ITERATIVE EQUITABLE PARTITION OF GRAPH AS A MODEL OF CONSTANT STRUCTURE DISCRETE TIME CLOSED SEMANTIC SYSTEM

E.E. Ivanko

Institute of Mathematics and Mechanics, Ural Branch of the RAS, Ekaterinburg, Russian Federation
Ural Federal University, Ekaterinburg, Russian Federation
E-mail: viatore@ya.ru

Constant structure closed semantic systems are the systems each element of which receives its definition through the correspondent unchangeable set of other elements (neighbors) of the system. The definitions of the elements change iteratively and simultaneously based on the neighbor portraits from the previous iteration. In this paper, I study the behavior of such model systems, starting from the zero state, where all the system's elements are equal. The development of constant-structure discrete time closed semantic systems may be modelled as a discrete time coloring process on a connected graph. Basically, I consider the iterative redefinition process on the vertices only, assuming that the edges are plain connectors, which do not have their own colors and do not participate in the definition of the incident vertices. However, the iterative coloring process for both vertices and edges may be converted to the vertices-only coloring case by the addition of virtual vertices corresponding to the edges assuming the colors for the vertices and for the edges are taken from the same palette and assigned in accordance with the same laws. I prove that the iterative coloring (redefinition) process in the described model will quickly degenerate into a series of pairwise isomorphic states and discuss some directions of further research.

Keywords: closed semantic system; graph; isomorphism.

Introduction

A closed semantic system (CSS) may be thought as a system each element of which is defined through other elements of this system. One of the most natural and important CSSs is language. Every child at the age of 3–5 years is full of questions: "Why?", "What for?", "How?" [1]. At that time, the child's world view is growing and getting as closed as possible: every word claims to be explained in terms of other words.

One of the most simple CSSs is a discrete time system with a permanent structure: 1) the number of the elements that are involved in the definition of each element of such system does not change with time; 2) all the elements are redefined simultaneously, basing on the states of the neighbours taken at the previous simultaneous iteration. In this paper, I study the behavior of such trivial systems, starting from the "zero" state, where all the system's elements are equal. In contrast with the above mentioned growing CSS of small kids, a CSS with a constant structure resembles an adult's world view, where the addition of new notions and connections between them ceases as the personal world view approaches the best known contemporary world view of the humanity. The simultaneous iterative changes in discrete time is a reasonable assumption as far as we consider an artificial CSS designed for a deterministic Turing machine equivalent [2]. However, the discreteness can

hardly correspond to real-world examples, so it would be challenging to get rid of it in the future.

The development of the described constant-structure discrete time CSS may be modelled as a discrete time coloring process on a connected graph. From now and till almost the end of the paper I will consider the iterative redefinition process on the vertices only, assuming that the edges are plain connectors, which do not have their own colors and do not participate in the definition of the incident vertices. Initially, all the vertices have the same "type", so at the first iteration the only difference between any 2 vertices is the number of their neighbours (degree). For illustrative purposes, we may assign each degree a specific "color". At the second iteration the "neighbour portrait" of each vertex becomes more complicated (e.g., "5 neighbours" at the first iteration becomes "2 red and 3 green" at the second). This iterative coloring process (see Fig. 1) produces equitable partitions [3] (perfect colorings, regular partitions or graph divisors), which are well known [4–6] and were successfully used in e.g. graph isomorphism heuristic Nauty [7] (the coloring process in the latter is similar to the Algorithm of this paper).
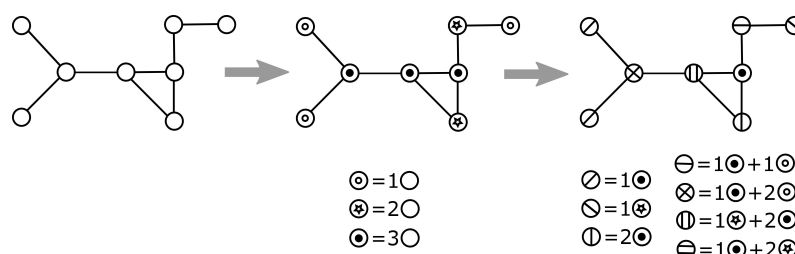

Fig. 1. Two iterations of the coloring process

How will this iterative coloring process behave? Will the size of the palette ever increase or it can decrease and then oscillate? Will the process become self-repeating? If yes, then how fast and what will be the cycle size? The next section gives the answers to these questions.

1. Mathematical Model

There are some little less common mathematical notations used below, which I would rather state explicitly: a) the number of adjacent vertices for each $v \in V$ in simple G is equal to the degree of v and is referred to as $deg(v)$; b) the image Y of the mapping $\phi\colon X \to Y$ is referred to as $\phi[X]$ and c) an arbitrary (possible multi-valued) mapping between X and Y is referred to as $X \rightrightarrows Y$.

Let $G = (V, E)$ be a simple graph with vertices V and edges E. Each vertex possesses a color, which changes iteratively depending on the colors of the vertex's neighbours. Let $Pal_i \subset \mathbb{N}$ be the set (palette) of colors and $Col_i\colon V \to Pal_i$ be the coloring function at the i-th iteration. The iterative coloring process may be represented by Algorithm (see also Fig. 2).

How will this IICP behave at infinity? Below I prove that, starting from some iteration, all the following colorings will be pairwise isomorphic in the sense of the following definition.

Definition 1. *Two colorings Col_i and Col_j are isomorphic ($Col_i \sim Col_j$), if there exists a bijection $\phi\colon Pal_i \leftrightarrow Pal_j$ such that $\forall v \in V \ \phi(Col_i(v)) = Col_j(v)$.*

Algorithm. Infinite iterative vertices coloring process (IICP)

$Pal_0 = 0$;

$\forall v \in V \ Col_0(v) := 0$;

$\forall v \in V \ Port_0(v) := (k(v))$, where $k(v) = deg(v)$;

$i := 1$;

repeat

    let χ_i be some indexing bijection $\chi_i \colon Port_{i-1}[V] \leftrightarrow \overline{1, |Port_{i-1}[V]|}$, then the current palette $Pal_i := \chi_i[Port_{i-1}[V]]$ (consists of the indices of the elements of $Port_{i-1}[V]$);

    let $K_i := |Pal_i|$;

    build a new "neighbourhood portrait" of each vertex:
$$Port_i(v) := \left(k_1^i(v), \ldots, k_j^i(v), \ldots, k_{K_i}^i(v)\right),$$
where $k_j^i(v)$ is the number of adjacent vertices of $v \in V$ that possess the color $j \in Pal_i$;
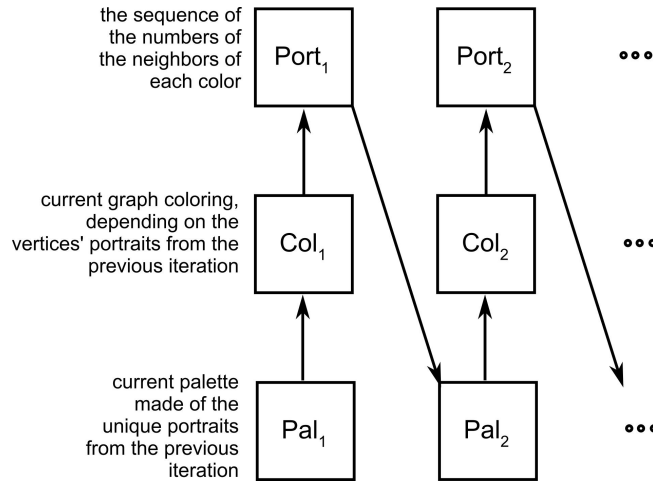
    $i := i + 1$;

until *False.*

Fig. 2. Scheme of infinite iterative coloring

Lemma 1 makes the first step in this direction. It shows that two consequent isomorphic colorings result in the degeneration of the following IICP.

Lemma 1. *If $Col_{L-1} \sim Col_L$ for some L, then all the following colorings will be pairwise isomorphic: $\forall L' \geq L \ \forall L'' \geq L \ \ Col_{L'} \sim Col_{L''}$.*

Proof. It is enough to prove that $Col_L \sim Col_{L+1}$. Let us select an arbitrary $v \in V$ and take the portrait of v from the previous iteration, which corresponds to $Col_{L+1}(v)$:

$$\left(k_1^L(v), \ldots, k_{K_L}^L(v)\right) = \chi_{L+1}^{-1}(Col_{L+1}(v)). \tag{1}$$

Let us consider the mapping $\psi \colon Port_{L-1}[V] \to Port_L[V]$:

$$\psi\left(\left(k_1^{L-1}(v), \ldots, k_{K_{L-1}}^{L-1}(v)\right)\right) = \left(k_1^L(v), \ldots, k_{K_L}^L(v)\right) \tag{2}$$

and prove that it is a bijection.

---

1) ψ is defined for all elements of $Port_{L-1}$, since it is defined for all $v \in V$.

2) Each portrait p from $Port_L$ corresponds to at least one vertex $v' \in V$ at the L-th iteration; selecting this v' in (2), we get at least one preimage for p in $Port_{L-1}$, which means that ψ is surjective.

3) The single-valuedness of ψ is slightly more complicated to prove; suppose there is a portrait from $Port_{L-1}$

$$p = \left(k_1^{L-1}(v_1), \ldots, k_{K_{L-1}}^{L-1}(v_1) \right) = \left(k_1^{L-1}(v_2), \ldots, k_{K_{L-1}}^{L-1}(v_2) \right) \tag{3}$$

that is mapped by ψ into two portraits from $Port_L$

$$p_1 = \left(k_1^L(v_1), \ldots, k_{K_L}^L(v_1) \right) \text{ and } p_2 = \left(k_1^L(v_2), \ldots, k_{K_L}^L(v_2) \right).$$

Since $Col_{L-1} \sim Col_L$, there exists an isomorphism $\varphi \colon Pal_{L-1} \leftrightarrow Pal_L$ such that

$$\forall v \in V \quad \forall i \in \overline{1, K_L} \quad k_i^{L-1}(v) = k_{\varphi(i)}^L(v). \tag{4}$$

Considering this, we can write

$$\forall i \in \overline{1, K_L} \quad k_i^L(v_1) \overset{(4)}{=} k_{\varphi^{-1}(i)}^{L-1}(v_1) \overset{(3)}{=} k_{\varphi^{-1}(i)}^{L-1}(v_2) \overset{(4)}{=} k_{\varphi(i)}^L(v_2),$$

which means that p_1 and p_2 are equal elementwise and proves the single-valuedness of ψ.

4) The injectiveness of ψ may be proved in the same manner; suppose there are two portraits from $Port_{L-1}$

$$p_1 = \left(k_1^{L-1}(v_1), \ldots, k_{K_{L-1}}^{L-1}(v_1) \right) \text{ and } p_2 = \left(k_1^{L-1}(v_2), \ldots, k_{K_{L-1}}^{L-1}(v_2) \right)$$

that are mapped by ψ into one portrait from $Port_L$

$$\left(k_1^L(v_1), \ldots, k_{K_L}^L(v_1) \right) = \left(k_1^L(v_2), \ldots, k_{K_L}^L(v_2) \right). \tag{5}$$

Returning to (4), we have

$$\forall i \in \overline{1, K_L} \quad k_i^{L-1}(v_1) \overset{(4)}{=} k_{\varphi(i)}^L(v_1) \overset{(5)}{=} k_{\varphi(i)}^L(v_2) \overset{(4)}{=} k_i^{L-1}(v_2),$$

which proves that ψ is injective and therefore is a bijection.

Applying ψ^{-1} to the right-hand side of (1), we get an element of $Port_{L-1}$, which can be mapped further to a color from Pal_L by χ_L. The final sought-after mapping $\phi \colon Pal_L \leftrightarrow Pal_{L+1}$ is the composition of the three above-mentioned mappings,

$$\phi = \chi_{L+1} \circ \psi \circ \chi_L^{-1},$$

which is a bijection since all the composing mappings are bijections.

$\square$

At this point, we know that two consequent isomorphic colorings make the further coloring process degenerated, but is this situation impossible, probable or inevitable? The answer to this question starts with the analysis of the behavior of the palette size. The number of colors in the palettes evidently belongs to $\overline{1, |V|}$, and intuitively this number either increases or remains constant during the IICP.

Lemma 2. *The size of the palette never decreases:* $\forall i \in \mathbb{N} \; \forall v_1, v_2 \in V \; (Col_{i+1}(v_1) = Col_{i+1}(v_2)) \Rightarrow (Col_i(v_1) = Col_i(v_2))$.

Proof. Let us use induction on i. The base case for $i = 0$ is true since all the vertices have the same color at the first iteration: $\forall v \in V \; Col_0(v) = 0$.

Inductive step: assume the statement is true for all $i < L$ and prove it for $i = L$. Let $V_1 = \{v_1^1, \ldots, v_1^n\}$ be the adjacent vertices of v_1 and $V_2 = \{v_2^1, \ldots, v_2^m\}$ be the adjacent vertices of v_2 in G.

Since $Col_{i+1}(v_1) = Col_{i+1}(v_2)$ and χ_{L+1} is a bijection,

$$Port_L(v_1) = \chi_{L+1}^{-1}(Col_{i+1}(v_1)) = \chi_{L+1}^{-1}(Col_{L+1}(v_2)) = Port_L(v_2),$$

which means that $n = m$ and there exists a bijection $\psi \colon V_1 \leftrightarrow V_2$, such that

$$\forall j \in \overline{1, n} \;\; Col_L(\psi(v_1^j)) = Col_L(v_1^j).$$

By induction,

$$\forall j \in \overline{1, n} \;\; (Col_L(\psi(v_1^j)) = Col_L(v_1^j)) \Rightarrow (Col_{L-1}(\psi(v_1^j)) = Col_{L-1}(v_1^j)),$$

which means that $Port_{L-1}(v_1) = Port_{L-1}(v_2)$ and thus

$$Col_L(v_1) = \chi_L(Port_{L-1}(v_1)) = \chi_L(Port_{L-1}(v_2)) = Col_L(v_2).$$

$\square$

Now we know that at each iteration the number of colors either increases or remains constant. We also know that the palette size cannot be arbitrary large, it is limited by $|V|$. Putting together both results, we get that an IICP may contain only a finite number of the steps where the palette size grows. In not more than $|V|$ iterations, an IICP will come to the situation, where the palettes at two consequent iterations have the same size. The consequences will be radical.

Lemma 3. *Either the size of the palette increases or the two last colorings are isomorphic:* $(K_i = K_{i+1}) \Rightarrow (Col_i \sim Col_{i+1})$.

Proof. I will prove that if $K_i = K_{i+1}$, then the mapping $\phi \colon Pal_i \rightrightarrows Pal_{i+1}$, defined as $\phi(Col_i(v)) = Col_{i+1}(v) \; \forall v \in V$, is a bijection. Let's argue by contradiction: suppose $K_i = K_{i+1}$, but ϕ is not a bijection. Since ϕ is defined for all $v \in V$, it is completely defined on Pal_i and is a surjection onto Pal_{i+1}; it is also injective by Lemma 2. The last option for ϕ not to be a bijection is multi-valuedness. Suppose that

$$\exists v', v'' \in V \colon (v' \neq v'') \;\&\; (Col_i(v') = Col_i(v'')) \;\&\; (Col_{i+1}(v') \neq Col_{i+1}(v'')). \quad (6)$$

Let $V_1 = \{v \in V \colon Col_i(v) = Col_i(v')\}$ and $V_2 = V \setminus V_1$ (see Fig. 3). The set V_2 is not empty, since otherwise

$$|Col_i[V_1]| + |Col_i[V_2]| = 1 + 0 = K_i = K_{i+1} \overset{(6)}{\geq} 2.$$

There are no colors from $Col_{i+1}[V_1]$ among the colors from $Col_{i+1}[V_2]$ since otherwise (see Fig. 3) $\exists v_1^* \in V_1, v_2^* \in V_2 \colon Col_{i+1}(v_1^*) = Col_{i+1}(v_2^*)$ but, by construction of V_1, V_2, $Col_i(v_1^*) = Col_i(v') \neq Col_i(v_2^*)$, which contradicts the injectiveness of ϕ. Thus,

$$Col_{i+1}[V_1] \cap Col_{i+1}[V_2] = \varnothing \;\; \text{and} \;\; K_{i+1} = |Col_{i+1}[V_1]| + |Col_{i+1}[V_2]|.$$

The number of distinct colors in $Col_{i+1}[V_2]$ is not less than $|Col_i[V_2]| = K_i - 1$ since ϕ is injective and does not map distinct colors into the same one.

By the definition of V_1 and assumption (6), we have $v', v'' \in V_1$ and

$$|Col_{i+1}[V_1]| \geq |\{Col_{i+1}(v'), Col_{i+1}(v'')\}| = 2,$$

which means that

$$K_{i+1} = |Col_{i+1}[V_1]| + |Col_{i+1}[V_2]| \geq 2 + (K_i - 1) = K_i + 1$$
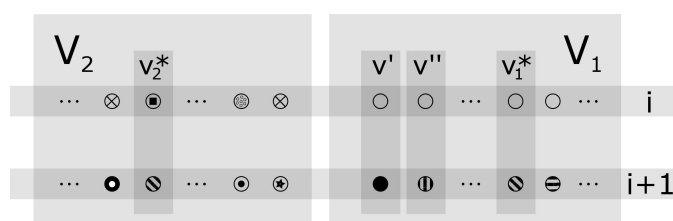
and contradicts the assumption $K_i = K_{i+1}$.

□

Fig. 3. Illustration for Lemma 3

The results of the previous lemmas may be summed up as a theorem.

Theorem 1. *The coloring process described in the Algorithm converges to an unchanging equitable partition in at most $|V|$ iterations.*

Edge coloring. Until now, all objects in the CSS were defined through other objects by means of one-type connections. One of the important generalizations of such approach is the introduction of connections of different types. As far as we talk about *closed* semantic systems, the "type" of a connection should be defined within the system. By analogy to the objects defined through their neighbours, the connections could be defined through the objects, which they relate. In the considered graph model of CSS, it means that the edges, just as the vertices, are subjected to an iteration coloring process based on the portraits of neighbours. During such an expanded coloring process, the color of a vertex is defined by the colors of the adjacent vertices and incident edges, and the color of an edge is defined by the colors of its 2 incident vertices.

This iterative coloring process for vertices and edges may be converted to the vertices-only coloring case by the addition of "virtual" vertices corresponding to the edges (Fig. 4) assuming the colors for the vertices and for the edges are taken from the same palette and assigned in accordance with the same laws.

2. Discussion

In the present paper I showed that each CSS with a constant structure and discrete time ceases to change rather quickly. The future research could aim at getting the CSS model closer to the real world. The consideration of more complex unsynchronized dynamical structures will change, to all appearances, the further research methods from
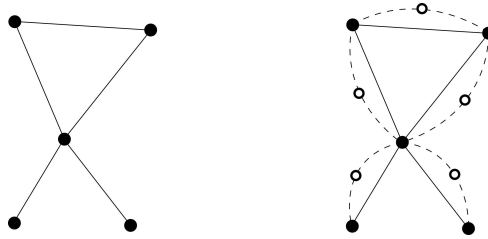
Fig. 4. Transformation of graph to consider the influence of edges

pure mathematical to statistical and computational. Below I concern several conceivable directions for the further investigations:

1. The first step towards real complex systems is to get rid of the discreteness of time. In the majority of real complex systems it seems unnatural to change the states of all elements simultaneously. Such desynchronization will mix the palettes, so the corresponding formal coloring process should be considerably different.

2. The second step is the consideration of the systems that start from an arbitrary state, not only the "zero state" of Algorithm, where $\forall v \in V \ Col_0(v) = 0$. It is easy to see that in this case Lemma 2 is not always true (see Fig. 5), therefore, Lemma 3 would also need a different proof since the current one relies on Lemma 2. Nevertheless, it seems the proof of convergence can be generalized to the case of arbitrary initial states.
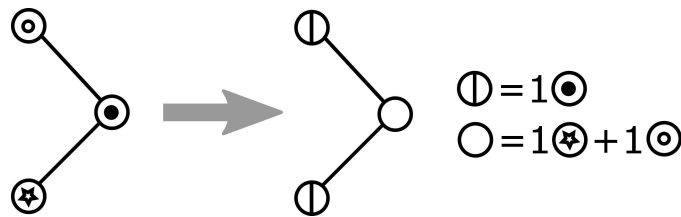

Fig. 5. Example of the palette's growth starting from a non-zero initial state

3. The most interesting research direction is the analysis of behavior of the CSSs with (stochastically) varying sets of objects and connections. Such CSSs could presumably be addressed by means of applied statistics and multi-agent modelling. In this respect, the underlying graph model could be replaced with the model of horizontal gene transfer [8] where each agent is "defined" through the influence of the agents encountered during stochastic motion.

References

1. Bloom P. *How Children Learn the Meanings of Words*. Cambridge, MIT Press, 2002.

2. Papadimitriou C.H. *Computational Complexity*, London, Pearson, 1993.

3. Godsil C.D. *Algebraic Combinatorics*. London, Chapman and Hall, 1993.

4. Unger S.H. GIT – a Heuristic Program for Testing Pairs of Directed Line Graphs for Isomorphism. *Communications of the ACM*, 1964, vol. 7, no. 1, pp. 26–34.

5. Weisfeiler B., Lehman A.A. [A Reduction of a Graph to a Canonical Form and an Algebra Arising During This Reduction]. *Nauchno-technicheskaya informatsia* [Scientific-Technical Information], 1968, vol. 2, no. 9, pp. 12–16. (in Russian)

6. Arlazarov V.L., Zuev I.I., Uskov A.V., Faradzhev I.A. An Algorithm for the Reduction of Finite Non-Oriented Graphs to Canonical Form. *USSR Computational Mathematics and Mathematical Physics*, 1974, vol. 14, no. 3, pp. 195–201.

7. McKay B.D., Piperno A. Practical Graph Isomorphism. *Journal of Symbolic Computation*, 2014, vol. 60, pp. 94–112.

8. Syvanen M. Horizontal Gene Transfer: Evidence and Possible Consequences. *Annual Review of Genetics*, 1994, vol. 28, no. 1, pp. 237–261.

ИТЕРАЦИОННОЕ РАВНОМЕРНОЕ РАЗБИЕНИЕ ГРАФА КАК МОДЕЛЬ ДИСКРЕТНОЙ ЗАМКНУТОЙ СЕМАНТИЧЕСКОЙ СИСТЕМЫ С ПОСТОЯННОЙ СТРУКТУРОЙ

Е.Е. Иванко

Институт математики и механики УрО РАН, г. Екатеринбург, Российская Федерация

Механико-машиностроительный институт, Уральский федеральный университет, г. Екатеринбург, Российская Федерация

Замкнутые семантические системы с постоянной структурой это системы, в которых каждый элемент определяется с помощью соответствующего ему фиксированного множества других элементов системы. Определения элементов изменяются итеративно и одновременно на основе «портретов соседей», полученных на предыдущей итерации. В настоящей статье автор рассматривает поведение подобных модельных систем, в которых процесс раскраски начинается с нулевого состояния, где все элементы идентичны. Изменение замкнутых семантических систем с постоянной структурой и дискретным временем может моделироваться как дискретный процесс раскраски на связном графе. В основном в статье рассматривается итерационный процесс переопределений только на вершинах, в предположении, что ребра являются не более, чем связями, не обладающими собственными цветами и не участвующими в процессе раскраски. Между тем, итерационный процесс одновременной раскраски вершин и ребер может быть сведен к процессу раскраски только вершин с помощью добавления виртуальных вершин, соответствующих ребрам при условии, что цвета для реальных и виртуальных вершин (ребер) выбираются из одного множества по одним правилам. В статье доказывается, что подобный итеративный процесс переопределений на основе цветов соседей быстро вырождается в последовательность попарно изоморфных состояний, а также обсуждаются возможные направления дальнейших исследований.

Ключевые слова: замкнутая семантическая система; граф; изоморфизм.

Литература

1. Bloom, P. How Children Learn the Meanings of Words / P. Bloom. – Cambridge: MIT Press, 2002.

2. Papadimitriou, C.H. Computational Complexity / C.H. Papadimitriou. – London: Pearson, 1993.

3. Godsil, C.D. Algebraic Combinatorics / C.D. Godsil. – London: Chapman and Hall, 1993.

4. Unger, S.H. GIT – a Heuristic Program for Testing Pairs of Directed Line Graphs for Isomorphism / S.H. Unger // Communications of the ACM. – 1964. – V. 7, № 1. – P. 26–34.

5. Вейсфейлер, Б.Ю. Сведение графа к канонической форме и алгебра, возникающая при этом сведении / Б.Ю. Вейсфейлер, А.А. Леман // Научно-техническая информация. – 1968. – Т. 2, № 9. – С. 12–16.

6. Арлазаров, В.Л. Алгоритм приведения конечных неориентированных графов к каноническому виду / В.Л. Арлазаров, И.И. Зуев, А.В. Усков, И.А. Фараджев // Журнал вычислительной математики и математической физики. – 1974. – Т. 14, № 3. – С. 195–201.

7. McKay, B.D. Practical Graph Isomorphism / B.D. McKay, A. Piperno // Journal of Symbolic Computation. – 2014. – № 60. – P. 94–112.

8. Syvanen, M. Horizontal Gene Transfer: Evidence and Possible Consequences / M. Syvanen // Annual Review of Genetics. – 1994. – V. 28, № 1. – P. 237–261.

Евгений Евгеньевич Иванко, доктор физико-математических наук, ведущий научный сотрудник, отдел управляемых систем, Институт математики и механики УрО РАН; ведущий научный сотрудник, лаборатория оптимального раскроя промышленных материалов и оптимальных маршрутных технологий, Механико-машиностроительный институт, Уральский федеральный университет (г. Екатеринбург, Российская Федерация), viatore@ya.ru.

34

Bulletin of the South Ural State University. Ser. Mathematical Modelling, Programming
& Computer Software (Bulletin SUSU MMCS), 2017, vol. 10, no. 4, pp. 26–34