes dead-end and tail events, closed loops. The choice of the initial number of the source operation does not affect the calculation of the network model.

The algorithm will increase the efficiency of solving practical tasks of network planning of maintenance of agricultural machinery, shorten working hours and time for the building of the network diagram and reduce the associated possible errors.

References

1. M. Cecchini, F. Piccioni, S. Ferri, G. Coltrinari, L. Bianchini, A. Colantoni, *Preliminary investigation on systems for the preventive diagnosis of faults on agricultural operating machines.* Sensors, **21**, 1547 (2021).

2. M. B. Woolf, *CPM Mechanics: The critical path method of modeling project execution strategy*, (ICS-Publications, 2012).

3. N. E. Caicedo Solano, G. A. Garcia Llinas, J. R. Montoya-Torres, L. E. Ramirez Polo, *A planning model of crop maintenance operations inspired in lean manufacturing*, Computers and Electronics in Agriculture, v. **179**, 105852 (2020).

4. A. Yu. Popov, N. N. Sharaby, V.V. Zhurba, E.A. Chaika, *Analysis of the preparation technology of a row seeder for work,* E3S Web Conferences, v. **193,** 01035 (2020).

5. A. L. Kuznetsov, *Matrix method for finding the paths on weighted oriented graphs in the tasks of port net operational planning,* Bulletin of the state university of the sea and river fleet named after admiral S. O. Makarov, v. **12(2),** pp. 230-238 (2020).