Наведені результати теоретичних досліджень пошуку найбільшої кліки на прикладі оптимізації інформаційних систем залізничного транспорту. У більшості аналогічних розробок не описана можливість реалізації подібних методів для паралельних обчислень. Розроблений метод пошуку найбільшої кліки в довільному неорієнтованому графі з часовою складністю $O(2n(n^2\log_2 n+n))\approx O(2n^3\log_2 n)$ і малою похибкою. Ця розробка перспективна для складання алгоритмів паралельної обробки даних

Ключові слова: кліки в довільному неорієнтованому графі, максимальні кліки, методи для паралельних обчислень

Приведены результаты теоретических исследований поиска наибольшей клики на примере оптимизации информационных систем железнодорожного транспорта. У большинства аналогичных разработок не описана возможность реализации подобных методов для параллельных вычислений. Разработан метод поиска наибольшей клики в произвольном неориентированном графе с временной сложностью $O(2n(n^2\log_2 n+n))\approx O(2n^3\log_2 n)$ и малой погрешностью. Данная разработка перспективна для составления алгоритмов параллельной обработки данных

Ключевые слова: клики в произвольном неориентированном графе, максимальные клики, методы для параллельных вычислений

UDC 519.854: 004.75: 656.2

DOI: 10.15587/1729-4061.2017.111056

DEVELOPMENT OF MAXIMUM CLIQUE DEFINITION METHOD IN A NON-ORIENTED GRAPH

S. Listrovoy
Doctor of Technical Sciences, Professor*
E-mail: om1sergeyvladimirovih@gmail.com
V. Butenko
PhD, Associate Professor*
E-mail: docent.butenko@gmail.com
V. Bryksin
PhD, Senior Lecturer
Department Information Technologies**
E-mail: vladimir.bryksin@gmail.com
O. Golovko
PhD, Associate Professor
Department of Computer Engineering
and Control Systems**
E-mail: golovko.aleksandra1@gmail.com
*Department of Specialized Computer Systems**
**Ukrainian State University of Railway Transport
Feierbakh sq., 7, Kharkiv, Ukraine, 61050

1. Introduction

Implementation of information and measuring systems with analog-digital converters of high capacity and quick operation considerably enlarge the data flows [1]. The facilities of diagnosing the railway automatic equipment and mobile complete units [2] require communication resources in technological communication networks.

The change to digital networks based on optical dielectric waveguides makes the technicians optimize the resources providing the network traffic of branch subnetworks [3]. The loading irregularity of railway computing recourses is reasonable to be used for distributed computations. For this purpose, the computing system is simulated as a non-oriented graph. Programmatic definition of maximum cliques will let us optimize the computing resources and withhold us from premature hardware infrastructure investment.

2. Literature review and problem statement

The development of the distributed computing system monitoring method based on the definition of the shortest paths and the shortest Hamiltonian cycles in a graph is analyzed in the paper [4]. The model of the systems under discussion is equivalent to formal definitions of sparse graphs. It may be replaced by the Maximum Clique Problem (*MCP*), which is known as one of the hardest mathematical problems of the graph theory. As yet, we have no algorithms for solving the problem within polynomial time by parallel computing.

Meanwhile, the problem under discussion has lots of applications. In bioinformatics, *MCP* is used with a computer-based analysis of genomic databases, for example, when searching the potential regulatory structures of ribonucleic acids. In social networks, *MCP* is used while clustering the data, i. e. when dividing different communities into groups (clusters) that have general properties. Clustering allows each cluster to be processed using an individual helper server. In chemistry, *MCP* is the basis for searching “the maximum generic substructure” in a graph describing the structure of a chemical compound. Besides, *MCP* is a mathematical model of a number of problems arising from electronics design automation. The input data amount is vast (input graphs may contain up to million vertices). Thus, the present-day research direction of *MCP* is the development of fresh approaches to finding the exact and approximate solutions with regard to graph peculiarities resulted in applications. All the algorithms for *MCP* solution are classified as exact and approximate. Among the exact ones, there are clique-based algorithms, for instance, [5]; as well as backtracking algorithms, for instance, [6]. The exact algorithms operate in a time exponential from an input data amount. Among the approximate ones, there are combinatorial optimization and genetic algorithms [7]. These algorithms have an indefinite accuracy. The most known algorithms of finding the exact solution of *MCP* are Bron-Kerbosch and Wilf algorithms.

In the paper [8], the authors represent an advanced parallel binary algorithm for *MCP*. However, such an algorithm, even when it is advanced, has not been intended for high-dimensionality graphs. In the papers [9, 10], the authors have investigated a biphase algorithm for a clique problem solution, and clique issues for sparse graphs of high dimensionality, as well as decomposition of minimal clique separator. However, the presented methods leave out the admissibility of parallel computing.

The paper [11] presents the correction procedure of only a single computer error and in the papers [12, 13] the group of authors has suggested *MCP* solution for a degenerate graph and the approaches to finding the maximum quasi-cliques. So, it seems promising to develop the method of determining the maximum cliques in non-oriented graphs that makes possible the execution of different applications in information systems in a real-time scale.

3. The aim and objectives of the study

The aim of the paper is developing the method for defining the maximum cliques in non-oriented graphs with small time complexity. This will expand the admissibility of data clustering in computer systems and raise the effectiveness of electronics development.

To achieve the aim, it is necessary to do the following:

– to introduce procedure *B* making possible the definition of estimates of the largest graph clique size value;

– to introduce procedure *A* allowing the cliques to be formed on the base of each graph vertex with the help of procedure *B* and the maximum graph clique to be selected.

4. Materials and methods of the study

4. 1. Formal characterization and solution of the problem

Let us consider the *MCP* solution because of getting the estimates of the largest value of graph clique sizes based on a sufficiently clear assertion 1.

Assertion 1. If in graph *G* (*V*, *E*) there is a clique of *k* size, then the number of vertices *i* with $d_i \geq k-1$ power should be at least *k*, i. e. we shall say that there is an estimate of the largest value of maximum clique size $\Delta_{\max}=k$ in *G* (*V*, *E*) graph.

Let us assume that there is *G*(*V*, *E*) graph with *n* vertices. Let us build up *n* subgraphs as follows: select the vertex *i*=1, point out all the vertices related to it, and connect all the vertices with ribs according to the links of the source graph *G* (*V*, *E*). As a result, we've got subgraph G_1 without a vertex, subgraphs G_2, G_3,..., G_n are build up in the same way. Thus, generation of some arbitrary subgraph G_i based on some vertex *i* is resulted in selecting the subset of vertices related to vertex *i*. They are connected with ribs according to those vertex links that were held in *G* graph. It is clear, that if there is the maximum clique in source graph *G* (*V*, *E*), then the above clique is in one of subsets G_i.

Let us consider the possible generation of procedure *B* for defining the estimates of the largest clique size value in G_i subgraphs. Let us suppose that G_i graph is preset with d_v vertices selected in a descending order of powers. At that, the vertices with equal powers d_r are pooled in subsets $\{\alpha^r\}$ with $|\alpha^r| = p_r$ cardinality (where α^r – the numbers of corresponding vertices in G_i graph), and G_i is the number of vertices in G_i graph with equal powers d_r. The total number of such subsets is denoted by γ. As a result of sorting, we obtain a sequence of vertices satisfying the inequation:

$$\{d_{r=1}\} \in \alpha^{r=1} > \{d_{r=2}\} \in \alpha^{r=2} > ... > \{d_{r=\gamma}\} \in \alpha^{r=\gamma}. \qquad (1)$$

The sequence (1) will be stored as list S_γ. Because of assertion 1, the following procedure of defining the largest value of estimate $\Delta_{\max}$ as per list S_j can be suggested.

Procedure B of definition $\Delta_{\max}$:

Step 1. Sorting the vertices and forming the subsets $\{\alpha^r\}$.

Step 2. *r*:=1.

Step 3. Checking the inequation.

$$\sum_{r=1}^{r} P_r + P_r \geq d_r + 1.$$

If the inequation is done, then $\Delta_{\max} := d_r + 1$, and the procedure stops, if not, let us proceed to the following step.

Step 4. $r := r+1$ and let us proceed to step 3.

It is well to bear in mind, that for the inequation analyzed in step 3 when *r*=1, the sum is

$$\sum_{r=1}^{r=1} P_r = 0.$$

The example of finding $\Delta_{\max}$ for a graph.

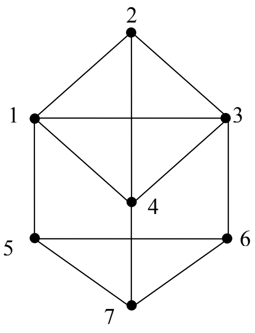


Fig. 1. Graph G

After sorting the vertices (Table 1), we obtain two subsets $\alpha^{r=1} = \{1, 3, 4\}$ with powers d_r=4 and $\alpha^{r=2} = \{2, 5, 6, 7\}$ with powers d_r=4. Let us check the inequation.

$$\sum_{r=1}^{r} P_r + P_r \geq d_r + 1,$$

in this particular case, it is as follows: $0+3 \geq 3+1$. The inequation isn't done, therefore $r := r+1 = 1+1 = 2$ and we pass into $\alpha^{r=2}$. Let us check inequation $P_{r=1}+P_{r=2} \geq d_{r=2}+1$, then we obtain 3+4≥3+1, that inequation is done, hence $\Delta_{\max}$=4. Thus, the maximum graph clique cannot exceed the size equal to 4. If the size of the graph clique is bigger, for instance size 5, then the number of vertices with power 4 should be more or equal to 5, and for clique size 4 the number of vertices with powers 3 should be more or equal to 4. The sorting results of graph cliques are presented in Table 1.

If the estimates of the largest values

$$H = \{\Delta_{\max i}\}$$

of maximum clique sizes are defined and they may be formed on the base of vertices $\{i\}, i=\overline{(1,n)}$ of some graph G, and in $H=\{\Delta_{\max i}\}$ set there are k vertices with the estimates

$$\{\Delta^{*}_{\max i}\}=k,$$

then the subset of those vertices will be as $h_{\max}=\{\Delta^{*}_{\max i}=k\}$. Let us say that $h_{\max}=\{\Delta^{*}_{\max i}=k\}$ subset has feature ν, if inequation $\Delta^{*}_{\max i}>\Delta_{\max i}$, with $\Delta^{*}_{\max i}\in h_{\max}$; $\Delta_{\max i}\in H$ is done. Then the following assertion is true.

Table 1

The results of sorting the graph vertices

$\{\alpha^r\}$	$\alpha^{r=1}$			$\alpha^{r=2}$			
i	1	3	4	2	5	6	7
d_r	4	4	4	3	3	3	3
P_r	$P_{r=1}$=3			$P_{r=2}$=4			

Assertion 2. If in set $H=\{\Delta_{\max i}\}$, obtained for some graph G, there is subset

$$h_{\max}=\{\Delta^{*}_{\max i}=k\},$$

satisfying feature ν, then the maximum clique size in a graph is equal to $|h_{\max}|=\Delta^{*}_{\max i}=k$ and the subset of vertices $\{i\}$ related to $\{\Delta^{*}_{\max i}\}=k$, form the above clique.

The correctness of assertion 2 results from the fact that on the base of all the vertices related to $h_{\max}=\{\Delta^{*}_{\max i}=k\}$, it is impossible to build up the cliques bigger in size than the cliques that may be built up on the base of vertices related to $h_{\max}=\{\Delta^{*}_{\max i}=k\}$. If $h_{\max}=\{\Delta^{*}_{\max i}=k\}$ estimates are the same, i. e. $|h_{\max}|=\Delta^{*}_{\max i}$, then it is impossible to build up the clique of a bigger size than on the set of vertices $\{i\}$, related to G graph $\{\Delta_{\max i}\}$.

If in $H=\{\Delta_{\max i}\}$, set for graph G there is no $h_{\max}=\{\Delta^{*}_{\max i}\}$, subset satisfactory to feature ν, then let us sort the above graph vertices in a descending order of estimates $\Delta_{\max i}$. The vertices with the same estimates $\Delta_{\max r}$ are pooled in subsets $\{\lambda^r\}$ with cardinality $|\lambda^r|=p_r$, the total number of such subsets is denoted by variable β. As the result of sorting, we obtain the sequence of vertices satisfactory to the inequation

$$\{\Delta_{\max r=1}\}\in\lambda^{r=1}>\{\Delta_{\max r=2}\}\in\lambda^{r=2}>\ldots>\{\Delta_{r=\beta}\}\in\lambda^{r=\beta}. \qquad (2)$$

The sequence (2) will be stored as list S_β. With the help of procedure B, we define the subset of vertices r^* with the following inequation

$$\sum_{r=1}^{r}P_r+P_r\geq\Delta_{\max r}+1.$$

Here all the vertices in list S_β after r^* are deleted out of graph G with the incidence ribs and those vertices are deleted out of the list.

Assertion 3. If subset $G^{/}$ is built on the base of list S_β, i. e. it is the result of vertices deletion where $r>r^*$, then it possesses the clique of maximum size.

The correctness of assertion 3 is clear as far as on the base of the vertices in list S_β after r^* it is impossible to build up the cliques of a bigger size than the cliques on the base of vertices in the list prior to r^*.

4. 2. The problem-solving method

Using procedure B, let us build A procedure of forming the maximum clique in some graph G. The basic data for procedure A is graph G itself and list S of the graph vertices.

Procedure *A:*

Step 1. Check in graph G the availability of $Y=\{i\}$ vertices related to all the vertices of G graph. Delete the set of Y vertices and incident to them ribs out of the graph and bring them into U set. After that, check all the vertices for deleting out of graph G. If they are deleted, the procedure stops as the source graph presents the clique of maximum size, if not, proceed to the next step.

Step 2. Check all the remainder of graph vertices with like powers. If so, check the remainder of vertices for vertex j, related to all the vertices in U. If so, the procedure is stopped as the set of vertices $\{j\bigcup U\}$ form the maximum clique in graph, if not, proceed to the next step.

Step 3. Select i vertex out of list S and on the base of vertices adjacent to i form subgraph $G_i\,(V_i,\,E_i)$.

Check subgraph $G_i\,(V_i,\,E_i)$ for vertices $D=\{i\}$ related to all the vertices of $G_i\,(V_i,\,E_i)$ subgraph, and delete the set of vertices D and incident to them ribs out of the graph and bring them into the set U_i. After that, check the vertices for deleting out of subgraph G. If so, then $\Delta_{\max}i=|V_i|$, if not, proceed to the next step.

Step 4. For current subgraph $G_i^{/}\,(V_i,\,E_i)$ on the basis of procedure B, define $\Delta^{|}_{\max i}$, then $\Delta_{\max i}=\Delta^{|}_{\max i}+|U_i|+|U|$, and bring $\Delta_{\max i}$ into set H, here assign the estimates equal to the maximum estimate in set H to vertices j entered into U, if they are available in that set, and bring them into H.

Step 5. Check list S for elements. If there are no elements, proceed to step 3, if so, proceed to the next step.

Step 6. In $H=\{\Delta_{\max i}\}$ select subset $h_{\max}=\{\Delta^{*}_{\max i}\}$ of the largest estimate values of vertices with maximum like values. Check $|h_{\max}|=\Delta^{*}_{\max i}$ inequation. If it is done, then the set of vertices for which the estimates $\{\Delta^{*}_{\max i}\}$,are obtained, generates the clique of maximum size and the procedure stops, if not, proceed to the next step.

Step 7. Sort out the vertices of that graph in a descending order of estimates $\Delta_{\max i}$, here pool the vertices with like estimates $\Delta_{\max r}$ into subsets $\{\lambda^r\}$ with cardinality $|\lambda^r|=p_r$, and denote the total number of those subsets by variable β. As a result of sorting out, we obtain the sequence of vertices satisfactory to inequation:

$$\{\Delta_{\max r=1}\}\in\lambda^{r=1}>\{\Delta_{\max r=2}\}\in\lambda^{r=2}>\ldots>\{\Delta_{r=\beta}\}\in\lambda^{r=\beta}.$$

It is saved as list S_β and then let us proceed to the next step.

Step 8. With the help of procedure B, define the set of vertices r^* which is the beginning of inequation

$$\sum_{r=1}^{r}P_r+P_r\geq\Delta_{\max r}+1.$$

Here delete all the vertices in list S_β, after r^* out of the graph G with incident to them ribs and delete these vertices out of list S_β. Then, $S{:}=S_\beta$ and proceed to Step 1.

4. 3. Example of problem solution by the developed method

Let us consider an example of procedure A performance for the graph shown in Fig. 2.

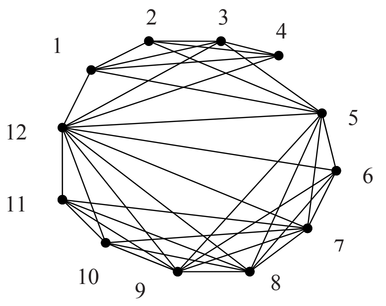
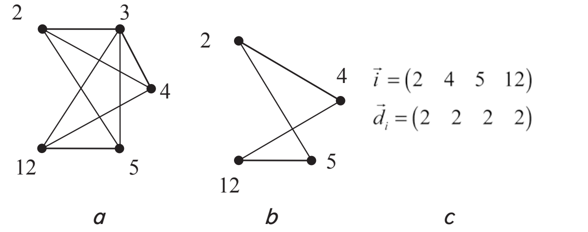


Fig. 2. Example of graph G for procedure *A*

The vertices list S={1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12} of graph G under analysis doesn't contain the vertices related to all the graph vertices, so the set U=Ø.

Next, start generating graphs G_i and find $\Delta_{\max i}$ for every graph G_i on the basis of procedure *B*. For this purpose, select the vertex i=1 which is related to vertices 2, 3, 4, 5, 10 out of the list, and on the ground of the vertices data subgraph G_1 is obtained (Fig. 3).



$$\vec{i} = (2 \quad 4 \quad 5 \quad 12)$$
$$\vec{d}_i = (2 \quad 2 \quad 2 \quad 2)$$

Fig. 3. Subgraphs and vectors: *a* – subgraph G_1; *b* – subgraph $G_1^|$; *c* – vectors of vertices and number of ribs

As it follows from Fig. 3, *a*, vertex 3 in a subgraph relates to all the subgraph vertices, so it is deleted along with the incident ribs and brought into set U1={3}. As the result, subgraph $G_1^|$ is obtained, Fig. 3, *b*. In Fig. 3, *c*, it is shown that as per procedure *B* $\Delta^|_{\max i=1} = 3$ and so

$$\Delta_{\max i=1} = \Delta^|_{\max i} + |U_i| + |U| = 3 + 1 + 0 = 4.$$

Let us define in the same way

$$\Delta_{\max 2} = 3;\ \ \Delta_{\max 3} = 4;\ \ \Delta_{\max 4} = 3;\ \ \Delta_{\max 5} = 5;$$

$$\Delta_{\max 6} = 5;\ \ \Delta_{\max 7} = 5;\ \ \Delta_{\max 8} = 6;\ \ \Delta_{\max 9} = 5;$$

$$\Delta_{\max 10} = 5;\ \ \Delta_{\max 11} = 6;\ \ \Delta_{\max 12} = 5.$$

Out of the obtained upper estimates set $H = \{\Delta_{\max} i\}$, let us select the upper estimates with maximum values, i. e. $h_{\max} = \{\Delta_{\max 8} = 6;\ \Delta_{\max 11} = 6\}$ and on their basis form sequence S_β, Table 2.

Table 2

Sequence S_β dependence on $\Delta_{\max i}$

$\{\beta^r\}$	$\beta^{r=1}$		$\beta^{r=2}$						$\beta^{r=3}$	$\beta^{r=4}$		
i	8	11	12	10	9	7	6	5	3	4	2	1
$\Delta_{\max i}$	6	6	5	5	5	5	5	5	4	3	3	3
P_r	$P_{r=1}$=2		$P_{r=2}$=5						$P_{r=3}$=1	$P_{r=4}$=3		

As is seen from Table 2, when procedure *B* starts the inequation

$$\sum_{r=1}^{r} P_r + P_r \geq \Delta_{\max r} + 1$$

is done at r^*=2 and, hence, vertices {3, 4, 2, 1} matched to r>2 values are deleted out of the source graph G and the list. Here a new list S_β={8, 11, 10, 9, 7, 6, 5} appears and graph G is converted into graph $G^/$ (Fig. 4) with vertices list S=S_β. And then procedure *A* cycle repeats for new graph $G^/$ with vertices list S=S_β.

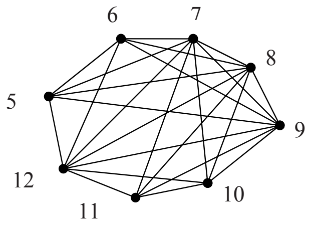


Fig. 4. Graph $G^/$

In graph $G^/$, vertices (7, 8, 9, 12) relate to all the vertices in $G^/$, so delete them with their incident ribs. In this case, bring them into set U={7, 8, 9, 12}, and graph $G^/$ will be as shown in Fig. 5.

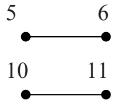


Fig. 5. Graph $G^/$ after deleting the vertices

It is easy to see that

$$\Delta^|_{\max i=5} = 2;\ \ \Delta^|_{\max i=6} = 2;\ \ \Delta^|_{\max i=10} = 2;\ \ \Delta^|_{\max i=11} = 2,$$

and hence

$$\Delta_{\max i=5} = \Delta^|_{\max i=5} + |U_{i=5}| + |U| = 2 + 0 + 4 = 6;$$

$$\Delta_{\max i=6} = \Delta^|_{\max i=6} + |U_{i=6}| + |U| = 2 + 0 + 4 = 6;$$

$$\Delta_{\max i=10} = \Delta^|_{\max i=10} + |U_{i=10}| + |U| = 2 + 0 + 4 = 6;$$

$$\Delta_{\max i=10} = \Delta^|_{\max i=11} + |U_{i=11}| + |U| = 2 + 0 + 4 = 6.$$

Assign the obtained maximum estimate equal to 6

$$\Delta_{\max 7} = 6;\ \ \Delta_{\max 8} = 6;\ \ \Delta_{\max 9} = 6;$$

to vertices set U={7, 8, 9, 12}. Thus, there are two subsets:

$$h_{\max 1} = \{\Delta^*_{\max i}\} = \{\Delta_{\max 7} = 6;\ \ \Delta_{\max 8} = 6;\ \ \Delta_{\max 9} = 6;\ \ \Delta_{\max 12} = 6;\ \ \Delta_{\max 5} = 6;\ \ \Delta_{\max 6} = 6\},$$

$$h_{\max 2} = \{\Delta^*_{\max i}\} = \{\Delta_{\max 7} = 6;\ \ \Delta_{\max 8} = 6;\ \ \Delta_{\max 9} = 6;\ \ \Delta_{\max 12} = 6;\ \ \Delta_{\max 10} = 6;\ \ \Delta_{\max 11} = 6\}.$$

For both the subsets, the condition $|h_{\max}| = \Delta^*_{\max i}$, is met, so 6=6. It follows that two cliques of maximum size equal to 6 consisting out of vertices (7, 8, 9, 12, 5, 6) and (7, 8, 9, 12, 10, 11) are formed.

4. 4. Assessment of problem solution algorithm complexity

Let us assess the complexity of performing procedure *A*. To form the sequences (1) and (2) for each vertex *i* based on procedure *B,* it is necessary to do no more than $n\log_2 n$ comparison operations and *n* addition operations. Hence, the maximum number of comparison operations for analysis of all the vertices will not exceed $2(n^2\log_2 n+n)$. As far as the cycle related to the omission of feature ν cannot be repeated more than n times, then the general complexity of procedure *A* implementation will not exceed

$$O(2n(n^2\log_2 n+n))\approx O(2n^3\log_2 n).$$

In this case, in compliance with assertions 2 and 3, we obtain either an optimal solution of the MCP or sufficiently close to it.

5. Discussion of investigation results of the developed problem solution method

Hence, procedure *A* that allows solving the problem of maximum independent set and minimum vertex cover is proposed. As far as graphs G_i may be formed independently, then the above procedure *A* may be vectorized. If system *n* is used for forming G_i processor, then the time complexity of procedure *A* falls down to $O(2n^2\log_2 n)$, and makes the problem of maximum clique definition in different applications be solved in a real-time scale.

It is also interesting to discuss the connection of the problem of defining the maximum independent sets, vertex covers and cliques with the problem of graph isomorphism and, in particular, the isomorphic embedding problem, i. e. whether graph $G(V, E)$ possesses isomorphic subgraph $H(V^{\backslash}, E^{\backslash})$.

It is more suitable to discuss not the Vizing product itself, but an additional to it graph. Later on, such a product will be called a modular product. Modular product $G\lozenge G^{|}$ of graphs $G(V, E)$ and $\mathrm{G}(V^{|}, E^{|})$ is called a graph specified by the following conditions:

1. $V\,(G \lozenge G^{|})=V\times V^{|}$ – Cartesian product of sets *V* and $V^{|}$;

2. Vertices $(u, u^{|})$ and $(v, v^{|})$ of graph $G\lozenge G^{|}$ are adjacent, if and only if at a time $u\neq v$ and $u^{|}\neq v$ or $uv\in V$, $u^{|}v^{|}\in\ V^{|}$, or $uv\notin V$, $u^{|}v^{|}\notin V^{|}$. Later, for convenience, we'll denote $\widetilde{xy}$ – unordered couple of elements x, y; $\overrightarrow{xy}$ – ordered couple of elements x, y.

In general, it is clear that density $\varphi\ (G\lozenge G^{|})\leq n$, as none of cliques of graph $G\lozenge G^{|}$ can contain two vertices out of one line or one column and $n\leq n^{|}$. Equality $\varphi(G\lozenge G^{|})=n$ is valid only if graph $G^{|}$ includes an isomorphic subgraph *G*. Let' assume that the vertices of both the graphs are natural numbers $V=\{1, 2, .., n\}$, $V^{|}=\{1, 2, ..., n, ..., n^{|}\}$. Let us suppose that at first $G^{|}$ includes subgraph $G^{||}(\mathrm{V}^{||}, \mathrm{E}^{||})$ isomorphic *G* and that isomorphism is generated by vertices correspondence.

$$\begin{array}{lllll} 1 & 2 & \ldots & n & \textit{in graph}\, G \\ \updownarrow & \updownarrow & \ldots & \updownarrow & \\ i_1 & i_2 & \ldots & i_n & \textit{in graph}\, G^{||} \end{array} \tag{3}$$

It is obvious that the subgraph of $G\lozenge G^{|}$ graph generated by *n* vertices $\overrightarrow{1i_1},\overrightarrow{2i_2},\ldots,\overrightarrow{ni_n},$ is a clique. As far as, if $\overrightarrow{ki_k}$ and $\overrightarrow{li_l}$ are any two different vertices $(k\neq l)$, so as correspondence ↔ between $G\lozenge G^{|}$ vertices is isomorphism, then either *k* and *l* are adjacent in *G*, and i_k and i_l are adjacent in $G^{||}$, or *k* and *l* are nonadjacent in *G*, and i_k and i_l are nonadjacent in $G^{||}$. In both cases, the vertices $\overrightarrow{ki_k}$ and $\overrightarrow{li_l}$ of graph $G\lozenge G^{|}$ are adjacent to each other. If conversely, graph $G\lozenge G$ includes the clique with vertices $\overrightarrow{1i_1},\overrightarrow{2i_2},\ldots,\overrightarrow{ni_n},$ then correspondence (3) is graph *G* isomorphism of graph $G^{|}$ subgraph $G^{||}$ generated by the set of vertices $V^{||}=\{i_1, i_2 ,..., i_\mathrm{n}\}$, so as follows from the definition $G\lozenge G^{|}$ $\widetilde{kl}\in V$ if and only if $\widetilde{i_k i_l}\in V^{|}$. So, all the *n*-cliques of graph $G\lozenge G^{|}$ one-to-one uniquely correspond to all possible *G* isomorphic embeddings as a subgraph in *G*.

It should be noted that because of the known connection of the clique problem with the problems of maximum independent set and minimum vertex cover [7], procedure *A* may be also used for solving those problems [4, 14].

6. Conclusions

Procedure *A* with small time complexity $O(2n^3\log_2 n)$, that allows from one viewpoint solving such problems as defining maximum cliques in non-oriented graphs, defining minimum independent sets and minimum vertex covers in graphs, as well as isomorphism of graphs and isomorphic embedding is generated.

Procedure *A* may be effectively vectorized that allows the time complexity of its operation to be decreased to $O(2n^2\log_2 n)$, and the mentioned problems to be solved in a real-time scale.

References

1. Butenko, V. M. Yakist informatsiyno-vymiriuvalnykh system na zaliznychnomu transporti Ukrainy [Text] / V. M. Butenko // Zb. naukov. prats. UkrDAZT. – 2008. – Issue 99. – P. 151–155.
2. Boynik, A. B. Diagnostirovanie ustroystv zheleznodorozhnoy avtomatiki i agregatov podvizhnyh edinic [Text]: ucheb. / A. B. Boynik, G. I. Zagariy, S. V. Koshevoy et. al. – Kharkiv: ChP Izdatel'stvo "Novoe slovo", 2008. – 304 p.
3. Listrovyi, S. V. Matematychne modeliuvannia v rozpodilenykh informatsiino-keruiuchykh systemakh zaliznychnoho transportu [Text]: monohrafiya / S. V. Listrovyi, S. V. Panchenko, V. I. Moiseienko, V. M. Butenko. – Kharkiv: FOP Brovin O.V., 2017. – 220 p.
4. Listrovoy, S. V. Monitoring distributed computing systems on the basis of the determined shortest paths and shortest hamiltonian cycles in a graph [Text] / S. V. Listrovoy, S. V. Minuhin, E. S. Listrovaya // Eastern-European Journal of Enterprise Technologies. – 2015. – Vol. 6, Issue 4 (78). – P. 32–45. doi: 10.15587/1729-4061.2015.56247
5. San Segundo, P. A new exact maximum clique algorithm for large and massive sparse graphs [Text] / P. San Segundo, A. Lopez, P. M. Pardalos // Computers & Operations Research. – 2016. – Vol. 66. – P. 81–94. doi: 10.1016/j.cor.2015.07.013
6. El Mouelhi, A. Some New Tractable Classes of CSPs and Their Relations with Backtracking Algorithms [Text] / A. El Mouelhi, P. Jégou, C. Terrioux, B. Zanuttini // Lecture Notes in Computer Science. – 2013. – P. 61–76. doi: 10.1007/978-3-642-38171-3_5
7. Zhang, S. A fast genetic algorithm for solving the maximum clique problem [Text] / S. Zhang, J. Wang, Q. Wu, J. Zhan // 2014 10th International Conference on Natural Computation (ICNC). – 2014. doi: 10.1109/icnc.2014.6975933

8. San Segundo, P. An improved bit parallel exact maximum clique algorithm [Text] / P. San Segundo, F. Matia, D. Rodriguez-Losada, M. Hernando // Optimization Letters. – 2011. – Vol. 7, Issue 3. – P. 467–479. doi: 10.1007/s11590-011-0431-y
9. Bykova, V. The Clique Minimal Separator Decomposition of a Hypergraph [Text] / V. Bykova // J. of Siberian Federal University. Mathematics Physics. – 2012. – Vol. 5, Issue 1. – P. 36–45.
10. Bykova, V. On CLIQUE Problem for Sparse Graphs of Large Dimension [Text] / V. Bykova, R. Illarionov // Communications in Computer and Information Science. – 2014. – P. 69–75. doi: 10.1007/978-3-319-13671-4_9
11. Minukhin, S. Efficient method for single machine total tardiness problem [Text] / S. Minukhin // 2012 IV International Conference "Problems of Cybernetics and Informatics" (PCI). – 2012. doi: 10.1109/icpci.2012.6486283
12. Buchanan, A. Solving maximum clique in sparse graphs: An O(nm+n2d/4) algorithm for d-degenerate graphs [Text] / A. Buchanan, J. L. Walteros, S. Butenko, P. M. Pardalos // Optimization Letters. – 2014. – Vol. 8, Issue 5. – P. 1611–1617. doi: 10.1007/s11590-013-0698-2
13. Veremyev, A. Exact MIP-based approaches for finding maximum quasi-cliques and dense subgraphs [Text] / A. Veremyev, O. A. Prokopyev, S. Butenko, E. L. Pasiliao // Computational Optimization and Applications. – 2015. – Vol. 64, Issue 1. – P. 177–214. doi: 10.1007/s10589-015-9804-y
14. Listrovoy, S. V. Parallel algorithms of calculation process optimization for the Boolean programming problems [Text] / S. V. Listrovoy, V. F. Tretiak, A. S. Listrovaya // Engineering Simulation. – 1999. – Vol. 16. – P. 569–579.

UDC 681.518:004.93.1
DOI: 10.15587/1729-4061.2017.112121

INFORMATION-EXTREME MACHINE LEARNING OF THE CONTROL SYSTEM OVER THE POWER UNIT OF A THERMAL POWER MAIN LINE

A. Dovbysh
Doctor of Technical Sciences,
Professor, Head of Department*
E-mail: a.dovbysh@cs.sumdu.edu.ua
D. Velykodnyi
PhD*
E-mail: d.velykodnyi@cs.sumdu.edu.ua
I. Shelehov
PhD, Associate Professor*
E-mail: igor-i@ukr.net
M. Bibyk
Postgraduate student*
E-mail: bibikm@gmail.com
*Department of Computer Science
Sumy State University
Rimskoho-Korsakova str., 2,
Sumy, Ukraine, 40007

Розглядається інформаційно-екстремальний метод навчання системи підтримки прийняття рішень для керування генеруючим енергоблоком теплоелектроцентралі. В процесі машинного навчання оптимізація контейнерів класів розпізнавання, що відновлювалися в радіальному базисі простору ознак, здійснювалася за модифікованим критерієм Кульбака. При цьому показано, що застосування вкладених контейнерів класів розпізнавання підвищує функціональну ефективність машинного навчання у порівнянні з контейнерами класів розпізнавання, центри яких розподілено в просторі ознак

Ключові слова: інформаційно-екстремальна інтелектуальна технологія, машинне навчання, система підтримки прийняття рішень, інформаційний критерій, енергоблок

Рассматривается информационно-экстремальный метод обучения системы поддержки принячтия решений для управления энергоблоком теплоэлектроцентрали. В процессе машинного обучения оптимизация контейнеров классов распознавания, восстанавливаемых в радиальном базисе пространства признаков, осуществлялась по модифицированному критерию Кульбака. При этом показано, что использование вложенных контейнеров классов распознавания повышает функциональную эффективность машинного обучения в сравнении с контейнерами классов распознавания, центры которых распределены в пространстве признаков

Ключевые слова: информационно-экстремальная интеллектуальная технология, машинное обучение, система поддержки принятия решений, информационный критерий, энергоблок

1. Introduction

Application of intelligent information technology for data analysis makes it possible to increase a functional efficiency of systems of control by weakly formalized processes. Processes of control of power units of thermal power plants relate to such processes. A promising way to increase a functional efficiency of weakly formalized controlled processes