Моделирование и анализ информационных систем. Т. 25, №4 (2018), с. 402–410
Modeling and Analysis of Information Systems. Vol. 25, No 4 (2018), pp. 402–410

Технология блокчейн

DOI: 10.18255/1818-1015-2018-4-402-410

УДК 51-37

О некоторых подходах к решению задачи «Useful Proof-of-work for blockchains»

Дурнев В. Г., Мурин Д. М., Соколов В. А., Чалый Д. Ю.

получена 30 июля 2018

Аннотация. Технология блокчейн основана на принципе доказательства работой «Proof-of-work». Суть данного принципа состоит в том, что некоторое событие (например, перевод денежных средств с одного счета на другой) становится значимым только после того, как оно подтверждено определенным объемом вычислительной работы. Соответственно возникает потребность в вычислительных задачах, над которыми такую работу можно производить, причем на решение этих задач будет тратиться практически вся вычислительная мощность блокчейн-сети. На сегодня в качестве таких задач получили распространение «хэш-головоломки» – задачи поиска битовой строки с хэшем, удовлетворяющим определенным условиям. Существенным недостатком хэш-головоломок является отсутствие у них какого-либо полезного применения за пределами технологии блокчейн. В работе описываются подходы к решению задачи «Useful Proof-of-work for blockchains», а именно предлагается рассматривать в качестве вычислительных задач для доказательства работой возникающие на практике индивидуальные представители NP-полных задач, которые могут решаться, например, SAT- или LLL-решателями. Отдельной проработки требует вопрос об использовании FPT-задач. Предлагаемый подход позволяет обеспечить следующие свойства вычислительных задач для доказательства работой: полезность, управляемость сложностью задач (через изменение размерности, выбор задач определенного вида, указание точности необходимого решения), массовость. При этом допускается, что не каждая решенная задача может оказаться полезной, однако предоставляется возможность решать с помощью технологии блокчейн задачи, возникающие на практике. Кроме прочего, таким образом становится возможным сопоставить стоимость виртуальной криптовалюты через затраты электроэнергии при ее генерации с практическим результатом от решения вычислительных задач. Наиболее трудными вопросами в контексте рассматриваемого подхода являются реализация связи событий и задач, обеспечивающих эти события вычислительной работой, и реализация системы анализа сложности задач. Статью следует воспринимать как программу исследований, поскольку многие технические детали требуют отдельной проработки.

Ключевые слова: доказательство работой, блокчейн, выполнимость, SAT-решатель, NP-полнота, FPT, алгоритм

Для цитирования: Дурнев В. Г., Мурин Д. М., Соколов В. А., Чалый Д. Ю., "О некоторых подходах к решению задачи «Useful Proof-of-work for blockchains»", *Моделирование и анализ информационных систем*, **25**:4 (2018), 402–410.

Об авторах:
Дурнев Валерий Георгиевич, д-р физ.-мат. наук, профессор,
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: Durnev@uniyar.ac.ru

Мурин Дмитрий Михайлович, orcid.org/0000-0002-8068-0784, канд. физ.-мат. наук, доцент,
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: nirum87@mail.ru

Соколов Валерий Анатольевич, orcid.org/0000-0003-1427-4937, д-р физ.-мат. наук, профессор,
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: valery-sokolov@yandex.ru

Чалый Дмитрий Юрьевич, orcid.org/0000-0003-0553-7387, канд. физ.-мат. наук, доцент,
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: chaly@uniyar.ac.ru

Введение

С момента появления основополагающей статьи Сатоши Накомото [1], блокчейн является динамично развивающейся технологией [2], чему во многом способствовал взрывной рост популярности криптографических валют, созданных на основе этой технологии.

Непосредственно блокчейн представляет собой структуру криптографически связанных между собой данных, которые традиционно называются блоками. Блоки создаются на вычислительных устройствах, которые объединяются в блокчейн-сеть и содержат служебную информацию, в том числе ссылку на предыдущий блок, значение, получающееся в вершине дерева Меркла, в листах которого формируются значения на основе событий, привязываемых к этому блоку, а также специальное поле, предназначенное для размещения решения задачи специального вида.

Одним из основных примитивов при обеспечении безопасности в технологии блокчейн выступают криптографические хэш-функции. На их основе строятся так называемые «хэш-головоломки» – задачи поиска битовой строки с хэшем, удовлетворяющим определенным условиям. Решается хэш-головоломка следующим образом: для блока задаются значения всех полей за исключением специального, а значения специального поля перебираются (итерируются) до тех пор, пока значение хэш-функции от служебной информации блока не будет удовлетворять заданным условиям. Традиционное условие, которому должно удовлетворять значение хэш-функции, – это наличие определенного числа нулей в начале хэш-свертки. Задавая число нулей, можно управлять сложностью решаемой хэш-головоломки таким образом, чтобы число блоков, создаваемых в единицу времени, было в среднем одинаково.

Хэш-головоломки обладают следующими важными свойствами:

1) решение хэш-головоломки позволяет криптографически связать служебную информацию блока;

2) легкость создания;

3) управляемость сложностью.

К недостаткам хэш-головоломок относится отсутствие у них какого-либо полезного применения за пределами технологии блокчейн, скорее даже их полная бесполезность. Поэтому возникает потребность в создании технологии блокчейн, в которой для обеспечения работой используются полезные и интересные задачи.

Помимо хэш-головоломок существует большое число трудноразрешимых задач, представляющих с практической точки зрения существенно больший интерес. К таким задачам, безусловно, относятся NP-полные задачи, в том числе выполнимость

булевой формулы, на основе которой предложены наиболее эффективные на настоящий момент программные приложения для решения представителей NP-полных задач – CDSL SAT-решатели (SAT-solver) [3,4], задачи теории решеток, находящие свое применение в теории управления, радиосвязи и кристаллографии, задача о рюкзаке, находящая применение в логистике, и многие другие. Кроме того, есть целый класс FPT-задач с параметризованной сложностью.

Задача «Useful Proof-of-work for blockchains» сформулирована на международной студенческой олимпиаде по криптографии 2017 года [5] и на момент публикации статьи считается нерешенной [6]. Необходимо, однако, отметить, что в статье [7] предложен как минимум один из возможных вариантов решения задачи в поставленной на олимпиаде формулировке.

Итак, задача «Useful Proof-of-work for blockchains» состоит в создании задачи P, которая может быть использована для систем с доказательством работой таким образом, что информация, полученная в процессе решения, может быть использована за пределами системы. Более формально:

1. P – это семейство задач, параметризированных двумя переменными: I (входные данные, например, 256-битовая строка) и C (сложность, например, некоторое положительное целое число).

2. Для фиксированного входа и сложности $P(I, C)$ является проблемой, которая может быть решена с помощью некоторого алгоритма A. Невозможно найти доказуемое решение задачи $P(I, C)$, если I неизвестно.

3. Среднее время T (количество шагов или итераций вычислений), необходимое для нахождения решения $P(I, C)$ с использованием алгоритма A, известно (при условии, что входные данные I выбраны случайным образом и равномерно) и зависит от C, поэтому $T = T(C)$ и $T(C)$ могут быть сделаны очень маленькими, неимоверно большими или чем-то промежуточным, регулируя переменную сложности C.

4. Должно быть легко проверить, правильно ли выполнено любое предоставленное решение.

5. Является желательным любое доказательство того, что, вероятно, не существует значительно лучших алгоритмов для решения P, чем данный алгоритм A.

6. Необходимо описать, как информация, полученная в процессе решения P, может быть полезна вне системы доказательства работой.

Исходя из формулировки задачи, можно заключить, что приоритет при построении задачи P отдается все же тем свойствам, которые позволяют использовать эту задачу в качестве замены традиционных хэш-головоломок. При этом полезность предлагаемой задачи является косвенной, поскольку из существования I' и C', таких что $P(I', C')$ является практически полезной задачей, не следует, что доля таких параметров существенна, а также что выбор задач, обеспечивающих события вычислительной работой, будет в пользу этих параметров.

Мы предлагаем решать эту задачу, исходя из приоритета полезности решаемых задач. С этой точки зрения, ответы на вопросы 4–6 становятся очевидными. Если рассматривать в качестве P множество NP-полных задач, то частое появление их на практике дает ответ на вопрос 6. Для задач общего положения лучшие результаты в решении показывают на практике SAT-решатели, которые могут претендовать на роль алгоритма A. И, в силу принадлежности классу NP, решение задач легко

проверить на корректность. При этом технологии блокчейн, базирующиеся на NP-полных задачах, в доступной авторам литературе не обнаружены.

Остаются три первых вопроса, которые можно свести к двум задачам: первая состоит в том, чтобы связать решаемую задачу с блоком (в исходной формулировке за это отвечают входные данные I), вторая задача состоит в организации управления сложностью задач с целью обеспечения равномерной генерации блоков в условиях колебания вычислительной мощности блокчейн-сети (за это отвечает параметр сложности C, от которого зависит функция T).

1. Каким образом получать задачи, решение которых может оказаться полезным

Хорошо известен факт, что решение индивидуальных представителей NP-полных задач представляет существенную ценность как для практики, так и для развития некоторых теорий [8]. Ввиду того, что представители NP-полных задач достаточно часто возникают в различных областях знаний [9], мы считаем, что сбор задач, возникающих на практике, является более перспективным, чем попытка создания задач, решение которых только потенциально может найти практическое применение. Однако в случае нехватки гарантированно полезных задач допускается использование генераторов задач, «близких» к полезным, что, впрочем, приведет к вопросам о том, насколько много действительно полезных задач и, в связи с этим, насколько правомерна поставленная в работе цель.

Реализация сбора задач может быть организована по-разному. Мы видим два принципиальных варианта: либо задачи публикуются в самой цепочке блокчейна (в качестве смарт-контрактов или транзакций особого вида, связанных с одним из блоков), либо набор задач, имеющих практическое значение, создается вне цепочки блокчейна отдельным сервисом, создающим и поддерживающим базу данных задач. И в первом, и во втором случае должна формироваться база данных задач, позволяющая осуществлять выбор задач на основании оценки среднего времени для нахождения решения.

В первом случае, предполагая технологию блокчейн открытой, мы допускаем раскрытие задач, после чего они могут быть решены вне нашей блокчейн-сети, а решения впоследствии могут быть использованы для формирования блоков нашей блокчейн цепочки. Во втором случае такого раскрытия можно не допустить, но это приведет к вопросу о том, кто будет являться владельцем базы данных задач, и может привести к централизации управления, для устранения которого создавалась технология блокчейн. Мы считаем, что при большом числе задач, предназначенных для решения, их публикация не будет являться критичной по следующим причинам:

1) нецелесообразно тратить вычислительные ресурсы на решение всех задач подряд, в то время как их можно тратить на решение выбранных задач с целью получения премии за создание блока;

2) при большом числе задач вероятность выбрать для решения задачу, которая потом встретится в блокчейне, мала;

3) кроме решения задачи, необходимо также обеспечить привязку задачи к блоку и событиям, то есть нужно осуществить подбор событий таким образом, чтобы

значение хэш-функции от случайной информации блока являлось указателем на решенную задачу, что само по себе является трудной задачей.

Отметим, что в первом случае публикация задач в блокчейне может не преследовать цели обеспечения событий вычислительной работой. Например, за решение задач может предлагаться оплата.

Решенные задачи, после создания последующих блоков, могут удаляться из базы данных или помечаться как архивные, с публикацией задачи и решения и направлением решения поставившему задачу лицу или сервису.

Здесь необходимо сказать еще несколько слов о конфиденциальности. Многие полезные для практики представители NP-полных задач могут возникать в рамках коммерческих разработок. Сами задачи и их решения могут являться коммерческой тайной. Для закрытых (корпоративных) блокчейн проектов [10, 11] это не является существенной проблемой, однако невозможность использовать более широкие возможности открытых блокчейн проектов может вызывать некоторые неудобства. Поэтому мы хотим обратить внимание, что при нашем подходе возможно применение различных методов «маскирования» исходной задачи. К таким методам можно отнести, например, полиномиальное сведение одной задачи к другой, использование методов, близких к доказательствам с нулевым разглашением и другие. В качестве классического примера можно привести известную задачу о изоморфизме графа, используемую для иллюстрации доказательств с нулевым разглашением. Если у организации есть необходимость найти гамильтонов путь в большом графе, то она может обратиться к блокчейн-сети, предоставив ей запрос на решение этой задачи для графа, изоморфного тому, который представляет интерес. Изоморфный граф легко построить, осуществив перестановку вершин исходного графа. При этом исходный граф не будет опубликован.

2. Об управлении сложностью задач

По мере поступления задач и формирования базы данных необходимо обеспечить их классификацию, сведение к базовой задаче, например, выполнимости конъюнктивной нормальной формы, и в дальнейшем выбор задач для обеспечения событий вычислительной работой. Необходимость сведения к базовой задаче возникает для унификации решающего алгоритма A, в качестве которого мы в первую очередь рассматриваем SAT-решатели. Такое сведение позволяет применять один алгоритм ко всем индивидуальным представителям NP-полных задач. При этом естественно допустить другой подход: существование разных блокчейн проектов, которые при обеспечении событий вычислительной работой ориентируются на определенные задачи и не принимают к рассмотрению задачи другого вида. Этот подход потребует наличия источника полезных на практике задач определенного вида и применения к ним специализированного алгоритма, при этом подходе управление сложностью задач может быть более строгим.

Стоит обратить внимание на класс разрешимых с фиксированным параметром задач – класс FPT (Fixed-Parameter Tractable) [12, 13].

Определение 1. *Параметризированная задача* Π *принадлежит классу FPT, если*

она может быть решена некоторым параметризированным алгоритмом за время $t(n,k) = O(n^{O(1)} \cdot f(k))$ *для функции* f*, зависящей только от параметра* k.

Для FPT-задач естественно использовать параметр k для управления сложностью, при этом для фиксированного k FPT-задачи могут быть решены за полиномиальное время. Это позволяет рассматривать FPT-задачи в качестве хорошего примитива для обеспечения событий вычислительной работой.

Управление сложностью задач должно основываться на предварительной обработке множества доступных к решению задач. Полагая, что база данных задач сформирована и имеется определенная практика решения задач блокчейн-сетью, необходимо провести анализ задач по следующим направлениям:

1) классификация задачи (оптимизации, вычисления, распознавания; что собственно за задача представлена к решению, можем ли мы отнести задачу к какому-то особому виду);

2) размерности исходной задачи и базовой задачи, к которой была сведена исходная;

3) точность решения, которую следует обеспечить;

4) имеющаяся практика решения задачи блокчейн-сетью;

5) имеющаяся практика решения блокчейн-сетью близких по характеристикам задач.

Результатом анализа должно являться множество задач, которые блокчейн-сеть может решить за определенное в среднем время, зависящее от числа блоков, которые необходимо создавать за единицу времени.

Выбор задачи из данного множества может быть осуществлен, например, следующим образом: выбрав события, которые мы хотим обеспечить вычислительной работой, и сформировав блок, можно вычислить значение хэш-функции от служебной информации блока. Полученное таким образом значение хэш-функции может являться указателем на задачу, которую необходимо решить для обеспечения блока работой. Кроме прочего, таким образом может быть обеспечена связь между служебной информацией блока и информацией о задаче, обеспечивающей события, связанные с блоком, вычислительной работой.

Таким образом, мы переходим к следующему разделу, в котором обсудим вопрос связи событий и задач, обеспечивающих эти события вычислительной работой.

3. О связи событий и задач, обеспечивающих эти события вычислительной работой

В этом разделе мы опишем два способа, которыми можно связать служебную информацию блока, информацию о событиях и информацию о задаче, обеспечивающей события вычислительной работой.

Первый способ близок к традиционному блокчейну и использует хэш-головоломки. Мы можем использовать дерево Меркла событий и служебную информацию блока для выбора задачи из базы данных задач (например, как описано в предыдущем разделе). Далее можно включить выбранную задачу в дерево Меркла, а ее решение в служебную информацию блока, после чего в соответствии с традиционным блокчейном решить хэш-головоломку для блока в такой конфигурации.

Решение хэш-головоломки в этом случае решает традиционную задачу связывания служебной информации блока, информации о событиях и информации о задаче, обеспечивающей события вычислительной работой. Блок, полученный таким образом, можно назвать «тяжелым», поскольку на его формирование необходимо потратить больше вычислительных ресурсов за счет решения прикрепленной к блоку задачи, чем на обычный «легкий» блок. Поскольку на формирование тяжелого блока должно тратиться больше вычислительных ресурсов, то, естественно, возникает необходимость мотивировать участников блокчейн-сети к созданию таких блоков. Мы видим две основных возможности для этого: с одной стороны, можно увеличить вознаграждение за формирование тяжелых блоков, а с другой, можно, в отличие от традиционной блокчейн технологии, при которой участники всегда считают истинной самую длинную версию цепочки и работают над ее удлинением, считать истинной самую длинную версию цепочки, содержащей больше всего тяжелых блоков. Можно положить «вес» тяжелого блока равным трем «весам» легких блоков. Соответственно за генерацию тяжелого блока должно осуществляться трехкратное вознаграждение, и цепочка из одного тяжелого блока должна быть эквивалентна цепочке из трех легких блоков. Отметим, что на настоящий момент событие принимается как достоверное в блокчейне криптовалюты Биткойн после того, как будут созданы шесть блоков, следующие за блоком, к которому привязано событие. Таким образом, потенциальный отказ от трех легких блоков в пользу одного тяжелого не является, с нашей точки зрения, критичным для функционирования технологии блокчейн.

Соответственно тяжелый блок становится выгодно создавать, если затраты времени на его создание в среднем меньше времени, затрачиваемого на создание трех легких блоков. Следовательно, мы должны выбирать сложность задач таким образом, чтобы обеспечить их решение в течение времени, затрачиваемого в среднем на создание двух блоков. Кроме прочего, это дает дополнительный механизм для управления сложностью задач: при оценке сложности задач можно ввести дополнительный коэффициент, который указывал бы число раз, при которых задача выбиралась для решения, но решение которой не было получено из-за приращения альтернативной ветви блокчейна.

Второй способ отличен от традиционного блокчейна и предполагает, что при формировании блока в него должны быть записаны решение задачи, опубликованной в предыдущем блоке, и задача, решить которую требуется для формирования следующего блока. Выбор задачи, как и ранее, может осуществляться с помощью хэш-функции, значение которой на входе из служебной информации блока можно рассматривать как указатель на задачу (например, это может быть первая нерешенная задача в базе данных, k-битовый префикс которой совпадает с k-битовым префиксом значения хэш-функции). Следует обратить внимание, что во втором способе, как и в традиционном блокчейне, возможно формирование альтернативных цепочек по следующим причинам:

1) различный выбор задач участниками блокчейн-сети;

2) получение участниками блокчейн-сети различных решений одной и той же задачи.

В блоке может публиковаться не одна, а несколько задач для решения (например, первые пять нерешенных задач в базе данных, k-битовый префикс которых

совпадает с k-битовым префиксом значения хэш-функции), что породит еще одну возможность для создания альтернативной цепочки.

Как и в традиционном блокчейне, участники должны считать истинной самую длинную версию цепочки, «тяжелых» блоков в этом случае не возникает.

4. Заключение

В работе предложены подходы к решению задачи «Useful Proof-of-work for blockchains», которые базируются на полезности решаемой задачи. Данная постановка приводит к определенным трудностям в вопросе связывания в одном блоке служебной информации, информации о событиях и информации о задаче, обеспечивающей события вычислительной работой. В статье дается описание способов преодоления этих трудностей. В качестве вычислительных задач для доказательства работой предлагается рассматривать индивидуальные представители NP-полных задач (альтернативным вариантом может выступать рассмотрение FPT-задач). Задачи, полезность которых обеспечивается возникновением на практике, должны передаваться в специальную базу данных. Выбор задач для решения должен осуществляться в соответствии с описанными в статье подходами. В качестве алгоритмов их решения рассматриваются SAT-решатели. Управление сложностью задач предлагается реализовывать через изменение размерности, выбор задач определенного вида, указание точности необходимого решения, анализ опыта блокчейн-сети по решению задач со схожими параметрами. Многие вопросы, затронутые в статье, требуют проведения экспериментальных исследований, запланированных авторами.

Список литературы / References

[1] Nakamoto S., “Bitcoin: A Peer-to-Peer Electronic Cash System”, 2009, 1–9. https://bitcoin.org/bitcoin.pdf.

[2] Buterin V., “Ethereum White Paper: A next-generation smart contract and decentralized application platform”, 2014. https://github.com/ethereum/wiki/wiki/White-Paper.

[3] Marques-Silva J.Ṗ., Sakallah K.Ȧ., “GRASP: A new search algorithm for satisfiability”, *ICCAD ’96 Proceedings of the 1996 IEEE/ACM international conference on Computer-aided design*, 1996, 220—227.

[4] Bayardo Jr.Ṙ., Schrag R., “Using CSP look-back techniques to solve real-world SAT instances”, *AAAI’97/IAAI’97 Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence*, 1997, 203—208.

[5] “International Students’ Olympiad in Cryptography NSUCRYPTO. Useful Proof-of-work for blockchains”, 2017, 12–13. https://nsucrypto.nsu.ru/archive/2017/round/2/section/0/task/11.

[6] “International Students’ Olympiad in Cryptography NSUCRYPTO. Unsolved problems”, 2018. https://nsucrypto.nsu.ru/unsolved-problems/.

[7] Ball M., Rosen A., Sabin M., Vasudevan P. N., *Proofs of Useful Work*, 2017. https://eprint.iacr.org/2017/203.pdf.

[8] Garey M. R., Johnson D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co, San Francisco, Calif., 1979.

[9] Cormen T. H., Leiserson C. E., Rivest R. L., Stein C., *Introduction to Algorithms (third ed.)*, MIT Press, 2009.

[10] BitFury Group, "Public versus Private Blockchains. Part 1: Permissioned Blockchains. White Paper", 2015, 1–23 (совм. с Garzik J.). https://bitfury.com/content/downloads/public-vs-private-pt1-1.pdf.

[11] BitFury Group, "Public versus Private Blockchains. Part 2: Permissionless Blockchains. White Paper", 2015, 1–20 (совм. с Garzik J.). https://bitfury.com/content/downloads/public-vs-private-pt2-1.pdf.

[12] Downey R., Fellows M., *Parameterized complexity*, Springer Verlag, New York, 1999.

[13] Flum J., Grohe M., *Parameterized complexity theory*, Springer Verlag, Berlin, Heidelberg, 2006.

Durnev V. G., Murin D. M., Sokolov V. A., Chalyy D. Ju. , "On Some Approaches to the Solution of the Problem «Useful Proof-of-work for Blockchains»", *Modeling and Analysis of Information Systems*, **25**:4 (2018), 402–410.

DOI: 10.18255/1818-1015-2018-4-402-410

Abstract. The blockchain technology is based on the "Proof-of-work" principles. The essence of this principle is that some event (for example the bill-to-bill money transaction) becomes significant after the confirmation by a certain computer work. So, a demand arose for such computational problems to work on, and we will spend on it about the whole blockchain system computing capacity. Now the main kind of such a problem is a hash-puzzle – the problem to find a bit string with a hash that satisfies some conditions. The important hash-puzzle weakness is the lack of the useful application outside of the blockchain technology. In this work, we offer some approaches to "Useful Proof-of-work for blockchains" problem, namely, consider some practical variants of the NP-complete problems that could be solved with the help of SAT or LLL-solvers as the Proof-of-Work computational problems. The use of the FPT-problems requires special study. The offered approach allows to provide the following characteristics of the proof-of-work computational problems: usefulness, problems complexity management (through the dimension change, choosing problems of certain kind, the indication of necessary solution precision), mass character. Herewith we admit that not every solved problem can be useful but we consider the opportunity to solve some practical problems with the help of the blockchain technology. Among other things it is also possible to compare the virtual crypto-currency value (through the energy costs spent) and the effective result of the practical problems solution. The most complicated points of the described approach are the realization of the events-problems (providing the computer work for these events) relations and the realization of the problems complexity analysis system. This issue should be viewed as the study program because of many technical details that must be worked out further.

Keywords: proof-of-work, blockchain, satisfiability, SAT-solver, NP-complete, FPT, algorithm

On the authors:
Valeriy G. Durnev, Doctor, Professor,
P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl 150003, Russia, e-mail: Durnev@uniyar.ac.ru

Dmitry M. Murin, orcid.org/0000-0002-8068-0784, PhD, Docent,
P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl 150003, Russia, e-mail: nirum87@mail.ru

Valery A. Sokolov, orcid.org/0000-0003-1427-4937, Doctor, Professor,
P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl 150003, Russia, e-mail: valery-sokolov@yandex.ru

Dmitry Ju. Chalyy, orcid.org/0000-0003-0553-7387, PhD, Docent,
P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl 150003, Russia, e-mail: chaly@uniyar.ac.ru