

## **SIMULATION OF BEHAVIOR AND INTELLIGENCE**

### **Extremal Subsystems of Monotonic Systems. III**

J. E. Mullet

Department of Economics, Tallinn Technical University (1973 – 1980)  
Translated from *Avtomatica i Telemekhanika*, No. 1, pp. 109 – 119, January, 1977.  
Original article submitted February 23, 1976.

© 1977 Plenum Publishing Corporation, 227 West 17<sup>th</sup> Street, New York, 10011.

We alert the readers' obligation with respect to copyrighted material.

**Correspondence Address:** Byvej 269  
2650 Hvidovre  
Copenhagen Denmark  
Phone (45) 36 49 33 05  
E-mail: [joseph.mullet@email-telia.dk](mailto:joseph.mullet@email-telia.dk)

### **Abstract**

An attempt is made to find parts of a given graph that are more “saturated” than any other part parts with “small” graphs of the same type. On the basis of such a formulation, constructing a monotonic system from structural elements of graphs (arcs or vertices) solves this problem. The scheme of producing a monotonic system from a given graph is presented in general form, and the necessary constructions are illustrated by examples. This paper is a continuation of [1] and [2]; it has the purpose of illustrating the procedures (developed in the first two parts) of finding extremal subsystems for solving certain problems arising in tournaments, a-cyclic graphs, and undirected and directed trees.

Keywords: monotonic, system, matrix, graph, cluster

АКАДЕМИЯ НАУК СССР

АВТОМАТИКА  
И  
ТЕЛЕМЕХАНИКА

1

---

МОСКВА · 1977

**ЭКСТРЕМАЛЬНЫЕ ПОДСИСТЕМЫ МОНОТОННЫХ СИСТЕМ. III**

И. Э. МУЛЛАТ

(Галлин)

Рассматривается возможная постановка задачи выделения частей из заданного графа, более «насыщенных», чем какие-либо другие части, однотипными «малыми» графами. Решение этой задачи, исходя из предложенной постановки, осуществляется путем образования монотонной системы на структурных элементах графов (дугах или вершинах). Схема образования монотонной системы из заданного графа приводится в общем виде, и необходимые конструкции поясняются на примерах.

Работа является продолжением [1, 2] и ориентирована на иллюстрацию развитого там аппарата выделения экстремальных подсистем для решения некоторых задач, возникающих в турнирах, ациклических графах, неориентированных и ориентированных деревьях.

К объектам, к которым в настоящее время проявляется интерес исследователей сложных систем, относятся графы [3]. С одной стороны, граф — математический объект, а с другой — удобное средство описания и анализа взаимозависимостей между элементами в системе. В случае систем с небольшим числом элементов анализ графов не представляет никаких трудностей, но, когда число элементов велико, возникают проблемы.

В данной работе предлагается анализ графа заменить последовательным анализом выделяемых из графа частей. В теории графов имеется богатый аппарат выделения подграфов, частей и т. п., однако при анализе больших графов не всегда удается классические методы связать с конкретными нуждами исследователя. Известно, например, что экспериментальные графы довольно пусты и поэтому содержат много максимальных полных подграфов, которые выделять в отдельности нет смысла.

С нашей точки зрения, пригодный аппарат выделения частей в графе можно получить, если воспользоваться понятием монотонной системы [4]. Дело в том, что из графа можно образовать не одну, а множество монотонных систем и соответственно предложить не единственное решение поставленной задачи, а целое множество (даже бесконечное множество) решений. Исследователю графа следует на основе собственной интуиции выбрать допустимый класс решений и только затем уже воспользоваться разрабатываемым здесь формальным аппаратом.

В разделе 4 приводятся некоторые рекомендации о том, как следует выбирать классы решений в конкретных случаях на примере турниров, ациклических графов, возникающих при использовании техники модульного программирования, деревьев. В остальных разделах строится общая модель необходимого аппарата выделения частей, которая иллюстрируется примерами. Терминология теории графов заимствована из монографий [4–6].

**1. Содержательная постановка задачи выделения экстремальных подсистем — ядер на графах**

Рассмотрим на графах задачу такого типа: задан «большой» граф  $G$  и «малый» граф  $G'$ . Необходимо в графе  $G$  выделить такую его часть (множество дуг или ребер), чтобы эта выделенная часть была «насыщена»

## Introduction

Among the items that are at present of interest to investigators of complex systems, let us mention graphs [3]. On the other hand a graph is a mathematical object, and on the other hand it is a conventional means for describing and analyzing the relationship between the elements of a system. In the case of systems with small number of elements, the analysis of graphs does not present any difficulties, but in case of a large number of elements we have problems.

In this paper it is proposed to replace the analysis of a graph by a successive analysis of parts of this graph. In graph theory there exist many methods of selecting of subgraphs, parts, etc.; however, in the analysis of large graphs it is not always possible to adapt the conventional methods to the actual requirements of the investigator. For example, it is well known that experimental graphs are fairly empty, and therefore contain many maximal complete subgraphs whose individual selection makes no sense.

From our point of view it is convenient to select the parts of a graph by a method based on the concept of monotonic system [1]. As a matter of fact, from a graph it is possible to construct not one, but a whole set of monotonic systems. The investigator of a graph must select on the basis of its own intuition an admissible class of solutions, and only after that will he be able to use the formal method developed here.

In Section 4 we give some recommendations how to select the classes of solutions in actual cases, by using the example of tournaments and a-cyclic graphs that occur in the technique of modular programming, as well as trees. The other sections deal with the construction of a general model of the required procedure of selection of parts that is illustrated by examples. The terminology of graph theory has been adopted from [4 – 6].

## 1. Formulation of Problem of Selection of Extremal Subsystems (kernels in graphs)

Let us consider the following problem on graphs. We are given a “large” graph  $G$  and a “small” graph  $g$ . From the graph  $G$  it is required to select a part (i.e., a set of arcs or edges) in such a way that this part is “saturated” with small graphs  $g$ . The saturation part of a graph with small graphs  $g$  can have different interpretations. For example, it can be assumed that one part of a graph is more saturated than another part if the first part contains a large number of graphs  $g$  as compared to the second. The definition of saturation can be also obtained in the following “complex” manner. Let us consider a set of arcs or vertices of a graph  $G$  that occur only in the part of interest to us. That then we can calculate not the total number of small graphs  $g$  located there, but only the “individual” graphs located “near” each arc or vertex. The individual number of small graphs  $g$  located near an arc or vertex is defined as a number of such graphs containing this vertex or arc; hence this number is expressed by an integer. By proceeding in this way, we obtain precisely as many integers specifying the part of interest to us, as there are arcs or vertices in it, and each integer represents a “local” saturation of the graph  $G$  by small graphs  $g$ .

On the basis of these integers there can be many ways of defining the saturation of part of a graph. It is possible to calculate their mean value, their variance, etc. Here we shall consider the simplest characteristic, namely the least of all the local numbers of small graphs  $g$  located in the selected part of a large graph  $G$ . Figuratively speaking we can say that this is the number of subgraphs of  $G$  in the “emptiest” place.

Below we present an exact formulation of the problem of determination of the parts of a large graph  $G$  that have greatest saturation with small graphs  $g$ . This problem can be formulated as follows: among all possible parts of a graph  $G$  (or among the largest number of such parts), find the part in which the least of all the local numbers of a small graphs  $g$  that are entirely contained in it is maximal.

It is natural to expect that in the thus-obtained part it is possible to locate in the usual manner a large number of small graphs  $g$ . Indeed, at each vertex or arc the number of small subgraphs  $g$  is not less than at the vertex or arc at which this number is minimal. On the other hand in an external part this minimal number is nevertheless sufficiently large; we especially selected this part in such a way that the condition of global maximum of the minimum local number of graphs  $g$  is satisfied.

In the same way it is possible to formulate the problem of determination of the least saturated part of a graph  $G$  by small graphs  $g$ . In this case each part will be characterized by a number of subgraphs of  $g$  at the vertex or arc at which this number is maximal. Instead of seeking the graph part in which the minimal local number of graphs  $g$  is maximal, we seek on the contrary the part in which the maximal local number is minimal. In this case the number of subgraphs of  $g$  at each vertex or arc will not be larger than their number at the “maximal” vertex or arc, this number being small by virtue of the condition of global minimum.

Let us note yet another advantage of the above-defined external parts of graphs. As a rule, the saturation or non-saturation of these parts by small graphs is “uniform.” Usually a saturated extremal part cannot have an especially least number of graphs  $g$  at any vertex or arc, since the part of the graph  $G$  without this vertex or arc is apparently more saturated with subgraphs of  $g$  in the above-mentioned “complex” sense. Conversely, for the same reason an unsaturated extremal part cannot have an especially large number of subgraphs of  $g$  at any one arc or vertex.

The procedure of selection of parts of graph developed in this paper is based on the concept of a monotonic system. In considering actual applications of this technique, we must be able to calculate the number of distinct subgraphs of  $g$  located at any given part of the large graph  $G$ . This is not a simple problem, but many investigators dealing with the theory of graphs have considered the calculation of distinct parts of a graph, such as Euler circuits, regular trees [5], simple chains (paths) [7], and simple circuits [8]. Hence we possess a highly developed technique of calculation that can be used for finding the extremal parts of graphs as defined above.

Among the meaningful problems that can be solved with the aid of the method developed here, let us note the problem of selection, from family of  $n$  object that have to be ordered, of the most unordered (unmatched), or of the most (ordered) (matched) sets of objects. As a matter of fact, in the same way as in [9] we can take as a measure of compatibility the number of transitive triples, and as a matter of incompatibility the number of cyclic triples. In our terminology, cyclic and transitive triples are certain small graphs.

Such a development of monotonic systems on graphs can be used, for example, in finding the “bottlenecks” of operational systems described in the language of modules [10]. In such large systems it is not so easy to orient oneself in the hierarchy of mutually generating modules, and to understand the principal manners of construction of working programs.

## 2. General Model of Finding Kernels on Graphs

For a given graph  $G$  let us denote by  $V(G)$  or  $V$  the set of its vertices. The set of arcs of a directed graph  $G$  will be denoted by  $U(G)$  or  $U$ , and the set of edges of an undirected graph will be denoted by  $E(G)$  or  $E$ .

In graph theory we use the concept of a subgraph of a graph  $G$ . A graph  $G'$  is a subgraph of a graph  $G = [V(G), U(G)]$  if  $V(G') \subset V(G)$  and  $U(G')$  is the set of those and only those arcs of  $G$  that connect pairs of vertices belonging to  $V(G')$ . The definition of a set of subgraphs of an undirected graph has the same form. Instead of an arc, we must consider in this case an edge of  $G$ . Sometimes one uses the concept of part of a graph  $G$ . A part  $G''$  of a graph  $G = [V(G), U(G)]$  is a graph such that  $V(G'') \subseteq V(G)$  and  $U(G'') \subseteq U(G)$ . In  $G''$  some of the arcs of the graph  $G$  are simply absent. In the same way we can define a part of an undirected graph  $G = [V(G), E(G)]$ . Let us note that one of the most important concepts in this paper is the isomorphism of graphs [6].

The construction described in [1] begins with the specification of the elements of a system  $W$ . In graphs there exists two structural units – vertices and arcs. First of all let us consider the case that as an element of the system  $W$  we take a vertex of a graph  $G$ .

In accordance with the construction proposed in [1] it is necessary to define the concept of  $\oplus$  and  $\ominus$  actions over vertices (elements) of a system. The definition of  $\oplus$  action and  $\ominus$  action requires the assignment of special significance function  $\pi$  of the vertices of  $G$ . As a result of  $\oplus$  actions the significance of vertex in a system must increase, whereas the  $\ominus$  actions decreases the significance.

The construction carried out in [1] requires numerical arrays (weights) on each subset  $H$  of elements of the system  $W$ . In [1] we have shown that for this purpose we need an initial weight array on  $W$  and a method of realization of  $\oplus$  and  $\ominus$  actions. The initial weight array  $\{\pi(\alpha) \mid \alpha \in V\}$  can be defined, for example, as follows. In addition to a “large” graph  $G$ , let us consider also a “small” graph  $g$ . Let us calculate the number of distinct subgraphs of  $G$  that are isomorphic to a graph  $g$  whose set of vertices contains the vertex  $\alpha$ . Let us take this integer as the initial significance level  $\pi(\alpha)$ . For emphasizing the dependence of the just-introduced level  $\pi(\alpha)$  of “small” graphs, we shall also use the expression “the weight  $\pi(\alpha)$  of a vertex  $\alpha$  in the graph  $G$  with respect to  $g$ .”

Below we present two operations of generation new graphs from a graph  $G$ ; they are denoted by  $\oplus$  and  $\ominus$ . Let us consider a graph  $G$  and let  $\Lambda$  be an empty graph, i.e., a graph that does not contain any arc, but which has  $|V(G)|^1$  vertices. It is assumed that  $V(\Lambda)$  is an exact copy of  $V(G)$ , and in referring to a vertex  $\alpha$  we have in mind a vertex of graph  $G$ , through it apparently can be of two sorts, namely as a vertex of  $G$  and as a vertex of  $\Lambda$ .

An operation of type  $\ominus$  with a vertex  $\alpha$  in the graph  $G$  consists of removal of all the arcs leading to a vertex  $\alpha$  of  $G$ .

An operation of type  $\oplus$  with a vertex  $\alpha$  in the graph  $G$  consists of restoring on an empty graph all the arcs leading to a vertex  $\alpha$  of  $G$ .

---

<sup>1</sup>  $|M|$  is the number of elements of the set  $M$

It is easy to see that as a result of the  $\ominus$  operation on any vertex  $\alpha$ , the weights of all the other vertices with respect to a selected small graph  $g$  are either decreasing, or the at least remain at the previous level. In realizing the  $\oplus$  operation, there naturally arises the question of what can be regarded as a weight of vertex after its realization.

This problem can be solved by the following construction. On the graph  $\Lambda$  we calculate the proper weights of the vertices with respect to a small graph  $g$  and we add them together with the weights on the vertices of  $G$ . The thus-obtained sum is taken as a total weight of the vertices. In this case we can observe the opposite effect; i.e., as a result of  $\oplus$  operation the total weights either increase or they remain (as in case of  $\ominus$  weights) at the previous level. In general the initial array of weights  $\{\pi(\alpha) \mid \alpha \in V\}$  (i.e., the array of weights prior to any operations) on the vertices of the graph  $G$  can be taken as a total array of weights, since the contribution of the graph  $\Lambda$  is zero. Below we shall consider only the total weights  $\pi(\alpha)$  of graph vertices that are called weights in the above sense.

Summing up, we can say that a  $\ominus$  operation is equivalent to defining a  $\ominus$  action on elements of the system  $W$ , whereas  $\oplus$  operation is equivalent to  $\oplus$  action if we take the above-defined total weights as significance levels of the vertices of the graph  $G$ . Thus the monotonicity inequalities are satisfied in the above scheme, this being the principal property of monotonic systems [1].

In constructing the sets of weight arrays of the system  $W$  it is necessary to indicate in which manner the above-calculated array of initial weights  $\{\pi(\alpha) \mid \alpha \in V\}$  is redistributed as a result of  $\oplus$  and  $\ominus$  actions.

Suppose we have specified a sequence of vertices  $\alpha_1, \alpha_2, \alpha_3, \dots$  forming the set  $\bar{H} \subset V^2$ . Let us successively perform  $\oplus$  actions on the vertices of the graph  $G$  in accordance with this sequence. As a result we obtain on the set  $V(\Lambda)$  a part of the graph  $G$ . At each vertex belonging to  $V(\Lambda)$  in this part it is possible to calculate the number of subgraphs of the part that are isomorphic to a small graph  $g$ , and obtain the weights on the elements of the set  $H$ .

---

<sup>2</sup> In contrast to the general model described in [1], we do not allow here the repetition of elements  $\alpha_i$ . The set  $\bar{H}$  is the complement of  $H$ .

Following the notation used in [1], we can write that a new significance function has been defined on  $H$  that has the form

$$\pi_{\alpha_1}^+ \pi_{\alpha_2}^+ \pi_{\alpha_3}^+ \dots \quad (1)$$

and which has been constructed from the initial array of weights  $\{\pi(\alpha) \mid \alpha \in V\}$ .

Thus by specifying a sequence of vertices  $\langle \alpha_1, \alpha_2, \dots \rangle$  forming the set  $\bar{H}$ , we obtain on  $H$  a weight array specified by the function (1). This array denoted by  $\Pi^+ H$  and called a weight array on the set of vertices  $H$ . The weight arrays form a collection of weight arrays  $\{\Pi^+ H \mid H \subseteq V\}$ . Sometimes it is convenient to use the expression “collection of  $\oplus$  arrays with respect to a small graph  $g$ .”

The collection of weight arrays  $\{\Pi^+ H \mid H \subseteq V\}$  can be defined in a similar way. As above, the array of weights  $\Pi^- H$  is defined by the function

$$\pi_{\alpha_1}^- \pi_{\alpha_2}^- \pi_{\alpha_3}^- \dots \quad (2)$$

and specified on the part of the graph  $G$  left over after applying a sequence of  $\ominus$  actions to the vertices  $\alpha_1, \alpha_2, \alpha_3, \dots$ , forming the set  $\bar{H}$ . Let us only note that the array of weights on each subset  $H \subseteq V$  is actually a proper array of the remaining part, and not the total array, since in this case the contribution yielded by the graph  $\Lambda$  is equal to zero.

Let us continue the construction of the procedure (needed below) of finding extremal subsystems (kernels). In contrast to the foregoing, we shall take an arc as an element of the system. The system  $W$  will be defined as an interrelated set of arcs  $U(G)$  of the graph  $G$ . Following [1], it is necessary to specify  $\oplus$  and  $\ominus$  actions on the arcs of the graph  $G$ ; as in the case of a system of vertices, this requires the determination of the initial significance function  $\pi$  of arcs in the graph  $G$ .

Let us consider a small graph  $g$ . We shall calculate the number of distinct subgraphs of the graph  $G$  that are isomorphic to a graph  $g$  whose set of arcs contains the arc  $\alpha$ . This integer is taken as the initial significance level  $\pi(\alpha)$  of the arc  $\alpha$  in the graph  $G$ , and it is called the weight of the arc  $\alpha$  with respect to the graph  $g$ .

The concept of  $\oplus$  and  $\ominus$  actions on arcs of the graph  $G$  can be defined constructively and exactly according to scheme similar to the one used for the vertices of the graph  $G$ .

Let us consider a graph  $G$  and let  $\Lambda$  be an empty graph with  $|V(G)|$  vertices. We shall assume that the set of vertices  $V(\Lambda)$  is an exact copy of  $V(G)$ .

An operation of type  $\ominus$  on an arc  $\alpha$  of a graph is called an operation of removal of this arc on the graph  $G$ .

An operation of type  $\oplus$  on an arc  $\alpha$  is called an operation of restoration of this arc on an empty graph  $\Lambda$ .

At the first let us consider the  $\ominus$  operation. It is evident that as a result of removing the arc  $\alpha$ , the initial array of weights with respect to a small graph  $g$  can either decrease or remain at the previous level. A decrease in significance (weights) proves that the  $\ominus$  operation is equivalent to the definition of a  $\ominus$  action on an element of the system  $W$ .

Let us specify a sequence  $\alpha_1, \alpha_2, \alpha_3, \dots$  of distinct arcs of the graph  $G$  that form a set  $\bar{H} \subseteq U(G)$ . Let us perform  $\ominus$  actions on the arcs of the graph  $G$  in accordance with this sequence. As a result, a certain part of the graph  $G$  is left over on the set of vertices  $V(G)$ ; the elements of this part are the arcs of the set  $H$ ,  $H \subseteq U(G)$ . For each arc  $\alpha \in H$  let us calculate the number of subgraphs that are isomorphic to  $g$ ; this number is assumed to be the value of the weight of the element  $\alpha$  with respect to the set  $H$ . In accordance with our notations, this method of determination of weights specifies a function  $\pi_{\alpha_1}^- \pi_{\alpha_2}^- \pi_{\alpha_3}^- \dots$  on the elements (arcs) of the set  $H$ .

Thus, just as in case of assignment of collections of weight arrays on vertices of a graph, we obtain on the arcs belonging to  $H \subseteq U(G)$  a weight array  $\{\pi^- H \mid H \subseteq U(G)\}$  on the arcs of the graph  $G$ . We shall use also the expression “ $\ominus$  collection of weight arrays of  $\ominus$  actions on arcs with respect to a small graph  $g$ .”

The determination of  $\oplus$  actions on the basis of  $\oplus$  operations over an empty graph  $\Lambda$  requires a more detailed analysis. Suppose we have again specified a sequence  $\alpha_1, \alpha_2, \alpha_3, \dots$  of arcs of the graph  $G$  that form a set  $\bar{H}$ . Let us successively perform  $\oplus$  operations on arcs of the set  $\bar{H}$ . As a result we obtain on the set of vertices  $V(\Lambda)$  a part of the graph  $G$  with a string of arcs equal to  $\bar{H}$ . Previously we calculated with the aid of a model at the vertices the total weight of each vertex  $\alpha \in V(G)$ . In the present case we try to proceed in the same way and calculate the total weight of the arcs forming the set  $H$ . The arcs of the set  $H$  are not drawn on the graph  $\Lambda$ , and there naturally arises the question of how to calculate the number of subgraphs that are isomorphic to a graph  $g$  and that contain an arc  $\alpha$ , which is absent on the graph  $\Lambda$ . We shall proceed as follows: we shall assume that this arc has been fictitiously drawn only at the instant of calculation of subgraphs. Thus we obtain on the set of arcs  $H$  certain integers that depend both on the graph  $G$  and on the part of the graph  $G$  that appears on the empty graph  $\Lambda$ . These numbers are the sum of two arrays of numbers, i.e., of the initial array of weights on the arcs of the graph  $G$  with respect to the small graph  $g$ , and the array of weights with respect to this same graph  $g$ , but calculated only on the just-mentioned part.

In the manner described above we determine on the set  $H$  a function  $\pi_{\alpha_1}^+ \pi_{\alpha_2}^+ \pi_{\alpha_3}^+ \dots$  that specifies a weight  $\oplus$  array  $\Pi^+ = \{\pi^+ H(\alpha) \mid \alpha \in H\}$ . Thus also in case of  $\oplus$  operations we can determine a collection of weights arrays of  $\oplus$  actions with respect to a small graph  $g$ . It is justified to use the expression “ $\oplus$  action,” since the total weights of elements not yet subjected to  $\oplus$  action can either increase or remain at the same level.

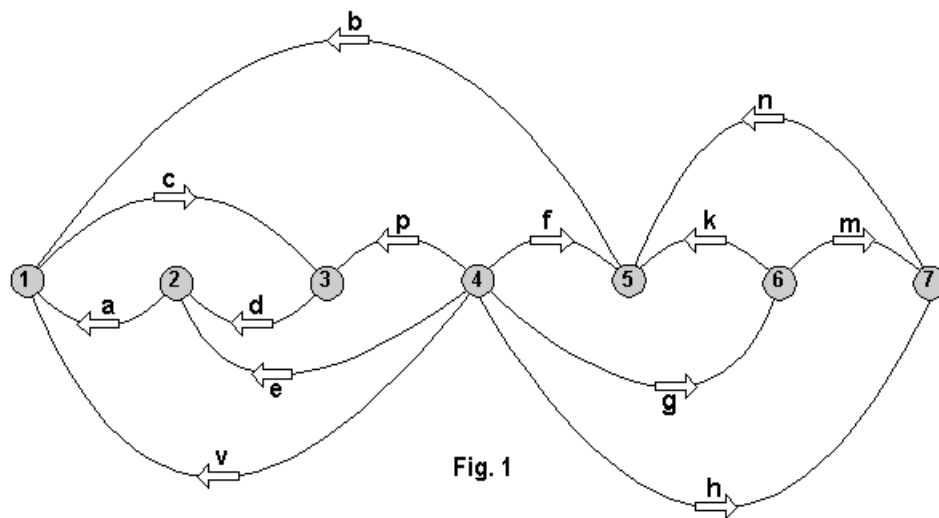
### 3. Illustrative Examples on Directed Graphs

A graph  $G$  of partial ordering is defined as a binary relation  $G$  with the following properties:

- a) reflexivity, i.e., if  $\alpha \in V(G)$ , then  $\alpha G \alpha$ . The graph  $G$  has a loop at the vertex  $\alpha$ .
- b) transitivity; if there exists an arc  $(\alpha, \beta)$  and  $(\beta, \gamma)$ , then the graph  $G$  has an arc  $(\alpha, \gamma)$ , or from  $\alpha G \beta$  and  $\beta G \gamma$  it follows that  $\alpha G \gamma$ .

A complete order is defined as a graph of partial ordering in which any pair of vertices  $\alpha$  and  $\beta$  is connected by an arc.

It is possible to formulate the following problem: in a given directed graph it is required to find the (in certain sense) most “saturated” regions that are “close” to a graph of partial ordering or to graphs of complete ordering. This problem will be solved by a method of organization (on a graph) of a monotonic system with subsequent determination of kernels.

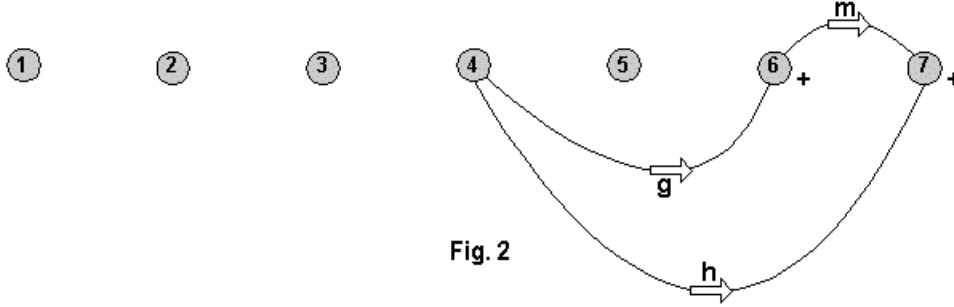


In accordance with the scheme of organization of a monotonic system on graphs described in the previous section, it is necessary to assign a small graph  $g$ . Suppose that this graph consists of three vertices  $x, y, z$ , and it is such that  $U(g) = \{(x, y), (y, z), (x, z)\}$ . The graph has a total of three arcs (a transitive triple).

Now let us consider the assignment of collection of weights arrays at the vertices of a graph shown in Fig.1. The loops on this graph have been omitted.

According to the scheme of assignment of collections of weight arrays at the vertices of a graph, it is required to determine an initial array of weights  $\{\pi(\alpha)\}$ , where  $\alpha = 1, 2, 3, \dots, 7$ . According to the method of calculation of the values  $\pi(\alpha)$  with respect to the graph  $g$  (a transitive triple), we obtain  $\pi(1) = 3$ ,  $\pi(2) = 2$ ,  $\pi(3) = 2$ ,  $\pi(4) = 7$ ,  $\pi(5) = 4$ ,  $\pi(6) = 3$ ,  $\pi(7) = 3$ . As an example, let us determine a weight array on a subset of vertices  $H = \{1, 2, 3, 4, 5\}$ . By successively performing  $\ominus$  actions on the set  $\bar{H} = \{6, 7\}$ , we obtain on the set  $H$  a new weight array  $\pi(1) = 3, \pi(2) = 2, \pi(3) = 2, \pi(4) = 4, \pi(5) = 1$ .

The values of the function  $\pi_6^+ \pi_7^+$  can be obtained in a similar way, but for this purpose it is necessary to use the assignment of collections of total  $\oplus$  arrays with respect to a transitive triple. According to Fig.2, the values of this function in their order at the vertices  $\{1,2,3,4,5\}$  are as follows:  $\pi(1) = 3, \pi(2) = 2, \pi(3) = 2, \pi(4) = 8, \pi(5) = 4$ . In exactly the same way we can determine on any subset  $H$  of vertices  $V = \{1,2,3,4,5,6,7\}$  a proper weight array of  $\oplus$  or  $\ominus$  actions with respect to a transitive triple.



Now let us consider a construction that is assigned not on vertices, but on the arcs of the graph presented on Fig.1. In this case the set of elements of the system  $W$  will be  $U(G) = \{a, b, c, \dots, n, m\}$ . As the small graph  $g$  we shall take the same graph as above, with a set  $U(g) = \{(x, y), (y, z), (x, z)\}$ .

By analogy with the foregoing, we realize the construction in the same succession. We determine an initial weight array  $\{\pi(\alpha) \mid \alpha \in U\}$  on the arcs of the graph  $G$  in accordance with the general scheme. We find that

$$\begin{aligned} \pi(a) = 1, \pi(b) = 1, \pi(c) = 1, \pi(d) = 1, \pi(e) = 2, \pi(f) = 3, \\ \pi(g) = 2, \pi(h) = 2, \pi(k) = 2, \pi(n) = 2, \pi(m) = 1, \pi(v) = 3, \pi(p) = 2. \end{aligned}$$

As an example, let us now perform  $\ominus$  actions on the arcs  $f, k$  and  $m$ , i.e., on the set  $H = \{f, k, m\}$ . On the set  $H$  we hence obtain

$$\begin{aligned} \pi(a) = 1, \pi(b) = 0, \pi(c) = 1, \pi(d) = 1, \pi(e) = 2, \\ \pi(g) = 0, \pi(h) = 0, \pi(n) = 0, \pi(v) = 2, \pi(p) = 2. \end{aligned}$$

In accordance with the adopted system of notations this array of numbers will be denoted by  $\Pi^- H$ . For obtaining a  $\Pi^+ H$  array, we must calculate the total weights. The dashed lines in Fig.3 represent the arcs of graph  $\Lambda$  that experience the effect of  $\oplus$  actions performed on the arcs  $f, k$  and  $m$ .

According to Fig.3, the total weight array will be as follows:

$$\begin{aligned} \pi(a) = 1, \pi(b) = 1, \pi(c) = 1, \pi(d) = 1, \pi(e) = 2, \\ \pi(g) = 3, \pi(h) = 2, \pi(n) = 3, \pi(v) = 2, \pi(p) = 2. \end{aligned}$$

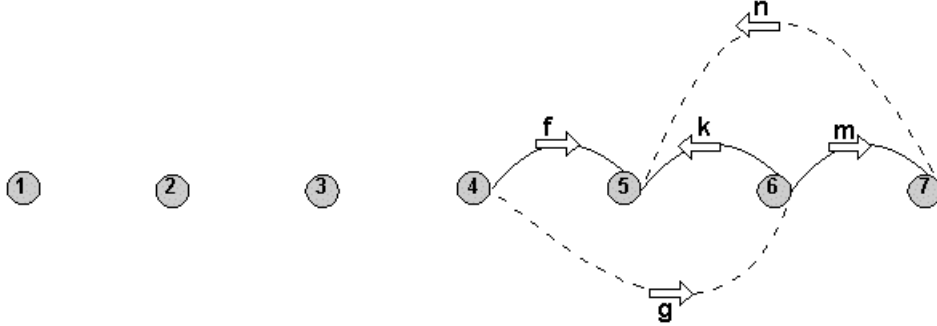


Fig. 3

Thus on any subset  $H$  of arcs of the graph shown in Fig.1 we can construct the weight arrays  $\Pi^-H$  and  $\Pi^+H$ .

Next we describe the procedures of construction of determining sequences of  $\oplus$  or  $\ominus$  actions, at first for vertices, and then for arcs of the graph shown in Fig.1. The construction is carried out for the purpose of illustrating the concepts of  $\oplus$  or  $\ominus$  kernels of the monotonic system [1], and also for ascertaining the effect of the duality theorem formulated in [2].

Let us consider an example in which  $\ominus$  weight arrays are assigned at vertices with respect to a transitive triple. According to the scheme prescribed in [2], the procedure of construction of a determining  $\ominus$  sequence of vertices of a graph on the basis of  $\ominus$  actions (the kernel-finding procedure KFP) consists of two steps (the zero-th and the first step) for the graph shown in Fig.1; it yields two subsets  $\Gamma_0^-, \Gamma_1^- \subseteq V(G)$ , where

$$\Gamma_0^- = V(G) = \{1,2,3,\dots,7\}, \Gamma_1^- = \{4,5,6,7\}, \text{ and the thresholds } u_0 = 2, u_1 = 3.$$

The determining sequence of vertices constructed with the aid of  $\ominus$  actions is as follows:  $\bar{\alpha}_- = \langle 3,2,1,4,5,6,7 \rangle$ . Thus on the basis of Theorems 1 and 3 of [1], and of Theorem 1 on KFP in [2], it can be asserted that the set  $\{4,5,6,7\}$  is a definable set of vertices of the graph shown in Fig.1, and hence this set is the largest  $K^\ominus$  kernel.

Now let apply the KFP for constructing a  $\oplus$ -determining sequence. We find that  $\bar{\alpha}_+ = \{4,5,6,7,1,2,3\}$ . The procedure terminates at the third step, and it consists of four steps, namely the zero-th, the first, the second and the third. According to the construction of  $\oplus$  sequences prescribed in the KFP, we produce the sets  $\Gamma_j^+$ :  $\Gamma_0^+ = \{4,5,6,7,1,2,3\}$ ,  $\Gamma_1^+ = \{5,6,7,1,2,3\}$ ,  $\Gamma_2^+ = \{6,7,1,2,3\}$ ,  $\Gamma_3^+ = \{2,3\}$  and a sequence of thresholds  $u_0 = 7$ ,  $u_1 = 4$ ,  $u_2 = 3$ ,  $u_3 = 2$ . As in the case of a  $\ominus$  sequence, we conclude on the basis of Theorems 2 and 3 of [1], and of Theorem 1 of [2], that  $\{2,3\}$  is the largest  $K^\oplus$  kernel of the system of vertices of the graph in Fig.1.

A careful analysis of Fig.1 shows that the  $K^\ominus$  kernel is in fact completely ordered, i.e.,  $\langle 4,5,6,7 \rangle$ . On the other hand the  $K^\oplus$  indicates from the point of view of the “structure” of a graph the region in which the vertices are least ordered. This is in agreement with the our formulation of the problem of finding kernels as representatives of “saturated” or “unsaturated” regions (parts of a graph) with small graphs  $g$ .

Now let us use the KFP for constructing determining sequences of arcs of the graph in Fig.1. The graph has a total of 13 arcs. After applying the KFP, we obtain on the basis of  $\ominus$  actions the following sequence:

$$\bar{\alpha}_- = \langle a,b,c,d,v,e,p,f,k,n,m,h,g \rangle.$$

The procedure terminates at first step and it consists of two steps, namely the zero-th step and the first step. At the zero-th step we have  $\Gamma_0^- = U(G)$ , and at the first step we have  $\Gamma_1^- = \{f,k,n,m,h,g\}$ , with the thresholds  $u_0 = 1$  and  $u_1 = 2$  respectively. Summing up, we can assert on the basis of the results of [1] and [2] that this is a definable set and at the same time the largest  $K^\ominus$  kernel in the system of arcs.

From the point of view of the graph structure, the application of the KFP to arcs in the construction of a  $\ominus$ -determining sequence does not yield anything new compared to the application of the KFP to vertices. We obtain the same complete order  $\langle 4,5,6,7 \rangle$  represented in the form of a string of arcs, and it also corroborates our assertions concerning the saturation of a  $K^\ominus$  kernel by transitive triples. On the other hand the use of KFP for constructing  $\oplus$ -determining sequence of arcs yields a  $K^\oplus$  kernel

$$\Gamma_1^+ = \{k,m,n,g,h,e,p,b,a,c,d\},$$

whose meaning with regard to “non-saturation” with transitive triples cannot be ascertained.

Below we shall illustrate the peculiar features of using the duality theorem from [2] for finding  $K^\ominus$  and  $K^\oplus$  kernels of a monotonic system specified by vertices or arcs of a directed graph.

At first let us consider the monotonic system of vertices of the graph in Fig.1. The sequence of sets  $\langle \Gamma_j^+ \rangle$  specified by the KFP on the basis of  $\oplus$  actions uniquely determines the sets  $V \setminus \Gamma_1^+ = \{4\}$ ,  $V \setminus \Gamma_2^+ = \{4,5\}$ ,  $V \setminus \Gamma_3^+ = \{1,4,5,6,7\}$ . Above we have found that  $F_+(\Gamma_2^+) = u_2 = 3$ . From the construction of a determining sequence  $\bar{\alpha}_-$  of vertices of a graph we know that  $F_-\{4,5,6,7\} = 3$ . Hence by virtue of Corollary 1 of Theorem 1 of [2] we can assert already after the second step of construction of an  $\bar{\alpha}_+$  sequence that the set  $\{1,4,5,6,7\}$  contains the largest  $K^\ominus$  kernel. Thus we have shown that the sufficient conditions of the duality theorem of [2] are satisfied in the example of the graph represented in Fig.1.

Now let us consider the set  $V \setminus \Gamma_1^- = \{1,2,3\}$ . As was shown above, inside this set there exists a set  $\Gamma_3^+ = \{2,3\}$  such that  $F_+(\Gamma_3^+) = 2$ . On the other hand,  $F_-(\Gamma_1^-) = 3$ . By virtue of Corollary 4 of the duality theorem we can assert that set  $\{1,2,3\}$  contains the largest  $K^\oplus$  kernel of the system of vertices of the graph (Fig.1); this likewise confirms that existence of the conditions governing the theorem.

At last let us consider a collection of weight arrays on the arcs of the graph. The determining  $\bar{\alpha}_+$  sequence of arcs specifies a set  $\Gamma_1^+ = \{k,m,n,g,h,e,p,b,a,c,d\}$ . It is easy to see that inside the set  $U \setminus \Gamma_1^+$  there does not exist a set  $H$  as required by the conditions of Corollaries 1 and 2 of the duality theorem in [2]. This shows that in comparison to arrays on vertices, weight arrays on arcs do not satisfy the duality theorem.

#### 4. Methods of Constructing of Monotonic Systems on a Special Classes of Graphs

In contrast to the previous section, we do not carry out here a detailed construction of collections of weight arrays and determining sequences and kernels on any illustrative example. Here we shall show how to select a small graph  $g$  and  $\oplus$  and  $\ominus$  actions so as to match the selection of these elements with the desired “saturation” of the investigated graph. The desired saturation of a graph can be understood as the saturation desirable for the investigator who usually has a working hypothesis with respect to the graph structure. In view of this, we shall consider the following classes of graphs: tournaments, a-cyclic (directed) graphs, and (directed or undirected) trees.

Let us recall the definitions of these classes of graphs. A tournament is a directed graph in which each pair of vertices  $(x, y)$  is connected by an arc [6]. An acyclic graph is a graph without cycles (in case of an undirected graph), and a graph without circuits (in case of a directed graph). Acyclic undirected graphs are trees, and we shall consider the most general class of trees, as well as the class of directed trees.

In tournaments it is appropriate to consider regions of vertices that are “saturated” with cyclic triples. A cyclic triple is a graph  $g$  such that  $V(g) = \{x, y, z\}$ ,  $U(g) = \{(x, y), (y, z), (x, z)\}$ . It can be assumed that a tournament in which there exists such a region represents a structure of the participants of the tournament. This structure is nonuniform; i.e., there exists a central region (set) of participants who can win against the other players, but they are in neutral position with respect to one another.

For solving the above problem, we propose the following exact formulation in the language of monotonic systems. In Section 2 we have considered weight arrays on vertices and arcs of a graph. Now let us consider the above models on vertices or arcs in a certain order. In both models we take a cyclic triple as the small graph  $g$  with respect to which the  $\pi$  function is calculated. Suppose that the methods of assignment of collections of weight arrays on vertices are the same as in Section 2. It is possible to modify this scheme by taking as a  $\ominus$ -action on the vertex  $\alpha$  the removal of all arcs of a tournament that originates at  $\alpha$ , whereas  $\oplus$ -action is the restoration of all the arcs in the graph  $\Lambda$  that originate at  $\alpha$ . In Section 2 we performed the opposite operation, i.e., the removal of incoming arcs and the restoration of these same incoming arcs.

The assignment of weight arrays on arcs of a tournament graph must be carried out in accordance with a scheme similar to that described in Section 2. Within the framework of the theory it is apparently impossible to decide whether the scheme of determination of kernels on arcs of a tournament is preferable to the scheme using vertices; therefore, it is necessary to carry out computer experiments. There exists only one heuristic consideration. If in a tournament there can exist several central regions saturated with cyclic triples, it will be preferable to use the scheme of determination of kernels on the arcs of tournament, since these regions can be found. The model based on vertices makes it possible to find a kernel that consists also of regions, but it does not permit finding an individual region. We do not possess a string of arcs representing these regions.

Acyclic directed graphs are a convenient language for describing operation systems [10]. An operation system can be regarded as a system of modules and interpreted as a library of programs. Each working program is a path in an acyclic graph, or, in other words, the set of modules of a library needed at a given instant. The modules are called one after another if not all of them can be stored in the main memory. In case of a library of a large size, there naturally arises the question of fixing the modules on information carriers. Prior to solving this problem, it is appropriate to ascertain the “structure” of an acyclic graph of a library of modules.

For ascertaining the structure of a graph and for just-mentioned task of fixing the modules, we have to find the principal (nodal) vertices or arcs. The nodes are the “bottlenecks” of graphs or, in other words, the modules that occur in many working programs.

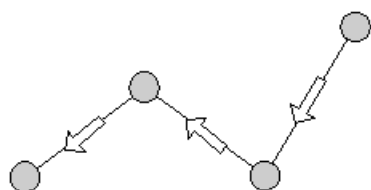


Fig. 4

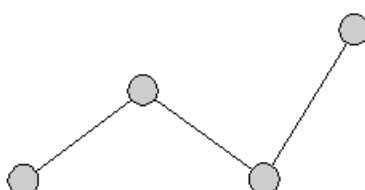
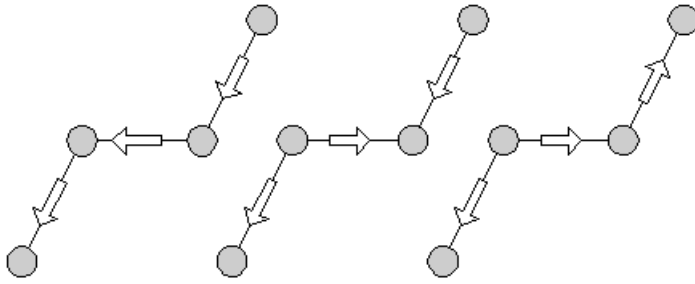


Fig. 5

We shall now formally describe this problem with the aid of a model of organization of a monotonic system on a graph. As a small graph we shall take directed graph in Fig.4. The structure of this graph is in accordance with the above definition of bottlenecks of the acyclic graph under consideration. It is possible to construct a monotonic system also on the arcs of an acyclic graph of a library of modules. The collection of weight arrays must be defined with the respect to the graph on Fig.4, and the  $\oplus$  and  $\ominus$  actions must be defined in accordance with the general scheme of Section 2. After this it is necessary to use the procedure of finding vertex kernels or arc kernels which in conjunction must indicate the bottlenecks in accordance with the above definition. As in case of tournaments, which a monotonic system is preferable of arcs or vertices requires experimental checking.





**Fig. 7**

The weight function  $\pi$  on a directed tree can be calculated separately with respect to each small graph  $g_1$ ,  $g_2$  and  $g_3$ ; then the values of all these three functions can be added up (a linear combination), thus yielding the overall function with respect to the graphs  $g_1$ ,  $g_2$  and  $g_3$ . In the same way we can assign a monotonic system on arcs of a tree if  $\ominus$  action signifies the removal of an arc of a tree, and  $\oplus$  action the restoration of an arc on a copy of given tree on  $\Lambda$ . Thus we can pose on directed trees a similar problem of finding cluster kernels. Let us note that we use in the last example with trees a more general model of assignment of collections of weight functions with respect to a series of small graphs. The model in Section 2 has been presented for one graph  $g$ . A collection of weight arrays with respect to a series of graphs has also the property of monotonicity, and apparently such a model is more interesting in solving problems of determination of “saturated” parts of graphs.

The author expresses his gratitude to L.K. Võhandu for his assistance and steady interest and to É.M. Braverman and I.B. Muchnik for valuable advice and remarks.

### L I T E R A T U R E C I T E D

1. J. E. Mulla, “Extremal subsystems of monotonic systems,” I, *Avtom. Telemekh.*, No.5, 130(1976), <http://www.dataundering.com/download/extrem01.pdf>.
2. J. E. Mulla, “Extremal subsystems of monotonic systems,” II, *Avtom. Telemekh.*, No.8, 169(1976), <http://www.dataundering.com/download/extrem02.pdf>.
3. I. B. Muchnik, “Analysis of structure of experimental graphs,” *Avtom. Telemekh.*, No.9, 62 (1974).
4. O. Ore, *Theory of Graphs*, American Mathematical Society, Providence, R.I. (1962).
5. C. Berge, *Theory of Graphs and its Applications*, Dunod, Paris (1958).
6. F. Harari, *Graph Theory*, Addison-Wesley, Reading; Mass. (1969).
7. G. N. Povarov, “Structural theory of communication networks,” in: *Probl. Peredachi Inf.*, No.1, 126 (1959).
8. V. L. Arlazarov, A.V. Uskov, and I.A. Faradzev, “Algorithms of determination of simple cycles in a directed graph,” in: *Discrete Mathematical Studies*, Nauka (1973), pp. 178-183.
9. M. G.Kendall and B.B. Smith, “Method of paired comparisons,” *Biometrika*, 31, 324 (1940).
10. V. A. Mikhalkov, “Method of organization of the computational process in operation systems,” *Kibernetika*, No.2, 141 (1970).