Юрий Алексеевич ЕГОРОВ[1]
Марина Сергеевна ВОРОБЬЁВА[2]
Артём Максимович ВОРОБЬЁВ[3]

УДК 004.021

АЛГОРИТМ FDET ДЛЯ ПОСТРОЕНИЯ ПРОСТРАНСТВА ПРИЗНАКОВ КЛАССИФИКАЦИИ СЛОЖНЫХ ОБЪЕКТОВ В РАМКАХ ГРАФОВОЙ МОДЕЛИ

[1] аспирант,
Тюменский государственный университет
y.a.egorov@utmn.ru

[2] кандидат технических наук, доцент
кафедры программного обеспечения,
Тюменский государственный университет
m.s.vorobeva@utmn.ru

[3] старший преподаватель
кафедры программного обеспечения,
Тюменский государственный университет
a.m.vorobev@utmn.ru

Аннотация

В статье рассматривается графовая модель для классификации объектов сложной структуры. В рамках данной модели рассматривается алгоритм gBoost, осуществляющий решение задачи классификации. Решением задачи классификации является множество признаков, значимых для классификации объектов заданной обучающей выборки. Каждый признак представляет собой подграф, входящий хотя бы в один граф обучающей выборки, наличие или отсутствие которого позволяет отнести объект к тому или иному классу.

Цитирование: Егоров Ю. А. Алгоритм FDET для построения пространства признаков классификации сложных объектов в рамках графовой модели / Ю. А. Егоров, М. С. Воробьёва, А. М. Воробьёв // Вестник Тюменского государственного университета. Физико-математическое моделирование. Нефть, газ, энергетика. 2017. Том 3. № 3. С. 125-134. DOI: 10.21684/2411-7978-2017-3-3-125-134

Для построения пространства признаков классификации предложен алгоритм FDET. Входными данными алгоритма являются графы обучающей выборки, выходными — дерево подграфов, в каждом узле которого находится уникальный элемент пространства признаков классификации. В статье приводятся ограничения, накладываемые на входные данные, описание алгоритма и его вычислительная сложность.

Разработанный алгоритм был апробирован для решения задачи классификации открытых образовательных курсов по прикладной геологии и нефтегазовому делу.

Ключевые слова

Дерево подграфов, решение задачи классификации, графовая модель, объекты сложной структуры.

DOI: 10.21684/2411-7978-2017-3-3-125-134

Введение

Для решения различных задач науки, промышленности и бизнеса необходимо проводить анализ сложных объектов, состоящих из компонент, между которыми установлены отношения и связи. При этом и компоненты объектов, и отношения между компонентами могут обладать некоторыми свойствами. Такими объектами могут быть: банковские транзакции, web-сайты, транспортные сети, керны, сети белковых взаимодействий, социальные сети.

Универсальной структурой данных, с помощью которой можно представить такие объекты в виде, позволяющем проводить их анализ, являются графы. Например, используя решение задачи изоморфизма подграфов для анализа web-сайтов, можно разработать оптимальную структуру web-сайта, упрощающую его эффективное продвижение [5].

Анализ объектов, представленных в виде графов, позволяет получить дополнительную информацию и найти способ более эффективного использования как статичных, так и динамичных объектов сложной структуры [3]. В частности, разрабатываются математические модели и алгоритмы для решения задачи классификации объектов, представленных в виде графов, — поиска признаков, по которым объекты можно отнести к тому или иному классу.

Ввиду сложной природы анализируемых объектов также необходимо разрабатывать вспомогательные алгоритмы. В частности, в данной работе предлагается алгоритм DFET (Depth First Enumeration Tree) для построения пространства признаков классификации, среди которых осуществляется поиск решения задачи классификации графов.

Графовая модель для классификации объектов сложной структуры

Каждый граф является помеченным связным графом, который представляется в виде кортежа $G = \langle V, E, \mathcal{L}, l \rangle$, где V — множество вершин, $E \subseteq V \times V$ — множество ребер, $\mathcal{L}$ — множество меток вершин и ребер, $l: V \cup E \to \mathcal{L}$ — отобра-

жение множества вершин и ребер на множество меток, такое, что между двумя любыми вершинами $v_i \in V$, $v_j \in V$ существует путь $\mathcal{P}_{ij} \subseteq E$.

Даны множество графов обучающей выборки *X*, множество меток классов *Y* = {-1,1} и обучающая выборка $\mathcal{X} = \{\langle G_i, y_i \rangle\}_{i=1}^{n}$, где $G_i \in X$ — граф обучающей выборки, $y_i \in Y$ — один из двух непересекающихся классов, которому принадлежит граф G_i. Необходимо построить правило классификации $f: X \to Y$, способное классифицировать произвольный объект $G \in X$.

Для решения поставленной задачи используется алгоритм gBoost [5], разработанный X. Сайго с соавторами и позволяющий находить правила классификации графов на основе обучающей выборки.

Для работы алгоритма gBoost необходимо пространство признаков, среди которых можно найти признаки, значимые для классификации объектов заданной обучающей выборки. В заданном пространстве каждый признак представляет собой подграф, входящий хотя бы в один граф обучающей выборки, наличие или отсутствие которого позволяет отнести объект к тому или иному классу.

Алгоритм FDET

Дано множество графов обучающей выборки $G_i = \langle V_i, E_i, \mathcal{L}_i, l_i \rangle \in X$. Необходимо найти все подграфы, изоморфные хотя бы одному графу $G_i \in X$, и получить множество подграфов $\mathcal{T}$.

Для решения поставленной задачи необходимо ввести следующие утверждения.

Утверждение 1. Ребром графа называется кортеж $e = \langle a, b, l_a, l_{ab}, l_b \rangle \in \in V \times V \times \mathcal{L}_V \times \mathcal{L}_E \times \mathcal{L}_V$, где *a* — индекс выходной вершины, *b* — индекс входной вершины, l_a — метка выходной вершины, l_{ab} — метка ребра, l_b — метка входной вершины. Для краткости ребро будет записываться как кортеж $e = \langle a, b \rangle$.

Утверждение 2. Два ребра $e_1 \in E$ и $e_2 \in E$ в неориентированном графе *G*, где $e_1 = \langle a_1, b_1 \rangle$, $e_2 = \langle a_2, b_2 \rangle$, считаются эквивалентными, если выполнено условие $a_1 = b_2$ и $b_1 = a_2$.

При перечислении ребер неориентированного графа к каждому ребру $e = \langle a, b \rangle \in E$ добавляется эквивалентное ребро $e = \langle b, a \rangle \in E$.

Утверждение 3. Пусть *V* — множество всех вершин, входящих хотя бы в один граф обучающей выборки $G = \langle V, E, \mathcal{L}, l \rangle \in X$. Зададим линейный порядок $\prec_V$ для вершин из множества *V*. Пусть для $\forall v_1, v_2 \in V$ выполняется $v_1 \prec_V v_2$, если $l(v_1) \leq l(v_2)$.

Утверждение 4. Пусть *E* — множество всех ребер *e* = *E*, входящих хотя бы в один граф обучающей выборки $G = \langle V, E, \mathcal{L}, l \rangle \in X$. Зададим линейный порядок $\prec_E$ для ребер из множества *E*. Пусть для $\forall e_1, e_2 \in E$ выполняется $e_1 \prec_E e_2$, если:

1) $l(a_1) < l(b_2)$;
2) $l(a_1) = l(a_2)$ и $l(b_1) \leq l(b_2)$.

Если $G = \langle V, E, \mathcal{L}, l \rangle \in X$ является ненаправленным графом, то для того, чтобы применить алгоритм перечисления подграфов, необходимо привести граф

G к форме $\tilde{G} = \langle \tilde{V}, \tilde{E}, \mathcal{L}, l \rangle$, где $\tilde{E}$ — множество всех ребер, из которого исключены эквивалентные ребра по правилу:

1) если для двух эквивалентных ребер e_1 и e_2 выполняется $e_1 \prec_E e_2$, то $e_1 \in \tilde{E}$ и $e_2 \notin \tilde{E}$;

2) если для двух эквивалентных ребер e_1 и e_2 выполняется $e_2 \prec_E e_1$, то $e_1 \notin \tilde{E}$ и $e_2 \in \tilde{E}$.

В случае, когда *G* является направленным графом, приводить его к форме $\tilde{G} = \langle \tilde{V}, \tilde{E}, \mathcal{L}, l \rangle$ нет необходимости, и далее под графом $\tilde{G}$ подразумевается либо ненаправленный граф, в котором исключены эквивалентные ребра, либо направленный граф.

Для перечисления всех подграфов, входящих хотя бы в один граф из множества *X*, строится дерево подграфов *T*, являющееся выходными данными алгоритма и удовлетворяющее следующим условиям:

1) каждый узел $t \in T$ может иметь неограниченное количество потомков;

2) каждый узел $t \in T$ является уникальным изоморфным подграфом, входящим хотя бы в один граф из *X*;

3) каждый дочерний узел $t' \in T$ является суперграфом для родительского узла $t \in T$;

4) каждый родительский узел $t \in T$ является изоморфным подграфом дочерних узлов $t' \in T$;

5) все подграфы, принадлежащие одному уровню дерева *T*, имеют одинаковое количество ребер;

6) высота дерева *T* не превышает $N + 1$, где $N = \max_{G_n \in X} |E_i|$;

7) корень дерева — всегда пустой граф, который является изоморфным подграфом для любого суперграфа.

На рис. 1 представлен пример графа и соответствующего ему дерева подграфов *T*.

Алгоритм перечисления подграфов

1. Преобразовать граф *G* к виду $\tilde{G}$.
2. Инициализировать дерево подграфов *T* с корнем — пустым подграфом.
3. Для каждой вершины $\tilde{v} \in \tilde{V}$:
   a. Пометить текущую вершину $\tilde{v}$ как рассмотренную.
   b. Создать подграф $\tilde{G}'$, состоящий из одной вершины $\tilde{v}$, и добавить в дерево *T* как дочернюю вершину *t′* корня *t*.
   c. Узел *t′* становится текущим узлом.
   d. Заполняется *q* — очередь ребер, инцидентных вершине $\tilde{v}$.
   e. Выполнить обход ребер из очереди *q*.
4. Вернуть результат.

С помощью обхода ребер из очереди *q* осуществляется перечисление всех подграфов, содержащих текущую вершину $\tilde{v}$ и не содержащих вершин, помеченных как просмотренные. *T* — ссылка на текущий узел *t′* дерева, *q* — список ребер, которые потенциально могут быть добавлены в новые подграфы.

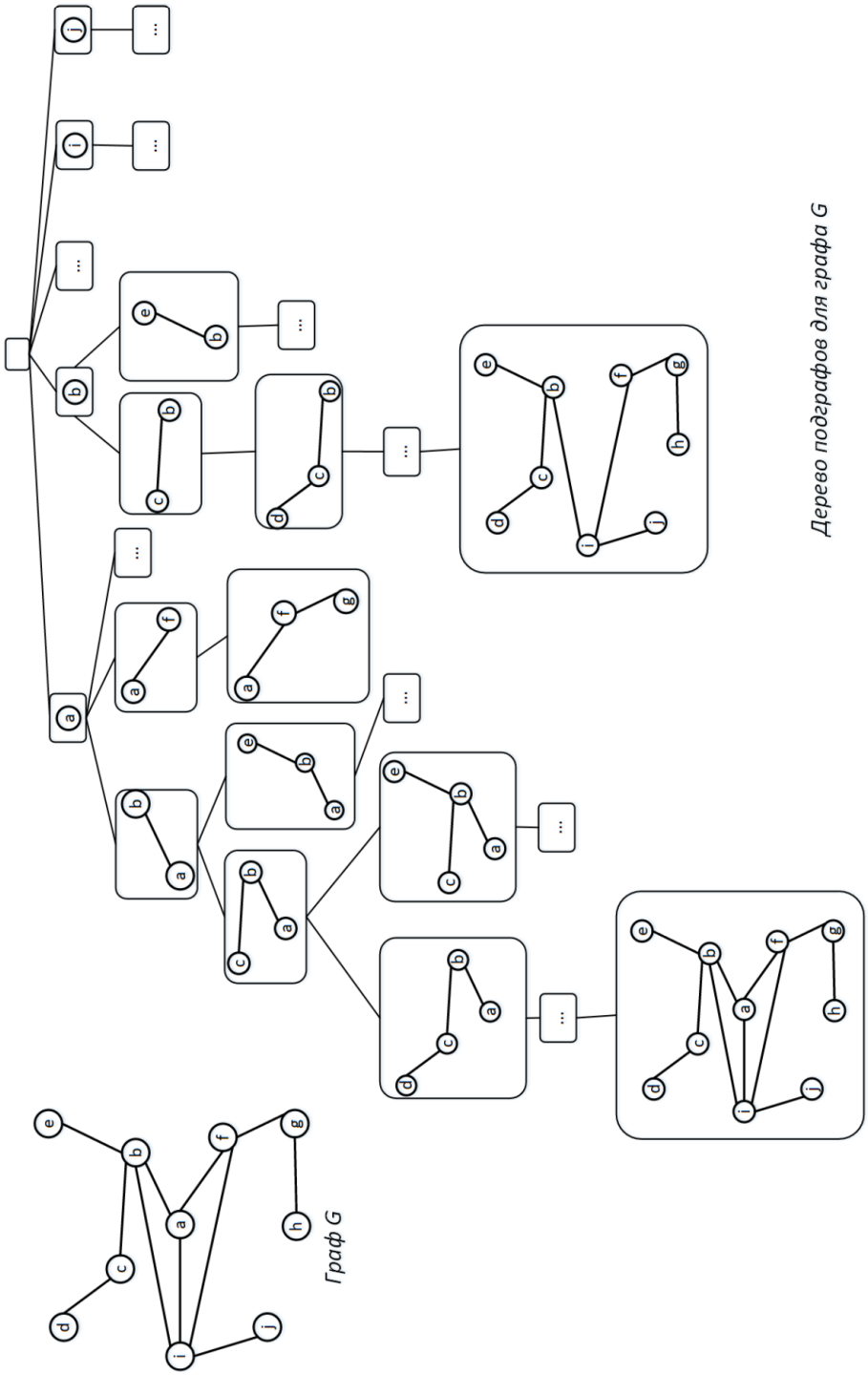


Рис. 1. Пример графа *G* и соответствующего дерева подграфов

Fig. 1. The example of graph *G* and corresponding subgraphs tree

Обход ребер из очереди q

1. Если очередь q пуста, завершить обход.
2. Извлечь из очереди q ребро $\tilde{e} = \langle a, b \rangle$.
3. Если граф в текущем узле t не содержит вершины b:
   a. Получить множество $\tilde{E}^*$ всех ребер, инцидентных вершине b (кроме ребра $\tilde{e}$ и ребер из очереди q) и не содержащих вершин, помеченных как рассмотренные.
   b. $q^* \leftarrow q \cup \tilde{E}^*$ — новая очередь ребер, которые могут быть добавлены в новые подграфы.
4. Создается новый узел $t' \leftarrow t \cup \tilde{e}$.
5. Если среди потомков узла t нет изоморфных узлу t' графов, узел t' добавляется как дочерний узел t.
6. Выполнить обход ребер, где T' — текущая вершина, копия q^* — очередь ребер.
7. Выполнить обход ребер, где T — текущая вершина, q^* — очередь ребер.

На рис. 2 представлена блок-схема алгоритма поиска подграфов.

Апробация алгоритма FDET

Разработанный алгоритм FDET был апробирован для решения задачи классификации открытых образовательных курсов по прикладной геологии и нефтегазовому делу.

Для решения данной задачи разработан программный продукт, представляющий собой приложение, разработанное с помощью следующих языков и инструментов:

— язык программирования Java 8;
— cplex — библиотека, разработанная на языке Java компанией IBM и предназначенная для решения задачи оптимизации;
— Jackson — библиотека для сериализации и десериализации объектов Java в формат json.

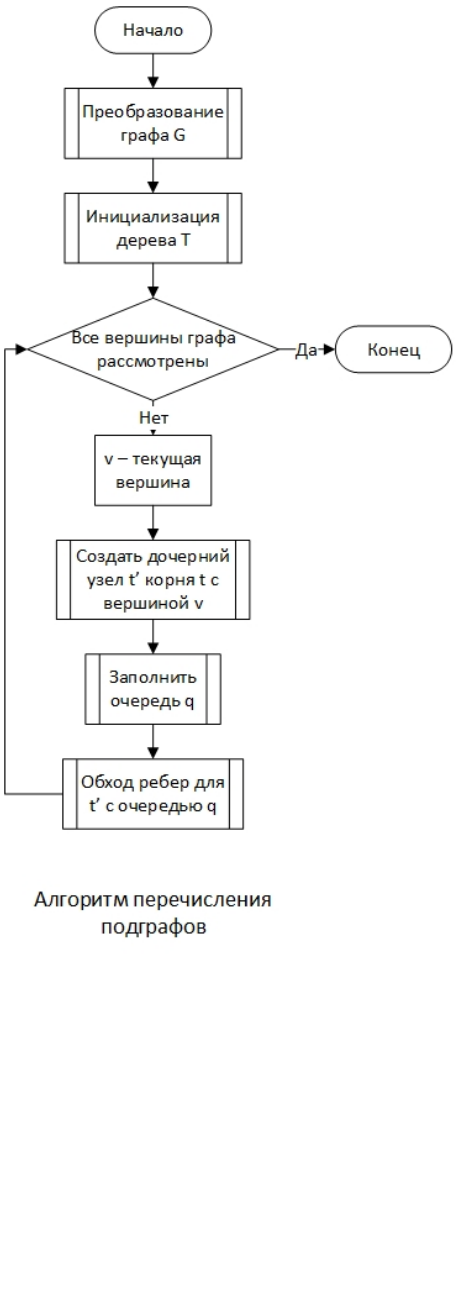
Для тестирования приложения была использована тестовая обучающая выборка, состоящая из 500 графов образовательных курсов и разбитая на два непересекающихся класса, в каждом из которых присутствовало 50% исследуемых объектов.

Тестирование разработанного решения проводилось с помощью кросс-проверки. В каждом тесте ошибка классификации вычислялась по формуле

$$error = \frac{|\mathcal{X}_{cont}^{err}|}{|\mathcal{X}_{cont}|},$$

где $|\mathcal{X}_{cont}^{err}|$ — количество элементов контрольной выборки, классифицированных ошибочно, $|\mathcal{X}_{cont}|$ — общее количество элементов контрольной выборки.

В результате было получено итоговое среднее значение ошибки классификации, равное 0,022.



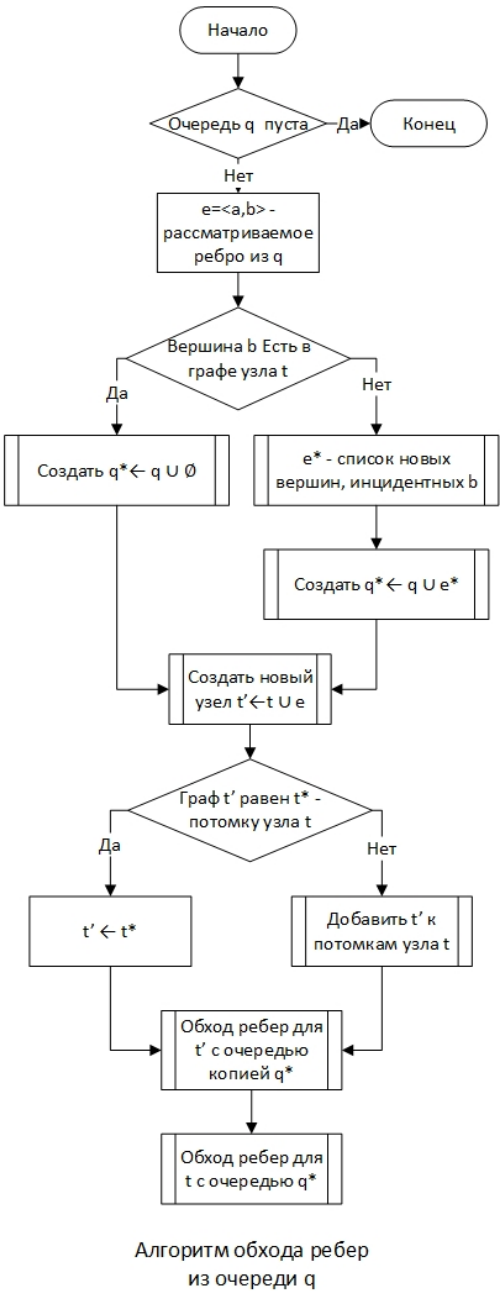


Рис. 2. Алгоритм поиска подграфов

Fig. 2. The subgraphs searching algorithm

Заключение

В ходе работы была предложена графовая модель сложных объектов и рассмотрено решение задачи классификации сложных объектов, представленных с помощью графов. Предложен алгоритм FDET для генерации пространства признаков классификации, элементами которого являются подграфы, входящие хотя бы в один граф обучающей выборки. Предложенное решение имеет следующие особенности.

Предложенное решение является универсальным и применимо для классификации любых объектов, представимых в виде графов.

Решение является масштабируемым, так как возможна частичная или полная параллельная реализация предложенных алгоритмов [5].

Вычислительная сложность разработанного алгоритма FDET $O(2^M)$, где M — количество ребер в графе, следовательно, при решении задачи классификации с помощью алгоритмов gBoost [5] и FDET необходимо использовать дополнительные критерии, позволяющие ограничить пространство признаков.

СПИСОК ЛИТЕРАТУРЫ

1. Воробьева М. С. Построение распределенного алгоритма поиска структурных различий в категориях изоморфизма / М. С. Воробьева, А. М. Воробьев, Ю. А. Егоров // Международный научно-исследовательский журнал. 2017. № 4 (58). Часть 4. С. 24-28.
2. Егоров Ю. А. Модификация алгоритма Ульмана для многопроцессорных систем / Ю. А. Егоров // Материалы XVII Всероссийской конференции молодых ученых по математическому моделированию и информационным технологиям, г. Новосибирск / Новосибирск: ИВТ СО РАН, 2016. С. 86-87.
3. Захарова И. Г. Алгоритм поиска минимального пути в графе с динамически изменяющимися весами / И. Г. Захарова, И. А. Муравьев // Математическое и информационное моделирование. Издательство ТюмГУ, 2015. С. 173-179.
4. Demiriz A. Linear Programming Boosting via Column Generation / A. Demiriz, K. P. Bennett, J. Shawe-Taylor // Machine Learning. 2002. Vol. 46. Pp. 225-254. DOI: 10.1023/A:1012470815092
5. Saigo H. gBoost: A Mathematical Programming Approach to Graph Classification and Regression / H. Saigo, S. Nowozin, T. Kadowaki // Machine Learning. 2009. Vol. 75. Pp. 69-89. DOI: 10.1007/s10994-008-5089-z

Yurij A. EGOROV[1]
Marina S. VOROBYOVA[2]
Artem M. VOROBYOV[3]

FDET ALGORITHM FOR BUILDING SPACE OF CLASSIFICATION PATTERNS IN GRAPH MODEL

[1] Postgraduate Student,
University of Tyumen
y.a.egorov@utmn.ru

[2] Cand. Sci. (Tech.), Associate Professor,
Department of Software,
University of Tyumen
m.s.vorobeva@utmn.ru

[3] Senior Lecturer,
Department of Software,
University of Tyumen
a.m.vorobev@utmn.ru

Abstract

This paper considers the graph model of the complex objects classification. Within this model's framework the authors consider gBoost algorithm for solving the classification problem. A classification problem solution is a set of patterns which are valuable for classification of the training sample objects. A pattern is some subgraph included in at least one graph from the training sample and whose presence or absence allows to classify the object.

The authors propose FDET for classification patterns space building algorithm. The input data are graphs from the training sample. The output data is the subgraph tree with the unique classification patterns space element in each node. The paper provides the input data constraints, algorithm description and computational complicity.

The algorithm was developed and tested for solving the open courses in applied geology and oil and gas business classification problem.

Citation: Egorov Yu. A., Vorobyova M. S., Vorobyov A. M. 2017. "FDET Algorithm for Building Space of Classification Patterns in Graph Model". Tyumen State University Herald. Physical and Mathematical Modeling. Oil, Gas, Energy, vol. 3, no 3, pp. 125-134.
DOI: 10.21684/2411-7978-2017-3-3-125-134

Keywords

Subgraph tree, classification problem solution, graph model, complicated structure objects.

DOI: 10.21684/2411-7978-2017-3-3-125-134

REFERENCES

1. Vorobyova M. S., Vorobyov A. M., Egorov Yu. A. 2017. “Postroyeniye raspredelennogo algoritma poiska strukturnyh razlichiy v kategoriyah izomorfizma” [Construction of Distributed Algorithm for Structural Difference in Isomorphism Catecories]. Mezhdunarodniy nauchno-issledovatelskiy zhurnal [International Research Journal], no 4 (58), vol 4, pp. 24-28.
2. Egorov Yu. A. 2016. Modifikaciya algoritma Ulmana dlya mnogoprocessornyh system [Ullmann Algorithm Improvement for Multiprocessor Systems]. Proceedings of the 17th All-Russian Conference for Young Researchers “po matematicheskomu modelirovaniju i informacionnym tehnologiyam” [On Mathematical Modeling and Information Technologies], pp. 86-87.
3. Zakharova I. G., Muravyev I. A. 2015. “Algoritm poiska minimalnogo puti v grafe s dinamicheski izmenyayuzchimisya vesami” [The Algorithm for Finding the Minimum Path in a Graph with Dynamically Changing Weights]. In: TSU Publishing House. Matematicheskoe i informacionnoye modelirovaniye [Mathematical and Informational Modeling], pp. 173-179. Tyumen State University Publishing House.
4. Demiriz A., Bennett K. P., Shawe-Taylor J. 2002. “Linear Programming Boosting via Column Generation”. Machine Learning, vol. 46, pp. 225-254. DOI: 10.1023/A:1012470815092
5. Saigo H., Nowozin S., Kadowaki T. 2009. “gBoost: A Mathematical Programming Approach to Graph Classification and Regression”. Machine Learning, vol. 75, pp. 69-89. DOI: 10.1007/s10994-008-5089-z