METHOD FOR DETERMINING THE OPTIMAL LOCATION OF THE CONTROLLER IN SOFTWARE-CONFIGURABLE NETWORKS

Bekbaev G.
PhD, Associate Professor of the Department of Social, Humanitarian and Exact Sciences,
Tashkent State University of Economics

ABSTRACT

In the context of a digital economy, it is essential to incorporative innovative technologies in the production process to analyze the status of telecommunication networks. This requires solving emerging technical problems through the transition to software-configurable networks. By doing so, the network can be managed more efficiently, resulting in accelerated data transmission by relocating control functions to the controller. Recent research has failed to address the issue of a single point of failure of software-configurable networks, i.e., the controller. However, some scientific papers have proposed an optimal controller placement method to improve the network's reliability. Protecting redundancy, recovery, and optimal controller placement are the three primary ways to ensure the reliability of software-configurable networks. This article proposes a technique and algorithm for finding the optimal location of the controller on a given communication network.

Key words: *software-defined network, algorithm, graph, oriented graph, vertex, function, controller, router*

АННОТАЦИЯ

В условиях цифровой экономики важно внедрять инновационные технологии в производственный процесс для анализа состояния телекоммуникационных сетей. Это требует решения возникающих технических проблем за счет перехода на программно-конфигурируемые сети. Таким образом, сетью можно управлять более эффективно, что приводит к ускорению передачи данных за счет передачи функций управления контроллеру. Недавние исследования не смогли решить проблему единой точки отказа программно-конфигурируемых сетей, то есть контроллера. Однако в некоторых научных работах предложен оптимальный метод размещения контроллера для повышения надежности сети. Защита резервирования, восстановление и оптимальное размещение контроллера — три основных способа обеспечения надежности программно-конфигурируемых сетей. В

данной статье предлагается методика и алгоритм поиска оптимального расположения контроллера в заданной сети связи.

Ключевые слова: *программно-определяемая сеть, алгоритм, граф, ориентированный граф, вершина, функция, контроллер, маршрутизатор.*

INTRODUCTION

The research problem is an innovative approach to building communication network architecture. The proposed method allows increasing the network management capability and data transmission level by transferring functions to the controller [1, 2]. This method has the following benefits:

1. It offers programmable approaches to network management, which can be modified through the creation of new applications, leading to improved network management automation;

2. It improves network management by enabling the network to be reconfigured in real-time, adapting to changing operating conditions;

3. It allows for the modification of the network device structure, thereby reducing the number of protocols required to be handled, and only the instructions received from the controller need to be executed;

4. It reduces the cost of the network infrastructure as a whole, owing to cheaper system upgrades and reduced energy consumption.

The mathematical structure of the controller placement problem depends on the configuration of the domain of admissible points and the method used to evaluate the placement quality. This paper only considers placement problems in which the area of acceptable points of service centers is a graph, and the centers can be located at any vertex or on any arc of the graph [3].

METHODOLOGY FOR SELECTING A LOCATION FOR THE CONTROLLER

To facilitate the description of points on arcs and distances within the graph, it is essential to establish some definitions. The graph's vertex set consists of vertices numbered from 1 to n. Consider an arbitrary arc (i,j), whose length is equal to $a(i,j)>0$. Let f denote a point on the arc (i,j), which for all $0\le f\le 1$ is at $f\cdot a(i,j)$ units from the vertex i and at $(1-f)\cdot a(i,j)$ units from the vertex j. Let us call it f – point. Thus, the quarter point of the arc (i,j) is the point that is a quarter of the length of the arc (i,j) from the vertex i. The zero point of the arc (i,j) is the vertex i, and the unit point of the arc (i,j) is the vertex j. Hence, the vertices of the graph can also be regarded as arc points. Arc points that are not vertices are called interior points. Any point on an arc must be either an interior point or a vertex. Denote by X the set

Oriental Renaissance: Innovative, educational, natural and social sciences
SJIF 2023 = 6.131 / ASI Factor = 1.7
(E)ISSN:2181-1784
www.oriens.uz
3(6), June, 2023

of all vertices of the graph, and by P the set of all points. Thus $\{P-X\}$ is the set of all interior points. Let $l(i,j)$ denote the length of the shortest path from vertex i to vertex j. Then through L denote the matrix $n\times n$, in which the element (i,j) is $l(i,j)$. The elements in matrix L are called vertex-top distances. To calculate the elements of matrix L any of the algorithms can be used: Floyd's algorithm [4, 5] or Danzig's algorithm [6-8]. Let $l\left(f-(r,s),j\right)$ be the length of shortest path from f – point on the arc (r,s) to the vertex j. This value is called the point-top distance. If the arc (r,s) is undirected, let it be traversed in both directions, i.e. The shortest of the following two distances should be chosen as $l\left(f-(r,s),j\right)$:

1. Distance from point vertex r plus distance from vertex r to vertex j;
2. The distance from f – point to the top s to the top j, thus (1):

$$l\left(f-(r,s),j\right)=\min\left\{f\cdot a(r,s)+l(r,j)(1-f)\cdot a(i,j)+l(s,j)\right\}$$

1)

If the arc (r,s) is oriented, i.e. Its traversal is only allowed from r to s, then the first term in formula (1) can be excluded, then (2):

$$l\left(f-(r,s),j\right)=(1-f)\cdot a(i,j)+l(s,j)$$

2)

For a given arc (r,s) and vertex j the point-top distance, is a function of f on the graph should have one of the three types of dependencies shown in Fig.1.

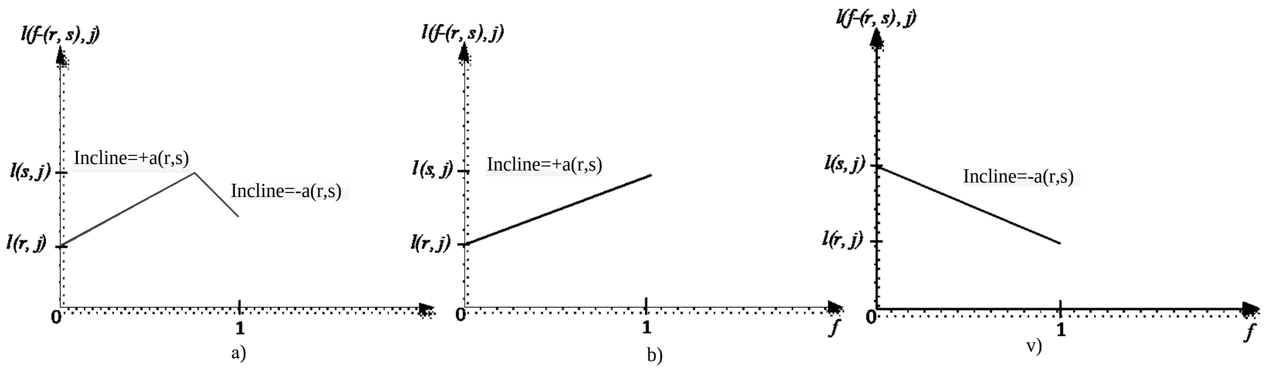

Figure 1. Graphs of the functions characterizing the point-to-top distance

Looking at the shortest distance from the vertex j to each point on the arc (r,s). For some point on the arc (r,s) this distance gives on a maximum value. It's denoted by $l'(j,(r,s))$ and called as vertex-arc distance. If there arc (r,s) is undirected, there are two transfer routes from the vertex j to f – point on the arc (r,s): through the vertex r or the vertex s. The shortest distance from these two routers is chosen. If these two routers from the vertex j to f – point on the arc (r,s) have other lengths, then some points adjacent to f – point on the arc (r,s) are even further away from the vertex j. For example, in figure 2 the quarter-point on the arc (3, 4) as an example 1,25 units or 2,75 units away from vertex 6, so it depends on whether they are moving through vertex 3 or through vertex 4.

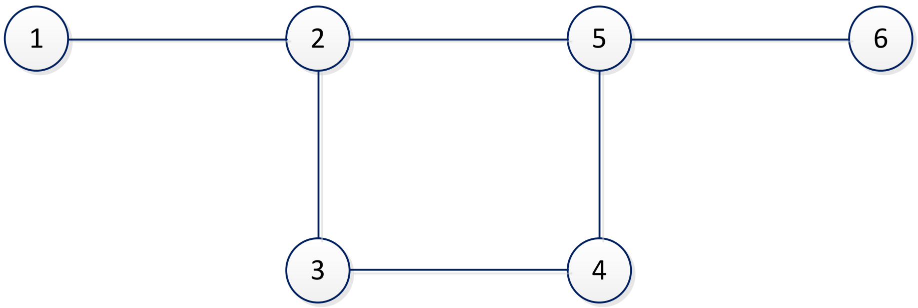

Figure 2. Example of a graph

If f increases from 0,25 to 0,26 then the shortest distance from vertex 2 to the value of 0,26 on the arc (3, 4) is $\min\{1,26;2;74\}=1,26$.

So, the two distances from a vertex j to some point on arc equal to each other if this point is the furthest point from the vertex j. It must be emphasized that total distances are always equal:

$$l(j,r)+f\cdot a(r,s)+l(j,s)+(1-f)\cdot a(r,s)=l(j,r)+l(j,s)+a(r,s)$$

Hence (3):

$$l'(j,(r,s))=\frac{l(j,r)+l(j,s)+a(r,s)}{2} \quad 3)$$

Looking at the other side, if (r,s) are oriented, then a few points on the (r,s) can be reached only with the vertex f. Hence, the most distant point on (r,s) from any vertex of the graph is the point closest to the vertex s. In this case (4):

$$l'\left(j,(r,s)\right)=d(j,r)+a(r,s)$$
4)

The graph includes and by pointing out L' a matrix of dimension $n\times m$, in which the element at the intersection of i-th row and k-column is the distance of vertex – arc from j-th vertex to k-th arc. These matrix values of elements can be calculated with a usage of equations (3) and (4), if the vertex-top distance given by matrix L' graph arc lengths are known.

And by highlighting $l\left(f-(r,s),(t,u)\right)$ the maximum distance from f – points on the arc (r,s) to points on the arc (t,u). This distance takes a name as the point-arc distance. If the arc (r,s) is undirected and $(r,s)\neq(t,u)$ then the route from f – point on the arc (r,s) to the farthest point on the arc (t,u) must come either through the vertex r, or through the vertex s. Hence (5):

$$l\left(f-(r,s),(t,u)\right)=\min\left\{f\cdot a(r,s)+l'(r,(t,u)),(1-f)\cdot a(r,s)+l'(s,(t,u))\right\} \quad 5)$$

If the arc (r,s) is oriented and $(r,s)\neq(t,u)$ then the first term in formula (5) can be excluded, then (6):

$$l\left(f-(r,s),(t,u)\right)=(1-f)\cdot a(r,s)+l'(s,(t,u))$$
6)

If $(r,s)=(t,u)$ is oriented, the furthest point on the arc (r,s) from f – point is g – point, where g tends towards f on the side of values smaller than f – point. It this case (7):

$$l\left(f-(r,s),(r,s)\right)=1-f\cdot a(r,s)+l(s,r)$$
7)

If $(r,s)=(t,u)$ and the arc (r,s) is undirected the maximum distance from f – point on the arc (r,s) to g – point on the arc (r,s), in case $g<f$, cannot exceed (8):

$$A\equiv\min\{f\cdot a(r,s),1/2\left[a(r,s)+l(s,r)\right]\}$$
8)

The first term in (8) equals the length of the route from f – point to g – point within the arc (r,s), the second term equals the length of the route from f and g –

points on the arc (r,s), but passing through the top s. It is resemble with the case of $g > f$, the maximum distance from f – point to g – point on the arc (r,s) cannot exceed (r,s) (9):

$$B \equiv \min\{(1-f)a(r,s), 1/2\left[a(r,s)+l(s,r)\right]\}$$
9)

The first term in expression (9) for B equals the length of the route from g-point within the arc (r,s), and the second term equals the length of the route from f-point to g-point on the arc (r,s) passing through the vertex r.

Hence, if the arc (r,s) is undirected, then $l'\left(f-(r,s),(r,s)\right)=\max\{A,B\}$, which is equal to (10):

$$l'\left(f-(r,s),(r,s)\right)=\max\begin{cases}\min\{f\cdot a(r,s), 1/2\left[a(r,s)+l(s,r)\right]\}\\ \min\{(1-f)a(r,s), 1/2\left[a(r,s)+l(s,r)\right]\}\end{cases} \quad 10)$$

If there is illustration of the point distance $l'\left(f-(r,s),(t,u)\right)$ as a function f for all $(r,s)\neq(t,u)$, then the corresponding curve on the graph will include the same forms as the point-top distance curve presented in Fig.3, since equations (5) and (6) will include accordingly the same form as equations (1) and (2). They have differences only in constants. Looking at the other, if $l'\left(f-(r,s),(r,s)\right)$ for any undirected arc (r,s) is represented as a function of point f, and then the function curve will include the form shown in Fig.3.

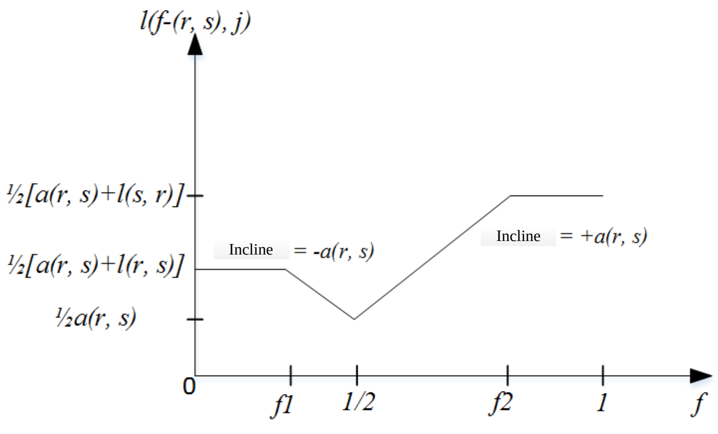

Figure 3. Diagram of the distance function point-arc $l'\left(f-(r,s),(t,u)\right)$

Therefore, the shown explanation can be systematized as illustrated in the table below:

Table-1

Designation	Name	Method of determination
$a(i, j)$	Arc length	Set
$l(i, j)$	Top-to-top distance	The Floyd or Danzig algorithm
$l'\left(f-(r,s), j\right)$	Point-to-top distance	Equations (1) and (2)
$l'\left(f-(r,s)\right)$	Distance apex-arc	Equations (3) and (4)
$l'\left(f-(r,s),(t,u)\right)$	Arc-to-arc distance	Equations (5), (6), (7) and (10)

Let (11):

$$MVV(i) = \max\left\{l(i, j)\right\} \quad 11)$$

– is the maximal distance from vertex i to the vertices of the graph. The distance from the vertex i to the most distant vertex of the graph. Then (12):

$$SVV(i) = \sum_j l(i, j) \quad 12)$$

– is the total distance from a vertex i to all vertices in the graph.

Let (13):

$$MTV(j-(r,s)) = \max\left\{l\left(f-(r,s), j\right)\right\} \quad 13)$$

– is the maximal distance from f-point on the arc (r,s) to the vertices of the graph. The distance from f-point on the arc (r,s) to the distant vertex of the graph, so (14):

$$STV(j-(r,s)) = \sum_j l\left(f-(r,s), f\right) \quad 14)$$

– is the total distances from f-point on the arc (r,s) to all vertices of the graph.

By providing clear definitions for the maximum and total values of these distances, it is possible to define with precision the different types of placements that will be discussed next.

1. The center of a graph G is any vertex x of this graphs that (15):

$$MVV(x) = \min\{MVV(i)\}$$

15)

Thus, the center is any vertex from which the distance to the vertex furthest away from it is minimal.

2. The main center G is any vertex x of this graph that (16):

$$MVD(x) = \min\{MVD(i)\}$$

16)

I.e. main center is any vertex whose distance from the furthest point on the graph arcs is minimal.

3. The absolute center of graph G is any f point on an arbitrary arc (r,s) of this graph, that (17):

$$MTV(f-(r,s)) = \min\{MTV(f-(t,u))\}$$

17)

I.e, the absolute center is any point – on the arc whose distance from the furthest vertex of the graph is minimal.

4. The main absolute center of the graph G a point f on an arbitrary arc (r,s) of this graph that (18):

$$MTD(f-(r,s)) = \min\{MTD(f-(t,u))\}$$

18)

Thus, the main absolute center – is any point from which the distance to the furthest point is minimal. The definitions of placement (1...4) are exactly the same as the definitions of the corresponding previous placement types, except that everywhere the maximization operator: $\left[\text{I.e. } MVV(i), MVD(i), MTV(f-(i,u)), MTD(f-(t,u))\right]$ is replaced by the summation operator$\left[\text{I.e. } SVV(i), SVD(i), STV(f-(i,u)), STD(f-(t,u))\right]$.

The next solution for the problem there is an introduction of the following concepts [9]. The eccentricity $e(a_i)$ of a vertex in a connected graph $G(A,\ B)$ is defined as $\max\left\{l(a_i, a_j)\right\}$. Radius of a graph $r(G)$ is the smallest of eccentricities of vertices. The vertices a_i – is called the central vertex of a graph if $e(a_i) = r(G),\ \ a_i \in A$. The center of a graph is the total of central vertices. Let's use the set $N_\lambda^0 = \left\{a_i \middle| l_{ij} \le \lambda,\ \ a_i \in A\right\}$. There is the total of all vertices the distance from a vertex a_i to which is not greater than λ. For each vertex there is a definition $C_0(a_i) = \max(l_{ij}),\ a_i \in A$. Let λ_0 – be the lowest value of λ such that for similar vertex $a_i : N_\lambda^0(a_i) = A$, i.e., the path length from the central a_i to any vertex of the graph does not exceed λ_0. Then $C_0(a_i) = \lambda_0$. A vertex a_0^*, such that $C_0(a_i^*) = \min\left[N_0(a_i)\right],\ \ a_i^* \in A$, is named the center of the graph $G(A,\ B)$ [10, 11].

SEARCH ALGORITHM AND RESULTS

So, the making a construction of formula the matrix $L_{n\times n}$ (n – is the array of the set A), where $l_{ij} = l(i,\ j)$ i.e. the matrix of shortest paths. We can use any of the above algorithms [4-12] to create it. Let us calculate the maximum in each row. Thus we get an array of length n, where i-th is the minimum length from i to the other vertices. Find the smallest element in this array. The vertex corresponding to these elements is the center of the graph. So, when there are a few vertices, all of them can be recorded as the center of diagram.

For instance: to find the centre of a weighted undirected graph. Observing weighted undirected graph $G(A,\ B)$; $A = \overline{1,7}$; $B = \left\{b_{ij}\right\}$ and $i,\ j = \overline{1,7}$ recorded in Fig.4.

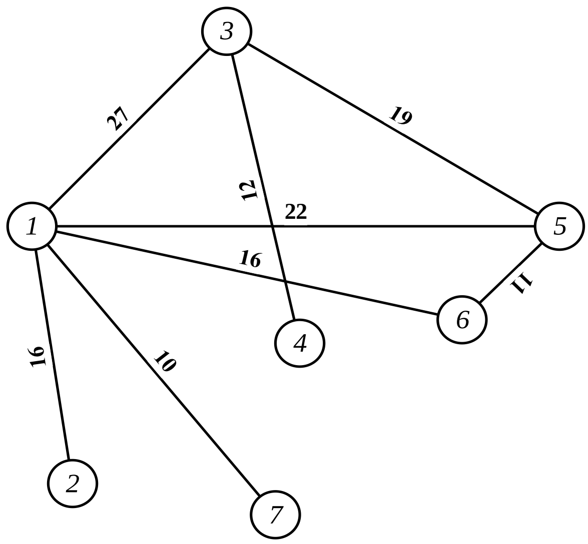

Figure 4. Weighted undirected graph $G(A,\ B)$

Let us make a matrix of the lengths of the shortest are is between each pair of vertices – L^0. If there is no arc between the vertices i and j is assigned to the element $l(i,j)$ of the matrix 0.

$$L^0 = \begin{pmatrix} 0 & 16 & 27 & 0 & 22 & 16 & 10 \\ 16 & 0 & 0 & 0 & 0 & 0 & 0 \\ 27 & 0 & 0 & 12 & 19 & 0 & 0 \\ 0 & 0 & 12 & 0 & 0 & 0 & 0 \\ 22 & 0 & 19 & 0 & 0 & 11 & 0 \\ 16 & 0 & 0 & 0 & 11 & 0 & 0 \\ 10 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

With the usage Floyd-Worschell algorithm [4, 5] we can get a matrix of shortest path lengths between each pair of vertices in the graph:

$$L = \begin{pmatrix} 0 & 16 & 27 & 39 & 22 & 16 & 10 \\ 16 & 0 & 43 & 55 & 38 & 32 & 26 \\ 27 & 43 & 0 & 12 & 19 & 30 & 37 \\ 39 & 55 & 12 & 0 & 31 & 42 & 49 \\ 22 & 38 & 19 & 31 & 0 & 11 & 32 \\ 16 & 32 & 30 & 42 & 11 & 0 & 26 \\ 10 & 26 & 37 & 49 & 32 & 26 & 0 \end{pmatrix}$$

So, according to the result of matrix of shortest path lengths, find the eccentricity for each vertex of the graph:

$e(a_i) = \max\{l(a_i, a_j)\}$;

$e(a_1) = 39$;

$e(a_2) = 55$;

$e(a_3) = 43$;

$e(a_4) = 55$;

$e(a_5) = 38$;

$e(a_6) = 42$;

$e(a_7) = 49$.

The centre of the graph is a vertex A, for which $e(a_i) = r(G),\ a_i \in A$. The vertex «5» has the minimum radius value $e(a_5) = 38$, which means that vertex «5» is the centre of the graph.

Algorithm for placing control points, taking into account physical and geographical conditions. The algorithm for placing control points in consideration of physical and geographical conditions has been a subject of various studies [4-12], in these studies, the problem of determining the node base is addressed by identifying the set of nodes A that meet the required network resource standards concerning channel quality. In addition to satisfying the requirements for channel quality and determining the number of nodes, solving the problem of optimal node placement in the designated area is also necessary. The node basis is a set $|A| = N$, where $|A| = \{a_i\}$, $\{x_i, y_i\}$, i=$\overline{1,N}$ – is a set of communication nodes (CN), and $\{x, y\}$ – geographical coordinates of communication nodes. At the time of determining the controller's location on the network, the location of the CN A – will be determined. Formally, the formulation of the problem of forming a node base can be written: $R_v = \sum_{i=1}^{N} a_i \to \min$.

In essence, fulfilling quality requirements can be seen as the process of creating channels that meet a specific quality standard. This involves selecting the desired path on the network and determining the channel rank, which dictates the allowable phase jitter and digital signal delay time. In order to make optimal use of line power, it is widely accepted in network and transmission system theory that simple channel lengths along the path must be equal:

$$l(a_1;a_2) \approx l(a_2;a_5) \approx ... \approx l(a_r;a_k) \approx ... \approx l(a_g;a_i) \approx ... \approx l(a_i;a_j) \approx R_0$$

where R_0 – is the rational distance between neighboring network nodes.

The rational distance among neighboring network nodes in the R_0 network is determined based on the maximum length of the composite link $l_{\max}$ and similarly as the most stringent demands for the type of communication required for the digital channel (DC) parameters (19):

$$R_0 = f\left(J_{vx1}, J_{vx}^{TP}, l_{\max}, n\right) \quad 19)$$

One of the alternative solutions is to cover the operational area with a circle of radius $r = R_0/2$, where the coordinates of the center of the circle will indicate the possible location of the transport network node to fulfill the necessary condition for the quality of the DC. To select a coverage option, it is necessary to solve the problem of geometric optimization [13, 14], where it is required to find the efficiency of packing on a plane by circles of a given radius.

Consider the problem of packing equal circles with a radius $r = 1/2$ on a plane in such a way that each circle touches six other circles (Fig.5). Each such circle can be seen as a circle inscribed into a regular hexagon with the same centre, these hexagons fill the plane. The efficiency F_t is obtained by comparing the area of a regular triangle, whose vertices are the centers of three adjacent circles, with the areas of the sectors of the three circles contained within the triangle. The efficiency of such packing is equal to $F_t = \dfrac{3\pi \dfrac{(1/2)^2}{6}}{\dfrac{\sqrt{3}}{4}} = \dfrac{\pi}{2\sqrt{3}} \approx 0,9069$.

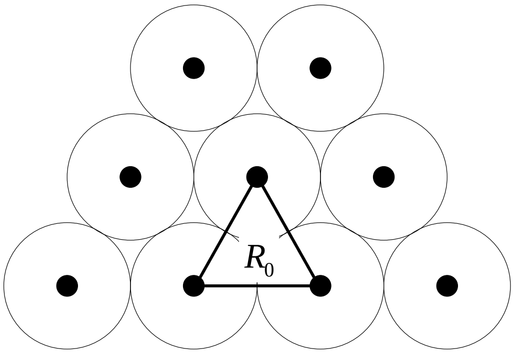

Figure 5. Covering a plane with circles using a triangle

A specific feature of the task of forming a nodal base is the presence of restrictions (physic-geographical conditions of the area and conditions of the operational situation) on the introduction of CTS (20):

$$\Re : F(a,b), g_k(\bar{a}) \le 0, \quad \text{k}=\overline{1,\ h} \qquad 20)$$

where a and b – are the width and depth of the operational area, and $g_k(\bar{a}) \le 0$ – are limitations on the introduction of CTS, determined by the physical and geographical features of the operational area.

If at the first version of the total location of the CS the coordinates of the input node got into the constraint area, then this point is brought the constraint line. This task is solved by the gradient projection method, where the aim function and the constraints are non-linear [15]. The algorithm of node base formation is presented in Fig.6.

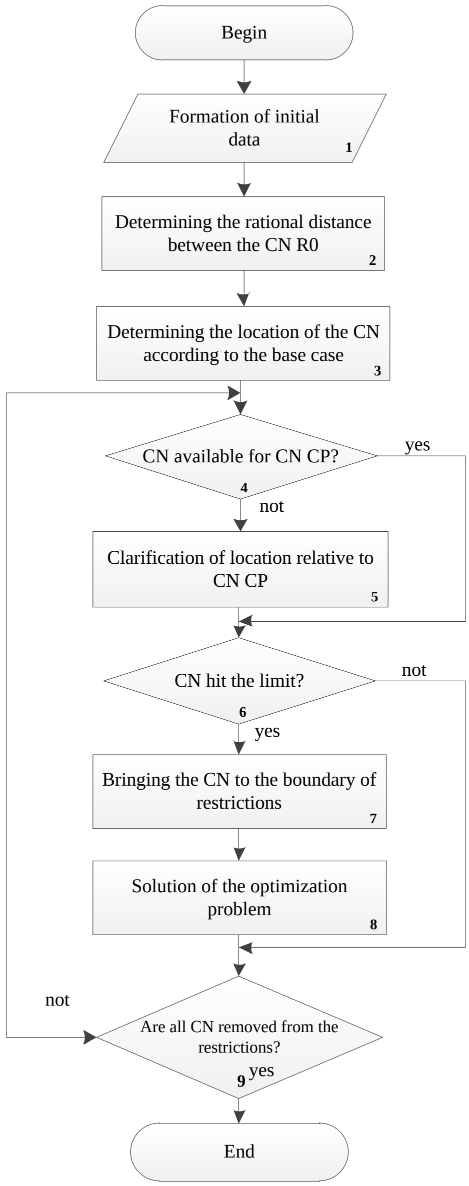

Figure 6. Block diagram of the algorithm for the formation of the nodal base

This algorithm includes two stages. At the first stage the basic solution (2) and (3) blocks of the algorithm are formed. It is based on the geometrical optimization methods and consists of covering the operational area with circles of $R_0/2$, where R_0 is a rational distance between the CN. Then an option is marked in terms of control point (CP) accessibility to CN and constraint on CN insertion.

CONCLUSION

In the case in the calculation, if the length of the reference lines of the CN CP does not allow to be tied to the CN, then the location is specified. In the case of a hit by the CN, in the basic case, it is necessary to find the optimal location with respect to the constraints. Determining the location of the CN relative to other CN and CN

CP is carried out using the modified Rosen algorithm [11] using blocks (7 and 8) of the current algorithm.

REFERENCES

1. Smelyansky R.L. Software-configurable networks. [Electronic resource] // Open Systems: Journal of Network Solutions / LAN. URL: https://www.osp.ru/os/2012/09/130324912
2. Kolomeyets A.E., Surkov L.V. "Software-configurable networks based on Open Flow protocol" // Engineering Herald 05.05.2014. pp. 519-524
3. Mainica E. Algorithms of optimization on networks and graphs: Translated from English: Mir, 1981. - 323 p.
4. Floyd's algorithm. URL: https://e-maxx.ru/algo/floyd_warshall_algorithm
5. Domnin L.N. Elements of graph theory: textbook. - Penza: Penza State University, 2007. - 144 p.
6. Novikov F.A. Discrete Mathematics for Programmers: a textbook for universities. 3rd ed. - St. Petersburg: Peter, 2009. 384 p.
7. Feofanova V.A., Vorotnikov V.I. Discrete Mathematics: textbook. Nizhny Tagil: NTI (branch) urfu, 2013. - 256 p.
8. The Danzig Algorithm. URL: https://scask.ru/g_book_dpr.php?Id=33
9. Tutt W. Graph Theory: Translated from English: World, 1988. - 424 p.
10. Berge C. Graph theory and its applications. Translated from French / Edited by I. A. Weinstein. - Moscow: IL, 1962. - 319 p.
11. Zykov A.A. Fundamentals of graph theory. - Moscow: 2004. - 664 p.
12. Gorbach A.N., Muravtsov A.A. Improvement of algorithm of node base construction of analogue-digital primary association communication network. Proceedings of XXXI Military Scientific Conference. 29 April 2002. Tver: Publishing house of All-Russian Military-Pilitary Association, 2002. pp. 56-58.
13. Yasinsky S. A. Unified mathematical models for analysis and synthesis of telecommunication network elements. Spb.: VAS, 2003. - 183 p.
14. Saati T. Integer optimization methods and related extreme problems. Moscow: Mir, 1973. – 304 p.
15. Bazara M., Shetty K. Non-linear programming. Theory and algorithms. Moscow: Mir, 1982. - 583 p.