Matching on Graphs

Shut V. N. [1], Svirski V. M. [2], Solomiyuk K. S. [3], Gryazev E. V.[4]
1) Brest State Technical University, Belarus Moskowskaja 267, 224017 Brest, Belarus rollv@mail.ru
2) Brest State Technical University,Belarus Moskowskaja 267, 224017 Brest,Belarus vsvirski@Gmail.com
3) Brest State Technical University, Belarus Moskowskaja 267, 224017 Brest,Belarus kcc-reg@yandex.ru
4) Brest State Technical University, Belarus Moskowskaja 267, 224017 Brest,Belarus gruazev-e@yandex.ru

Abstract: *This work is devoted to development of new algorithm of the decision of a matter about matching. In the result the algorithm of search maximal matching in graphs has been developed and realized, the estimation of its complexity, and comparison with existing algorithms have been made. Its characteristics, merits and demerits have been investigated too.*

Keywords: *Matching, Graphs, Trees.*

I. INTRODUCTION

Heuristic algorithms of the matter about matching are applied at designing engineering networks, communications, constructions of systems of support of decision-making in uncertain conditions, by development of bases of knowledge of intellectual systems etc. [1]. Tasks of such type concern to resetted tasks with exhibited time complexity. In this connection various heuristic approaches for construction of algorithms with polynomial time complexity are developed. There are algorithms of definition matching in the graph, based on use of streams in networks [2,3], imitating modeling [4], and other heurisms. In this work the method of definition maximal matching is offered, it is based on ideas of transformation any the graph in a tree in which search maximal matching is carried out.

The problem of search of matchings in graphs is solved for a long time, and for their finding a number of effective algorithms is developed:

- Ford-Fulkerson's algorithm [5];
- Edmond-Carp's algorithm [5];
- Kuhn's algorithm [5];
- Hopcroft-Carp's algorithm [5];
- Quantum algorithm [6];
- Parallel algorithm [7].

Lack of many algorithms is that they are applicable in the majority only to double-segmented graphs. Because of there is not in each real task double-segmented graph is applied it is essential lack. It is possible to notice, that algorithms which are applied by search of matchings in the double-segmented graph differ simplicity of understanding, simplicity of their realization, and smaller labour input. By looking at algorithms of search of matchings in any graphs it is possible to notice, that their basic advantage is universality, all other characteristics are worse, than for double-segmented graphs.
This implies, that development of algorithm which will be applicable to any the graph is expedient, it will be universal, but will have at the same time simplicity of realization and the small labour input inherent to double-segmented graph.

II. STATEMENT OF THE TASK. GENERAL IDEA OF THE ALGORITHM OF SEARCH OF MATCHING

The subset M of edges the graph G refers to matching if any two edges from M have no common top [8].

$$G = (V, E), \qquad (1)$$

where G – graph, V – number of tops , E – number of edges,. For example, in the graph, which is set on fig. 1 multitudes $M_1 = \{[\upsilon_2,\upsilon_3],[\upsilon_4,\upsilon_5],[\upsilon_6,\upsilon_8],[\upsilon_7,\upsilon_{10}]\}$ and $M_2 = \{[\upsilon_1,\upsilon_2],[\upsilon_3,\upsilon_5],[\upsilon_4,\upsilon_7],[\upsilon_6,\upsilon_8],[\upsilon_9,\upsilon_{10}]\}$; are matching. Thus M_2 is maximal matchings as, obviously, matchings in G cannot have more than $|V|/2$ edges. The task about matchings consists in a finding in the given graph $G=(V,E)$ maximal matchings M . If capacity of matchings is equal $|V|/2$, to the greatest possible value in the graph with $|V|$ tops, matchings refers to full, or perfect [8].

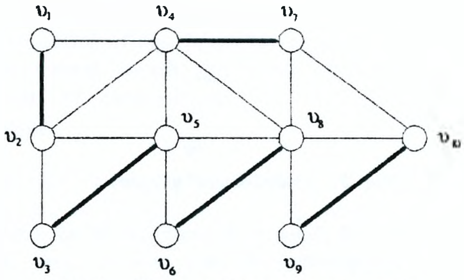

Fig. 1 – The matchings on any graph.

Integrated algorithm of search of matchings consists of the following stages:

1) Transformation the graph in a tree;
2) Construction of matchings for a tree;

3) Restoration the initial graph with application of algorithm of pushing out for preservation maximal matchings.

For transformation the graph in a tree the algorithm of wave search of contours in the graph is developed.

Let's look through the graphs, represented on fig. 2.

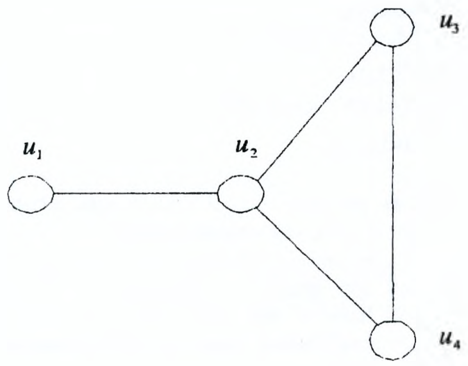

Fig. 2 – Initial graphs.

The top in the graph (for example) υ_1 any way gets out. From this top on the graph the wave is started which changes a flag of a condition of each top the graph through which it passes. If the flag of the condition on any top varies twice in the graph there are contours. After finding of a contour splitting top is made in which the flag changed twice, on two tops. In the given example such tops will be tops υ_3 and υ_4. The top υ_3 further gets out and its splitting into two tops υ_3^1 is carried out and υ_3^2. After that transformation of graphs becomes, represented on fig. 3.

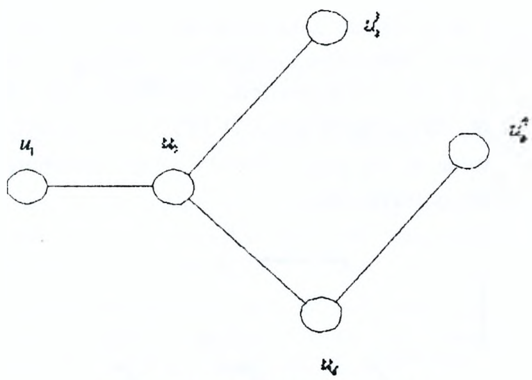

Fig. 3 – Transformed graphs.

After that values of flags of a condition of all tops are nulled, and the algorithm all over repeats again. Work comes to the end, when in the graph will not be found any contour or if the flag of a condition will not change twice at one top. It is obvious, that this algorithm of transformation the graph in a tree will work on any graphs because for anyone the graph containing a contour, removal of this contour by the algorithm described above is possible. And the graphs, which is not containing contours, by definition is a tree.

III. CONSTRUCTION OF MATCHINGS FOR TREES

The matchings for a tree constructs by the following way:
Step 1: Gets out any trailing top of a tree, and an edge incidental to this top appears an edge of matchings.

Step 2: Edges the graph, adjacent chosen are painted over, and further are not examined.

Step 3: All tops of the tree which has been not used earlier, adjacent painted over, appear trailing.

Step 4: Check if there were trailing tops, we come back to a step 1, differently the matchings is constructed.

If to continue examining the example on a step 1 any trailing top gets out, for example, υ_3^2 and the edge by the edge of matchings appears adjacent to it. Further the edge adjacent to it $[\upsilon_2, \upsilon_4]$, is painted over, and further is not examined. Two tops remains. Some of them gets out the edge adjacent to it an edge of matchings, for example, υ_3^1 appears. Further the edge adjacent to it $[\upsilon_1, \upsilon_2]$, is painted over, and further it is not examined. Trailing tops did not remain any more, work of algorithm is completed. The tree will be transformed in represented on fig. 4.

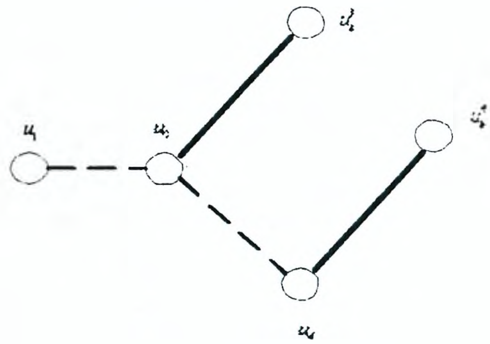

Fig. 4 – The matchings in a tree.

After construction of matchings it is necessary to transform a tree in initial graphs. For this transformation it is prospected in a tree of tops which were tops of splitting of a contour earlier, and these tops incorporate in one (fig. 5).

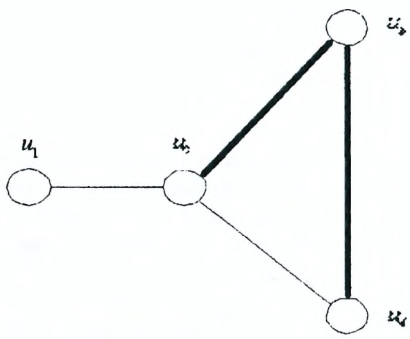

Fig. 5 – Transformation of a tree in initial graphs.

May be such situation at which at association of tops the condition of an accessory of edges the graph to matchings (as well as has taken place in an example) is broken, it means that adjacent edges will concern to the matchings. For elimination of this lack, the algorithm of pushing out is applied. This algorithm means pushing out of an edge of the matchings on other adjacent edges the graph (if they cannot concern to other edges of matchings, they are not painted over), or may be exception of this edge of set of edges of the matchings if pushing out of any edge is impossible.

In the given example the edge $[\upsilon_3,\upsilon_4]$ is examined. Pushing out of it is impossible, as the edge $[\upsilon_2,\upsilon_4]$ is the painted over edge concerning an edge $[\upsilon_2,\upsilon_3]$ which is an edge of the matchings. The edge $[\upsilon_2,\upsilon_3]$ is examined. It can be pushed out on an edge $[\upsilon_1,\upsilon_2]$. If the edge $[\upsilon_1,\upsilon_2]$ did not exist, the edge $[\upsilon_2,\upsilon_3]$ should be removed from set of edges of the matchings. All edges which concern to the matchings are looked through. All of them satisfy to a condition of existence of matchings, work of algorithm is finished. Final graphs is represented on fig. 6.

After association of all broken before tops and check of all edges of the matchings the algorithm comes to the end, matchings is constructed.

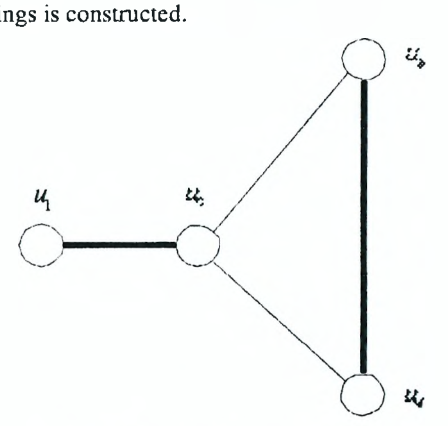

Fig. 6 – The matchings on the graph.

IV. NECESSARY STATEMENTS AND POSSIBLE PROOFS

The important stage in development of algorithm is the proof that given matchings really is maximal.

The tree is represented as set of stars and circuits. The tree consisting of two stars and three additional edges is examined. The matchings will be constructed for it by the method which was described above. It is maximal. We shall assume that the developed algorithm is fair for any tree. For the proof we shall take advantage of a method of a mathematical induction.

The theorem 1.

For any natural n number of stars of a tree and m additional branches of a tree the matchings, constructed by algorithm of movement from periphery to the center is maximal.

The proof.

Direct check of this statement for each value n also m is impossible, as the set is indefinite. For the proof of this statement, its validity is checked all over again for $n=1$ $m=1$. Then it is proved, that at any natural value k, p from validity of the examined statement at $n=k$, $m=p$ its validity and follows at $n=k+1$ $m=p+1$.

Then the statement takes for granted for all n and m. Really, the statement is fair at $n=1$ $m=1$. But then it is fair and for the following number $n=1+1=2$ $m=1+1=2$. From validity of the statement for $n=2$, $m=2$ its validity follows for $n=2+1=3$ $m=2+1=3$. Validity of the statement for $n=4$ $m=4$, etc. from here follows. Clearly, that, eventually, we shall reach any natural number n m. Means, the statement is true for anyone n m.

The assumption of that finds that the developed algorithm maximal matchings for any tree is proved.

V. THE ESTIMATION OF COMPLEXITY OF ALGORITHM

Labour input (complexity) of algorithm of the decision of the given task is the function f putting in conformity to each natural number n an operating time $f(n)$ of algorithm (at worst on inputs of length n). Otherwise, function $f(n)$ is the maximal operating time of algorithm on all inputs of the task of length n.

At a presence of estimations of complexity of algorithms the symbolics O is used.

The algorithm consists of three stages:

1) Transformation the graph in a tree;
2) Construction of matchings for a tree;
3) Restoration initial the graph with application of algorithm of pushing out for preservation maximal matchings.

Labour input of each stage is examined. Transformation the graph in a tree means viewing all tops the graph so much time, how many the closed contours from three tops contain in the graph. The maximum quantity of contours from three tops equally to a number of combinations from $|V|$ on 3, and it is in turn equal $\frac{|V|!}{3!(|V|-3)!}$. If contours does not exist, complexity of this stage will consist in viewing all tops the graph and makes $O(|V|)$.

In case of full the graph, complexity makes $O(|V|*\frac{|V|!}{3!(|V|-3)!})$.

Proceeding them it, labour input of the given stage can change from $O(|V|)$ up to $O(|V|*\frac{|V|!}{3!(|V|-3)!})$.

Construction of matchings on a tree means viewing all edges of a tree. Proceeding from this, labour input of this stage $O(|E|)$.

Restoration initial the graph with application of algorithm of pushing out means viewing $|V|*|V-1|$ tops the graph during their connection and $|E|$ edges the graph for pushing out. Complexity of this stage $O(|E|+|V|*|V-1|)$.

General minimal complexity of algorithm makes: $O(|V|)+O(|E|)+O(|E|+|V|*|V-1|)=O(|V|+$ $+2*|E|+|V|*|V-1|)$. General maximal complexity of algorithm makes:

$$O(|V|*\frac{|V|!}{3!(|V|-3)!})+O(|E|)+O(|E|+|V|*|V-1|)=$$

$$O(|V|*\frac{|V|!}{3!(|V|-3)!}+2*|E|+|V|*|V-1|).$$

The memory size, required by algorithm is equal from V^2 up to $V^2+\left(\frac{|V|!}{3!(|V|-3)!}\right)^2$.

VI. RESULTS OF TESTING

To compare various methods of the decision of some task, it is necessary to lead at the first theoretical comparison, at the second practically to confirm the received results.

Theoretical comparison on volume of used memory and labour input for some algorithms of search maximal matchings is resulted in table 1.

Table 1. Theoretical comparison of methods of search maximal matchings.

The name of a method	Type the graph	Labour input	Used memory
Edmond-Carp's algorithm	double-segment	$O(V*E^2)$	$V^2+\frac{V}{2}$
Kuhn's algorithm	any	$O(V*E^2)$	$\|V\|^2$
Hopcroft-Carp's algorithm	double-segment	$O(\min(\|V\|,\|U\|$ $*\|E\|)$	$V^2+\frac{V}{2}$
Parallel algorithm	any	$O(\frac{V*E^2}{\Theta^3})$	$2*V^2+\frac{V}{2}$
Quantum algorithm	double-segment	$O(\sqrt{V}),O(V)$	$5*V^2$
Transform the graph in a tree	any	$O(\|V\|+2*\|V\|+$ $+\|V\|*\|V-1\|)$, $O(\|V\|*$ $*\frac{\|V\|!}{3!(\|V\|-3)!}+$ $+2*\|E\|+$ $+\|V\|*\|V-1\|)$	V^2, V^2+ $\left(\frac{\|V\|!}{3!(\|V\|-3)!}\right)$

It is visible from the table, that the most toilful are methods for any graphs. Algorithms for double-segmented graphs have approximately identical labour input and quantity of used memory. Realization of algorithms confirms theoretical results.

VII. THE CONCLUSION

As the result of the given work the algorithm of representation the graph as a tree and constructions for it maximal matchings has been developed. The method has included the following algorithms:

- Algorithm of transformation the graph in a tree;
- Algorithm of construction matchings in a tree;
- Algorithm of restoration the graph;
- Algorithm of pushing out.

In result it has been found out, that the developed algorithm of representation the graph as a tree has the following characteristics:

- Is simple in realization and understanding;
- Has comprehensible time of the decision;
- Uses memories no more, than other algorithms.

The developed algorithm of a presence maximal matchings can be applied to the decision of the broad audience of practical tasks, for example, for construction of systems of support of decision-making in uncertain conditions, by development of bases of knowledge of intellectual systems, for the decision of the task of routing of packages in local networks, etc.

The main advantage of algorithm is that it is universal and has comprehensible time of the decision.

Lack of the given algorithm is that fact, that its labour input is influenced by structure the graph. The more contours contains graphs, the above complexity of algorithm. Though at small quantity of contours the algorithm is effective enough.

REFERENCES

[1] Lovas L., Plammer M. *The applied tasks of the theory of graphs. The theory of matching in mathematics, physics, chemistry.* - M.: the World, 1998.

[2] Anderson D.A. *Discrete mathematics and the combination theory*: With English - M.:The publishing house "Williams", 2003.

[3] Cormen K., Laserson Ч., Rivest R. *Algoritm of the construction and analysis.* - M.: 2000.

[4] Svamy M., Thulasiraman K. *Graphs, networks and algorithms.* - M.: the World, 1984.

[5] M.O.Asanov *"Discrete optimization "* - Ekaterinburg: Ural SCIENCE, 1998.

[6] Kureitchic V.M. *The new approach to the decision of graph's tasks on the basis of quantum algorithms. //* Perspective information technologies and intellectual Systems. № 2 (18), 2004, - C. 14-16.

[7] Shungarov H.D. *The parallel algorithm of a presence of maximal matchings in the graph.* Electronic magazine "It is investigated in Russia "http:// zhurnal.ape.relarn.ru/articles/2003/107.pdf.

[8] Romanovsky I.V. *The discrete analysis.* BHV- Sent Petersburg, 2003. - 320 p.