One Approach for Describing the Configuration of the Routes Network   
in Open-Pit Mines

*Nikolay B. Dokev*

*New Bulgarian University*

In order to realize the operative shift control on the basis of an automatic system, it is necessary to have a mathematical model, that corresponds precisely to the plant controlled. For this purpose it is appropriate to construct a graph model of the route network, which gives possibility for efficient simulation of the truck-haulage functions.

The discrete messages from the passing by transport vehicles enter into a final number of information stations. They are located at appropriate points of the route network, such as loading unloading stations, garage and distribution points.

Since the open-pit enterprises have different route configurations, that constantly change, the development of a graph model and a method for describing the elements of a route network with an arbitrary configuration, corresponding to the requirements towards the functioning of the truck-haulage process, is necessary.

The correct operation of an automatic control system requires information about the route network configuration at each moment, regardless of the location and the connections of the stations and the transport process organization. The configuration description must be done for arbitrary alteration of the network parameters, on the basis of the apriori data and the information, entered by the dispatcher, or automatically input from the data acquisition system.

It is assumed, that the route network at moment t can be described with the help of a directed graph *G*(*Ei*\*, *i*\*) in which there is mutual unique correspondence between the set of nodes *Ei*\* and the set of arcs *i*\* and the route sections. The graph can be altered depending on the changes in the route network status. It is built on the basis of apriori information about the route network – the information stations and the connecting road sections.

Let the set of nodes *Ei*\* is divided into several classes *Ii*, *i* = 1, 2, …, *n*, each one connected with particular characteristics of the corresponding information station in the transport system.

The use of a complete graph for the description of the information stations and the connections among them, in a real time automatic control system, is inappropriate because of the necessity to record this graph and to increase the configuration time.

In connection with this, a subgraph is defined on the complete graph *G*(*Ei*\*, *i*\*), that characterizes the maximum feasible real network. For it:

(1) 

and from the maximum possible set of arcs of the complete graph only those are present in *m*, that have functional importance for the system operation.

Let *Ki* denote the greatest index of a node in class *Ii*. In this case *Ii* = {*Ki*–1 + 1, …, *Ki*}. If there exist arcs in both of the directions among each arbitrary nodes, the total number of the arcs is equal to *Kn*, where *Kn* is the greatest index of a node in the graph. For example, in order to describe the route network in the open-pit mine “ASSAREL”, where the information points are 57, 3249 arcs are necessary.

Let *I*1, *I*2, …, *I*6  denote the sets of the nodes, corresponding to the loading points, the information points, the garage, the ore and rubble unloading points.

Since the transport vehicles are sent from the set of nodes *I*2 towards nodes from the set *I*1 or *I*4, and the opposite is not done, it is obvious, that between *I*2 and *I*1 and *I*4 respectively, there are directed arcs in one direction only.



Fig. 1

Fig. 1 shows, that on the maximum accessible graph there can be built a subgraph   
*G*(*En*, *n*) for *Sn* = {*I*1, *I*2, …, *In*}, characterizing the functionally necessary arcs. In this case a considerable part of the complete graph arcs falls out.

In connection with this only a part of the complete graph arcs are considered, that can be grouped into five zones: arcs, coming out from distribution points and going into loading points; arcs, coming out from unloading or distribution points and going into distribution points; arcs, coming out of distribution points and going into garages; arcs, coming out of loading points and going into unloading points; arcs, coming out from loading points and going into loading points.

It is necessary to choose such numbering of the arcs, where through analytical relations, the control computing complex can:

* define uniquely with the help of the indexes of two nodes, or the indexes of two information points, the index of the arc, connecting them;
* define uniquely the nodes, by the index of the arc, that connets them.

The unique correspondence is very important when numbering of arcs. One approach is to do it in such a way, that for each node (point), the index arc will increase only with respect to the arcs, going into this node, keeping the following conditions:

* the node with a smaller index has entering nodes with smaller indexes, than the indexes of the arcs, coming into the nodes with larger indexes;
* if in any node there are coming several arcs of one and thhe same group of nodes, the increase of the indices of these arcs corresponds to the increase of the indices of the nodes, out of wich they go out;
* if there are coming arcs of one or more groups of nodes into one node, the indices of the arcs, coming into this node and going out of a group of nodes with smaller indexes, are with smaller indexes respectively than the arcs, going out of groups of nodes with greater indexes.

This way of numerating the graph nodes and arcs and the dividing of the index sets into classes enables the creation of the following procedure for obtaining the indexes of the arcs, coming into the information points of the various subsets *Ii*:

(2) *B*(*j*) = *B*(*j* – 1) + *ri*, *i* = 1, 2, …, *n*, *j* = 1, 2, …, *kn*,

where *B*(*j*) is the greatest index of the arc, coming into the node with an index *j*, and *ri* is the element of the set *R*{*r*1, *r*2, …, *rn*}, which is equal in power to *Sn*. Each element *ri* is used to obtain the indexes of the arcs in the corresponding region. The values of *ri* are obtained trough a set of functions *fi*, defined on the set of the maximum indexes *ki* from classes *Ii*.

(3) *ri* = *fi*(*k*\*), *k*\* = {0, 1, *k*1, , *kn*}.

Since the relation (2) is recursive, in order to obtain the set *B*(*j*) it is necessary to define the initial element *B*(0) = 0. The element *ri* is selected according to the belonging of *j* to any of the classes *Ii*, i.e.,

(4) *B*(*j*) = *B*(*j* – 1) + *rm*, for *m* ∈ {1, 2, …, *n*} ˄ *j* ∈ *Im*.

The logical expression (4) enables the computing of the elements of *B*(*j*) with simple program means.

Such description of the route network enables the construction of procedures, in which according to the arc index, the existing *m* and entering *d* node of the arc are found and the index of their connecting arc *u* also, which is equivalent to a mutual unique relation (*m*, *d*) ⬄ *u*.

In order to describe the route network, it is necessary to enter the apriori information for the initial route network through sets, defining a graph *G*(*E*\*, **\*), which is a graph of *G*(*Em*, *m*).

Since in a given moment in the addressing system not all the arcs of the maximum feasible graph *G*(*Em*, *m*) are present, it is necessary to enter the graph *G*(*E*\*, **\*), which is a subgraph of *G*(*Em*, *m*) and includes only the elements, participating in the control system. This is achieved, creating the set *Bi*(*t*), the elements of which show the current state of the arcs (whether the corresponding arc is in the system and the mode of including). This set gives the dispatcher the opportunity to exclude any element of *G*(*E*\*, **\*) and to include in the control system elements, not participating in the apriori and current route network configuration.

A machine model of the network can be designed on the basis of the graph *G*(*E*\*, **\*) discussed, that is describing it completely enough, but is not convenient for real-time mode, since the finding and describing of all the feasible vehicles routes with a beginning at point *j*, it is necessary to check the status of a large number of arcs – incidental to the primary node *j* at first, and then the arcs, incidental to the input nodes *d*. This will lead to exponential increase of the computer addressing time as a function of the route length.

This can be avoided, giving an explicit description of the feasible routes of the transport vehicles, which is appropriate for machine interpretation. This is achieved, using the functional connections between the sets {*Ii*}, reflected in *G*(*Sn*, *Ln*). Each element of the set *Ln* = (*Ii*, *Ij*)is a set of arcs *u* = (*m*, *d*), *m* ∈ *Ii*, *d* ∈ *Ij*, connecting arcs of respective classes.

In order to decrease the addressing computer time and to increase the efficiency of the control algorithms, it is appropriate to reduce the description of the route network to the defining and storing of certain sets, indicating the indexes of the arcs in the apriori configuration and such, that are used at the moment of addressing, the arcs, with which it is addressed from each node, and others.

In correspondence with the method for describing a route network, the practical description of the route network elements is given in the paper and the way for their computing also.

For this purpose the following denotations of the main information points are introduced:

*I* – a set of indexes for information at the excavators;

*I* = {1, 2, …, *k*};

*H* – a set of indexes of the information points, when addressing the loading points.

*H* = *H*1{*H*2, *H*3}, *H* = {*k*1 + 1, …, *k*4};

*H*1 – a set of the indexes at the information points with traffic lights, addressing the loading points.

*H*1 = {*k*1 + 1, …, *k*2};

*H*2 – a set of indexes at the information points with indicating tables for addressing the loading points.

*H*2 = {*k*2 + 1, …, *k*3};

*H*3 – a a set of indexes at the information points at all auxiliary points,

*H*3 = {*k*3 + 1, …, *k*4};

*G* – a set of the indexes of the information points at all the unloading points.

*G* = {*k*4 + 1, …, *k*6}; *G* = *G*1 ∪ *G*2;

*G*1 – a set of the indexes for information at all the unloading points (ore).

*G*1 = {*k*1 + 1, …, *k*5};

*G*2 – a set of the indexes of the information points at all the unloading points (rubble)

*G*2 = {*k*5 + 1, …, *k*6};

*E* – a set of the indexes of all the information points

*E* = (*I* ∪ *H*) ∪ *G*.

In the chosen way of denotation, there are six groups, forming the following set of all the indexes of the main information points

** where *k*0 = 0.

The following sets of information points have significance in the formation of route network arcs.

*E*0 – a set of the indexes of the information points, participating in the formation of the maximum feasible set of arcs between the information points.

*F*′*t*\* – a set of the indexes of the information points, participating in the formation of the arcs on the route network at moment *t*.

*Ft*\* ≤ *E*0, *t* = 1, 2, 3, …

The requirement for the formation of an unique modular software, independent on the route configuration, implies the introduction of arcs between the separate information points. These arcs denote the direction of the transport vehicles movement from one information point to another.

Only such arcs, that have functional significance for the operation of the automatic systems, are entered.

The arcs in the automatic system can be divided into 5 separate zones.

FIRST ZONE

It includes the arcs *u*0(1), entering the nodes, denoting the indexes of the information points, attached to the excavators. This set of arcs consist of two subsets:

* a subset, containing arcs only – i.e., arcs whose existing and entering node is one and the same information point at an excavator;
* a subset, the arcs of which come out of one and the same information point with traffic lights. If the points, in which arcs from the first zone, are denoted by *d* ∈ {1, 2, …, *k*1}, and by *m* ∈ {1, 2, …, *k*2} – the information points, from which these arcs go out, the following necessary condition exists for the set of arcs, so defined:

if *m* ≤ *k*1, then *d* = *m*.

Let *u* = (*m*, *d*) denote the index of the arc, connecting the nodes *m* and *d*. the analytical relation for the first zone, connecting the values *u*, *m* and *d*, has the type:

*u* = (*d* – 1)(*k*2 – *k*1 + 1) + *m′*, *u* = *B*(*d* – 1) + *m′*;

where



Since the approach of successive numerating of the arcs, coming into each separate node is selected, the set *B*(*d*) is introduced to define the limits of these arcs. The lower part of the arcs, coming into the node *d*, is equal to *B*(*d* – 1) + 1.

It is assumed, that *B*(0) = 0. For the first zone for each node *d* ∈ *I*.

*B*(*d*) = (*k*2 – *k*1 + 1)*d*.

The lower and upper bounds of the arcs indexes from the first zone have the following values,

*B*(0) + 1 = 1, *B*(*k*1) = (*k*2 – *k*1 + 1)*k*1.

As it follows from the relations for the first zone, the arcs indexing is realized at first for the first excavator, after that for the second one and so on. for each excavator the lowest index is assigned to the node-loop, after that the arc from the traffic light with the smallest index of the information point and so on.

SECOND ZONE

The second zone includes the arcs *u*0(2), which are entering the information points at the traffic lights. They consist of two subsets: arcs, going out of the set of indexes at the traffic lights and entering the same set of nodes; arcs, going out of a set of the unloading points indexes and entering the set of indexes at the traffic lights.

The relation between the nodes *m* and *d* and the connecting arc (*m*.*d*) for the second zone has the following type:

*u* = *B*(*k*1) + (*d* – *k*1 – 1)(*k*2 – *k*1 + *k*6 – *k*4) + *m′*,

*u* = *B*(*d* – 1) + *m′*,

where 

*d* ∈ {*k*1 + 1, …, *k*2}, *m* ∈ {*k*1 + 1, …, *k*2} ∪ {*k*4 + 1, …, *k*6}.

The upper limit *B*(*d*) of the indexes of the arcs, going into each node *d* for the second zone corresponds to the *m* = *k*6, i.e., *B*(*d*) = *B*(*k*1) + (*k*2 – *k*1 + *k*6 – *k*4)(*d* – *k*1).

The upper and lower limits of the arcs from the second zone is equal to *B*(*k*1) + 1 and   
*B*(*k*2) = *B*(*k*1) + (*k*2 – *k*1 + *k*6 – *k*4)( *k*2 – *k*1).

The arcs indexing is done at first for the traffic light with the lowest index, after that for the next and so on. The arcs, coming out of the traffic lights in ascending order get the lowest indexes for each point, and after that the set of unloading points in ascending order also.

THIRD ZONE

The third zone includes the arcs *u*0(3), that go out of the information points at the traffic lights and go into the auxiliary information point at the garage.

It is assumed, that there is only one garage point for information and *k*3 + 1 = *k*4.

In this case the relation between *m*, *d* and the arc *u* = (*m*, *d*) is the following:

*u* = *B*(*k*2) + *m* – *k*1,

where *d* = *k*3 + 1 = *k*4, *m* ∈ {*k*1 + 1, …, *k*2}.

The arcs in the third zone are a unidimensional array and for them the information point, that they enter is fixed, that is why the arc number does not depend on *d* = *k*3 + 1. The upper limit in (*d*) of the arcs for *m* = *k*2 is defined by the relation

*B*(*d*) = *B*(*k*2) + *k*2 – *k*1, *d* = *k*3 + 1.

The upper and lower limits of the arcs zone are equal to:

*B*(*k*2) + 1, *B*(*k*4) = *B*(*k*2) + *k*2 – *k*1.

FOURTH ZONE

The fourth zone includes the arcs *u*0(4), going out of the loading points and going into the unloading points, i.e.,

*m* ∈ {1, 2, …, *k*1}, *d* ∈ {*k*4 + 1, …, *k*6}.

The relations between *m*, *d* and *u* = (*m*, *d*) are as follows:

*u* = *B*(*k*4) + (*d* – *k*4 – 1)*k*3 + *m*, *u* = *B*(*d* – 1) + *m*.

The upper limit *B*(*d*) of the indexes of the arcs, entering each node *d* corresponds to the case   
*m* = *k*1, i.e.,

*B*(*d*) = *B*(*k*4) + (*d* – *k*4 – 1)*k*1 + *k*1, ⇒ *B*(*d*) = *B*(*k*4) + (*d* – *k*4)*k*1.

The upper and lower limits of the arcs are:

*B*(*k*4) + 1 and *B*(*k*6) = *B*(*k*4) + (*k*6 – *k*4)*k*1,

respectively.

A successive way of indexing is accepted, the arcs, entering the unloading point with the smallest number obtaining indexes first, and after that the arcs, going into the next unloading point in ascending order and so on.

FIFTH ZONE

The fifth zone includes the arcs *u*0(5), connecting the separate loading points, but without the arcs-loops towards each of that points. Only the arcs, going out of the points with greater index to points with smaller index are included in this zone.

Hence:

if *d* ∈ {1, 2, .., *k*1 – 1}, *m*′ ∈ {1, 2, .., *k*1 – 1} and *d*0 = *m*,

then *m*′ = max(*d*, *m*) and *d*′ = min(*d*, *m*).

The upper limit *B*(*k*6 + *d*′) of the indexes of the arcs, going into each node *d*′ corresponds to the case *m*′ = *k*1 and is defined by the recurrent relation:

*B*(*k*6 + *d*′) = *B*(*k*6 + *d*′ – 1) + *k*1 – *d*′.

If subsequent summing of the two sides is executed for *d*′ = 1, 2, …, *k*1, it will be obtained:

*B*(*k*6 + *d*′) = *B*(*k*6) + *k*1*d*′ – .

From the relation



the following is obtained:

*B*(*k*6 + *d*′) = *B*(*k*6) + 

The following expressions give the relations between the nodes *m*′, *d*′ and the index of their connecting arc: (*m*′, *k*6 + *d*′):

*u* = (*m*′, *k*6 + *d*′) = B(*k*6 + *d*′ – 1) + *m*′ – *d*′;

*u* = (*m*′, *k*6 + *d*′) = B(*k*6) + (*d*′ – 1) + *m*′ – *d*′;

*u* = (*m*′, *k*6 + *d*′) = B(*k*6) + *k*1(*d*′ – 1)*d*′ + *m*′;

The upper and lower limit for all the arcs of fifth zone, is equal to:

B(*k*6) + 1 and B(*k*7) = B(*k*6) + .

The arcs indexing is done at first for the loading point with the lowest index, after that for the next one and so on. The indexing of the entering arcs for each point is realized from the node with the lower towards nodes with greater indexes.

In the approach accepted for thescription of the separate zones, the upper limit of the indexes from a given zone is expressed through the upper limits of the indexes from the preceding zone. If the necessary operations for deciphering of the recurrent relations are realized, the following common relations for the upper limits of the indexes of the separate zones will be obtained:

*B*(*k*1) = (*k*2 – *k*1 + 1)*k*1,

*B*(*k*2) = (*k*2 – *k*1 + 1)*k*2 + (*k*6 – *k*4 – 1)(*k*2 – *k*1),

*B*(*k*3) = (*k*2 – *k*1 + 1)*k*2 + (*k*6 – *k*4)(*k*2 – *k*1);

*B*(*k*4) = (*k*6 – *k*4 + *k*2 – *k*1 + 1)*k*2,

*B*(*k*7) = (*k*6 – *k*4 + *k*2 – *k*1 + 1)*k*2 + .



Fig. 2

Fig. 2 gives a scheme with 4 excavators and 3 identification points with traffic lights.

The following example illustrates the way of indexing for the first zone:

*k*1 = 4, *k*2 = 7, *d* = 3, *m* = 6,

u = (m.d) = (6.3) = (3 – 1)(7 – 4 + 1) + 3; u = (6.3) = 11.

The arc, connecting node 6 with node 3 on Fig. 2, has index 11 also.

References

1. Berg, K. Graph Theory and its Applications. M., II, 1962 (in Russian).
2. Ford, L. D., R. Fulkerson. Flows in Networks. M., Mir, 1966 (in Russian).
3. Harrari, F. Theory of Graphs. M., Mir, 1973 (in Russian).