SECTION 13.
SYSTEM ANALYSIS, MODELING AND OPTIMIZATION

Markova Anastasiia
Faculty of Applied Mathematics
Oles Honchar Dnipro National University, Ukraine

Turchyna Valentina
candidate of physical and mathematical sciences, associate professor, head of the department of computational mathematics and mathematical cybernetics
Oles Honchar Dnipro National University, Ukraine

TRANSFORMATION OF GRAPHS WHEN CALCULATING CHARACTERISTICS IN PERT NETWORKS

PERT (Program Evaluation Review Technique) [1] is a network that graphically represents the chronology of the project and allows to manage the project by breaking it down into separate tasks for evaluation and analysis of work execution terms.

There are the following characteristics of a PERT network: length of the critical path, start moment of each event, permissible delays in the execution of each task. Calculation of each characteristic is a separate task.

An arbitrary directed graph $G(V, U)$ is given. If it is a network, then the characteristics are calculated on this network.

Conversion of the initial digraph is required if it is not a network. This can be in one of the cases: there is no source, there is no sink, there are vertices that have neither ingoing nor outgoing arcs. In these cases, the transformation of a graph into a network is trivial. But sometimes when planning projects, there may be cases when some task sequences forms a cycle. This may correspond, for example, to some verification of the fulfillment of the subset of tasks that forms a strongly connected component in the corresponding graph. In this case, when creating a PERT network, it is necessary to replace the corresponding strongly connected component of the graph with one vertex. For problems unrelated to PERT networks, there are the following algorithms for finding strongly connected graph components:

- Kosarajyu’s algorithm [2];
- Tarjan's algorithm [3];
- path-based strong component algorithm [4].

In this paper an algorithm for transition from a graph to a network if there are strongly connected components in the graph is proposed. Just like well-known algorithms for finding a strongly connected component, the proposed algorithm consists of several parts: arrangement of marks and transposition of the adjacency matrix that defines the graph.

For given graph G it is necessary to first create a vector of marks of the graph vertices, the i-th element of which corresponds to the number of the vertex. Initially, it is a vector of zero elements.

From the start, all vertices of the graph are unmarked. Next, we will mark the vertices.

1. Let $i = 1$.

2. If all vertices are marked, then the end of the first stage of placing marks.
3. Find the first occurrence of 0 in the vector.
4. For the corresponding vertex, do a breadth-first (or depth-first) search during which mark the vertices. As a result of this step, the vertex of the corresponding component will receive the mark i. $i++$. Go to 2.

Next, we should transpose the matrix A, which is the adjacency matrix of the initial graph G. Let's create a new vector of marks for the graph obtained from the transposed matrix. Denote this graph as G^T.

For this graph, we will do the following steps:

1. Sort the first vector of marks in non-ascending order.
2. For a fixed number i ($i > 0$), do a breadth-first (or depth-first) search for each vertex in random order. Passed vertices will receive a mark for the new vector. If the vertex was already passed in the previous iteration, its mark is increased by 1.

As a result, we obtain search trees, which are strongly connected components.

When building a corresponding PERT network by a given graph, we replace each strongly connected component with a new vertex. Ingoing to this new vertex arcs are new outgoing arcs from those vertices from which there were arcs to the vertices of the given component.

If such an arc went out to only one vertex, then its weight remains, and if there were more than one vertex, then the weight of the arc is equal to the sum of the weights of the corresponding arcs.

Outgoing arcs from the corresponding new vertex are new arcs to those vertices that were including arcs from the vertices of the strongly connected component. The weight of the corresponding arcs is determined by analogy with weight of ingoing arcs to the new vertex.

New vertex has a weight, which is the sum of the weights of all arcs of the corresponding strongly connected component.

The proposed algorithm makes it possible to take into account those cases when the order of execution of some tasks can be cyclical when constructing PERT networks.

References:

1. Harvey M. Sapolsky. The Polaris System Development: Bureaucratic and Programmatic Success in Government. – 1972. – 261 p.
2. Micha Sharir. A strong-connectivity algorithm and its applications in data flow analysis. Computers & Mathematics with Applications. – №7. – 1981. – P. 67-72.
3. Robert E. Tarjan. Depth-first search and linear graph algorithms. – 1972. – P. 146-160.
4. Edsger Dijkstra. A Discipline of Programming. Englewood Cliffs, NJ: Prentice Hall. – 1976. – 217 p.