

Развитие алгоритма двустороннего матчинга для повышения качества распределения студентов по руководителям выпускных квалификационных работ

Е.А. Антюхова, Д.Г. Лагереv
antyukhova14pri@gmail.com | LagerevDG@mail.ru
Брянский государственный технический университет

В статье рассматривается проблема распределения студентов по руководителям выпускных квалификационных работ, приводятся результаты опытной апробации подхода к распределению на основе алгоритма двустороннего матчинга с использованием автоматизированной системы распределения студентов, а также результаты сбора и анализа требований по результатам прохождения апробации. Анализируется целесообразность и возможность реализации потребности в переходе от распределения по преподавателям к распределению по проектам, которые находятся в иерархической зависимости по отношению к преподавателю, предлагается модификация математической модели подхода к распределению студентов на основе двустороннего матчинга с учетом интеграции в модель новых компонентов. Выявляются алгоритмические и понятийные проблемы, возникающие при таком изменении алгоритма распределения и приводится разбор путей их решения. В общем случае алгоритм двустороннего матчинга, а также предлагаемая его модификация, может быть использованы не только в процессе формирования списков распределения студентов по руководителям кафедры высшего учебного заведения, но и в других областях производственной сферы, где необходимо решать задачу распределения ограниченного целочисленного ресурса по потребителям в условиях отсутствия прямого антагонизма между ними.

Ключевые слова: двусторонний матчинг, распределение ресурсов между потребителями, выпускная квалификационная работа, принятие решений, анализ требований, адекватность алгоритма, автоматизация производственных процессов, интуитивная понятность программного продукта.

Development of the two-side matching algorithm for improving the quality of student's allocation to thesis advisors

E.A. Antyukhova, D.G. Lagerev
Bryansk state technical university

The article considers the problem of students allocation to thesis advisors, provides the result of the experimental approbation of student's allocation approach based on two-side matching algorithm that has been performed by using automated student's allocation system, and result of requirements management, which had taken place after performance testing. The article analysing user's requirements of changing allocation method replacing tutors with thesis projects embedded into tutor-project hierarchy, in terms of development feasibility and possibility. Mathematics model modifications for the approach to allocating students according to the two-side matching algorithm integrated with new model components is suggested below in the text of the article. The algorithmic and conceptual problems of such changes are pointed out along with solution for overcoming. In the general case, two-side matching model as well as presented modification might be used not only in case of educational and methodical activity aimed at creating mappings lists of students and thesis advisors of a department of a higher educational institution, but also regarding other production and service area, which faces with the problem of limited countable resource distribution to consumers with no outright antagonism.

Keywords: two-side matching model, resource distribution, thesis, decision-making, requirements management, adequacy of the algorithm, autoimmunization of production processes, software intuitive operation.

1. Введение

Решение задачи распределения студентов по руководителям выпускных квалификационных работ (ВКР) является обязательной частью учебного процесса любой выпускающей кафедры высшего учебного заведения, в которой, как правило, задействован профессорско-преподавательский состав (ППС) кафедры и студенты старших курсов. В зависимости от количества обучающихся и занятых в руководстве ВКР преподавателей, преследуемых целей при составлении распределения (получить список быстро, учесть предпочтения преподавателей и др.), имеющих ограничения на правила составления пар «руководитель – студент», а иногда и принятых традиций эта задача решается в рамках каждой из кафедр различными способами как с точки зрения методологий (от директивного распределения руководством кафедры без вовлечения или при минимальном вовлечении в процесс

других участников до применения алгоритмов, учитывающих разного типа ограничения, накладываемыми как «сверху»: учебным планом, решениями руководства кафедры, специализацией преподавателей и т.д., так и «внутри» от преподавателей и студентов при их активной роли в процессе), так и с точки зрения используемых средств (от бумажных до автоматизированных систем).

На базе кафедры «Информатика и программное обеспечение» («ИиПО») Брянского государственного технического университета (БГТУ), проведено несколько апробаций распределения студентов на основе алгоритма двустороннего матчинга Гейла-Шепли 0 с применением автоматизированной системы.

2. Описание исходной модели распределения

Целью реализации алгоритма двустороннего матчинга для распределения студентов по руководителям

ВКР была выработка подхода для формирования эффективного [1] с точки зрения удовлетворения взаимных предпочтений студентов и преподавателей, а также устойчивости алгоритма распределения. Такой показатель качества был выбран исходя из практического подтверждения положения о том, что пары «руководитель – студент», которые имеют взаимную заинтересованность в совместной работе, чаще выдают более хороший результат во время подготовки и защиты ВКР. Учитывая, что качество ВКР является прямым показателем «успешности» выпускающей кафедры, таким образом очевидно, что оно в свою очередь зависит от грамотно сформированных списков распределения, которые учитывают предпочтения обеих сторон процесса.

В результате апробации [1] данного метода в 2016-2017 учебном году было выявлено заметное улучшение по показателю удовлетворенности взаимных предпочтений.

Используемая модель двустороннего матчинга основывается на задаче о приеме в колледжи 0 и в общем виде может быть задана:

- 1) множеством студентов $S = \{s_1, s_2, \dots, s_n\}$;
- 2) множеством преподавателей $T = \{t_1, t_2, \dots, t_m\}$;
- 3) множеством квот преподавателей $Q = \{q_1, q_2, \dots, q_m\}$, где каждое значение q_i соответствует преподавателю t_i ;
- 4) множеством предпочтений студентов $T_j \subseteq T$ и преподавателей $S_i \subseteq S$;
- 5) распределением $M \subseteq T \times S$.

Процедура распределения проводится в несколько итераций $h = \{1, 2, \dots, l\}$, где на каждой итерации формируются списки ожидания преподавателей W_t^h , в которые на $h = 1$ для каждого из преподавателей из T_j входят студенты, указавшие этого преподавателя на первом месте, на $h > 1$ – студенты, имеющие этого преподавателя на минимальном месте после всех отвергнутых предпочтений. Конечное распределение имеет вид: $M(t_i) = W_t^h, h = l; M(s_j) = \{t_i\}$ если $s_j \in W_t^h$, иначе $M(s_j) = \emptyset$. Подробное описание модели двустороннего матчинга представлено в статье [2].

Подсистема управления распределением, разработанная в рамках программного комплекса распределения студентов по руководителям ВКР [2], реализующая логику такого алгоритма, была внедрена в организационно-методическую деятельность кафедры в 2018 и прошла апробацию при распределении бакалавров 3 курса в 2018-2019 учебном году.

3. Модификация модели и алгоритма распределения

Изменение требований к распределению

В процессе эксплуатации программного комплекса стали возникать новые требования или пожелания от разных участников распределения, что является естественным для активного, развивающегося и не до конца устоявшегося процесса. К ролевым группам участников процесса распределения на типичной выпускающей кафедре помимо студентов и преподавате-

лей (как субъектов двустороннего матчинга) и руководства кафедры (как источника некоторых бизнес-ограничений процесса), также относится роль ответственного за распределение, сопоставленная конкретному преподавателю кафедры. В обязанности ответственного входит управление процессом, взаимодействие с участниками, подготовка списков распределенных студентов, полученных в результате выполнения алгоритма, для формирования нормативных документов, внесение ряда ограничений в процесс формирования пар: регулирование доступных каждому преподавателю для руководства студентов отдельных направлений подготовки, предопределение некоторых пар. Наиболее значительным изменением стало возникновение необходимости получения результатов распределения не в виде пар «руководитель – студент», а с учетом проектов руководителей, в рамках которых преподаватели готовы работать со студентами над ВКР [3]. В этом были одновременно заинтересованы как преподаватели, так и руководство кафедры.

С точки зрения преподавателей проектный подход позволит более правильно оценивать намерения студентов, которых они видят в своих списках ожиданий. Анализируя данные, полученные по результатам тестирования на распределении 2018-2019 года, можно отметить, что информацию о преподавателе (которая доступна студентам на этапе составления списка предпочтений) заполнили порядка 80-90% преподавателей и в 100% заполненных описаний фигурировали списки возможных проектов, над которыми преподаватель готов работать в рамках ВКР; в то время как аналогичная информация о себе была заполнена только 5% студентов (2 из 40 участвующих). Таким образом, ответить на вопросы: «Чего ожидает студент, записываясь к мне?», «На какой проект планировать этого студента?», возникающие у многих преподавателей, система не могла по одной из причин:

внесение информации о себе было необязательным, поэтому многие пользователи весьма поверхностно заполняли какую-то информацию о себе или не заполняли ее вообще;

предоставление информации подразумевало свободную форму и, возможно, часть студентов просто не поняли, какую информацию от них ожидают;

в процессе информирования студентов основное внимание обычно уделяется особенностям заполнения списка предпочтений, поскольку он определяет ход алгоритма и напрямую затрагивает интересы студентов; поэтому большинство студентов не понимают важность заполнения дополнительной информации о себе.

Для руководства кафедры это нововведение в первую очередь рассматривалось как средство повышения качества распределения, вследствие обогащения информацией (студенты и преподаватели более правильно формируют взаимные ожидания), а также как способ в значительной мере ускорить формирование нормативных документов, обязательной частью которых является указание темы ВКР каждого студента.

Дополнительной возможностью при введении проектов становится учет квоты в рамках проекта, а не квоты преподавателя, что особенно важно для проектов с ограничением на число допустимых исполнителей, и возникает ситуация, когда к преподавателю распределяется студентов, желающих работать по такому проекту, больше, чем возможно взять на этот проект, и преподавателю приходится предлагать некоторым другим проектам, выбор которых, по факту может разрушить порядок реальных предпочтений этих студентов, хотя с точки зрения модели, не учитывающей проект, удовлетворенность распределением была максимизирована. Отказаться от таких студентов на этапе выполнения распределения нельзя, поскольку они не превысили квоту преподавателя.

Введение новой сущности «проект» потребовало значительных изменений, как в структуре базы данных системы, так и в логике работы программных модулей. «Проект» должен реализовывать большую часть функционала, который раньше обеспечивался объектом «преподаватель». Однако просто перенести программный код из одного объекта в другой было невозможно, поскольку преподаватель в процессе принятия решений должен руководствоваться полным видением ситуации по всем своим проектам, что не позволяет исходная реализация объекта «преподаватель».

В существующей системе ответственным является центром принятия решения об увеличении квоты, в ответ на поступивший запрос от преподавателя. При добавлении проектов количество запросов об увеличении квоты возрастает в несколько раз, поскольку у каждого преподавателя может быть несколько проектов, а у каждого проекта задается своя квота. Поэтому было принято решение разрешить преподавателям самостоятельно управлять квотами проектов.

Некоторые студенты (порядка 15%), не заполняют свои списки предпочтений. В существующей системе такие участники не могли быть распределены автоматически. Ответственному после окончания распределения необходимо было выполнить ручное распределение: простой маппинг нераспределенных студентов на свободные квоты преподавателей, хотя данных студентов можно было бы включить в автоматический процесс распределения.

Изменение модели с учетом новых требований

Преобразуя описание модели матчинга с учетом проекта, были получены следующие изменения:

- 1) добавлено множество проектов $P = \{p_1^1, p_2^1, \dots, p_g^m\}$, где каждому t_j сопоставлено множество $P^j = \{p_1^j, p_2^j, \dots, p_{g'}^j\}$;
- 2) множество квот преподавателей преобразовано во множество квот проектов $Q' = \{q_1^1, q_2^1, \dots, q_g^m\}$;
- 3) множество предпочтений студентов $P_j \subseteq P$;
- 4) получено новое множество предпочтений преподавателей в разрезе проектов $S_g \subseteq S$;
- 5) распределение теперь задается как $M \subseteq P \times S$.

В таком случае результат распределения представляется как: $M(p_i) = W_p^h, h = l$; $M(s_g) = \{p_i\}$ если $s_g \in W_p^h$, иначе $M(s_g) = \emptyset$.

Следует обратить внимание, что задание множества студентов, преподавателей и квот преподавателей остается неизменным. При этом, если задать каждое из Q' действительным числом в общем случае $\sum_1^{g'} q^{j'} \geq q^j$, более того, не выполняется даже ограничение $|P^j| \leq q^j$ (из последнего распределения более 50% преподавателей подтверждают эту особенность), что и является причиной невозможности отказа от понятия квот преподавателя.

Таким образом, алгоритм обязан учитывать как квоту уровня проекта, так и уровня преподавателя, оставаясь при этом в согласованном состоянии.

Разделение понятия квоты

В первую очередь из описанной модели вытекает разделение понятия квоты, которое ранее было единым, на две разные сущности: общая квота преподавателя и квота по проекту. Причем обе квоты накладывают ограничение на количество студентов доступных к выбору. При этом квота преподавателя (общая квота) верифицируется ответственным, в то время как квота по проекту задается преподавателем, но не может быть независимой от общей квоты.

В простом варианте, когда сумма квот по проектам равна общей квоте, проблем в порядке распределения не возникает. Для примера, рассмотрим список ожидания преподавателя A на одной из итераций процесса с общей квотой $q = 4$, некоторым количеством проектов и квот проектов $P_g(q')$, а также множество из студентов $a - g$.

		$q = 4$
$P_1(1)$	$P_2(1)$	$P_3(2)$
a		c
b		d
		e

В таком случае студенты a и e получают отказ на основе квоты по проекту несмотря на то, что из пяти студентов списка отбираются только трое, независимо от общей квоты равной четырем, однако противоречия с точки зрения модели здесь нет. Таким образом, приходим к выводу о том, что квота по проекту приоритетнее.

Но, что произойдет, если преподаватель A придет к пониманию о том, что он предпочитает взять троих студентов на проект P_3 ? Решением может быть увеличение и общей и проектной квоты на один.

		$q = 5$
$P_1(1)$	$P_1(1)$	$P_1(3)$
a		c
b		d
		e

Результат наталкивает на мысль о том, что, несмотря на формальное увеличение квоты, фактически выбранное количество студентов равно четырем, то есть в реальности общая квота не поменялась. И, как было ранее отмечено, выполнение условия этой суммы необязательно, и если оставить общую квоту

неизменной, то, например, на следующей итерации возможно появление противоречия между квотами.

$q = 4$		
$P_1(1)$	$P_1(1)$	$P_1(3)$
a	f	c
		d
		e

Все количество студентов, которое позволяет оставить в списке квота по каждому из проектов, превышает общую квоту, что не должно быть допустимо. Очевидно, что алгоритм должен заставить преподавателя A выбрать только четырех студентов. Таким образом, с квоты по проекту снимается статус более приоритетной, но теперь ее можно определить как ограничение для проекта на количество студентов сверху (но не снизу, поскольку сейчас при исключении одного из студентов по какому-то из трех проектов явным образом нарушается это ограничение).

Что произошло в данном случае?

1. A настолько заинтересован работой с каждым из студентов c , d и e , что готов вместо планируемого количества, взять на одного больше. При этом, он все еще заинтересован в одном студенте на проект P_2 . В этом случае произошло реальное увеличение общей квоты и отказ от ее обновления вместо увеличения в большей степени разрушает реальную картину мира.
2. A на самом деле не имеет желания взять на проект P_2 только одного студента, а на P_3 – двух, его так же устроит и вариант: $P_2(0)$ и $P_3(3)$ и, возможно, другие сочетания. Тогда, у него действительно не изменяется значение общей квоты, и он выбирает уже по предпочтениям по работе с конкретными студентами доступными для выбора.

В рассмотренных вариантах прослеживается недетерминированность поведения системы, которая должна сниматься решением преподавателя, но сможет ли пользователь действительно правильно определить с какой дальнейшей целью он выполняет это действие, которое в обоих вариантах на текущей итерации по сути приводят к одному и тому же результату (выбрано четыре студента, трое из которых записаны на проект P_2). В дополнение можно отметить, что последний вариант интуитивно диссонирует со здравым смыслом ($1 + 1 + 3 = 4$) и требует более глубокого знания системы со стороны преподавателей, что может быть необоснованным требованием к программному продукту, который в рядовом случае преподаватель использует 1-2 раза в год (при участии в бакалаврском или магистерском распределении) и не интуитивно-понятным с точки зрения интерфейса. Но поскольку такой вариант возможен, необходима более понятная альтернатива в способе конфигурации квот.

Другим вопросом при анализе этого случая является то, кого из студентов вообще возможно исключить? Возможно ли отказать студенту a , взятому ранее, если учесть, что на проект P_1 , других студентов не появилось и при возврате к алгоритму без проектов в таком случае считалось, что количество студентов в списке не превышает квоту и, соответственно, отбрасывать некого? Попытка подобного рода аргументации приведет к тому, что аналогичным образом будут

«закреплены» студенты c , d и e и безальтернативно исключен – f . Но, возвращаясь к «физическому смыслу» происходящего, что делать если преподаватель A хочет работать с f (и очевидно созданный самим A проект P_2 ему интересен в равной степени с другими); почему в условиях возможности выбора у преподавателя (в списке доступно пятеро студентов при общей квоте, равной четырем), предпочтения задают проекты? Для ответа на вопрос следует обратиться к ранее определенному понятию субъекта, осуществляющего выбор из списка ожидания, кем остается преподаватель. Перенесение свойств субъекта на проект (даже сопоставленный преподавателю) невозможно ввиду отсутствия у него специфичных предпочтений (например, для некоего преподавателя B $a \gg b$ вне зависимости от проекта). Подводя итог первоначальному вопросу, правильным ответом считается: преподаватель может исключить любого из студентов из списка любого проекта, соблюдая при этом два ограничения:

1. количество оставленных в списке студентов не должно превышать общую квоту;
2. в рамках каждого из проектов не должна быть превышена квота этого проекта.

При таком подходе исключение одного студента a – совершенно корректно.

$q = 4$		
$P_1(1)$	$P_2(1)$	$P_3(3)$
a	f	c
		d
		e

Казалось бы, попытка поддержать «адекватную» сумму квот по проектам (равную общей квоте), сняв квоту с P_1 , могла бы решить проблему понятности и детерминированности.

$q = 4$		
$P_1(0)$	$P_1(1)$	$P_1(3)$
a	f	c
		d
		e

Однако это, во-первых, осложняет работу преподавателю, заставляя его при каждой подобной потребности перераспределять проектную квоту, во-вторых, создает возможность манипуляции предпочтением с целью исключить из списка студентов более, чем было доступно в начальной конфигурации квот.

$q = 4$		
$P_1(0)$	$P_1(3)$	$P_1(1)$
a	f	c
		d
		e

Помимо этого, был нарушен принцип базового алгоритма, запрещающий снижение квоты (в отличие от корректного варианта, где квоты не изменились). В результате чего этот подход был отвергнут.

Переходя к другим вариантам задания проектов преподавателями, наиболее популярным на практике оказался случай, когда проектов даже по количеству больше, чем значение общей квоты.

$q = 4$					
$P_1(?)$	$P_2(?)$	$P_3(?)$	$P_4(?)$	$P_5(?)$	$P_6(?)$

При условии отказа от возможности задавать квоту по каждому из проектов, не заботясь о превышении в сумме общей квоты, нет ни одного варианта каким-либо образом задать квоты по проектам. Но, если рассмотреть, что подразумевается под такой комбинацией, то наверняка можно сказать: преподаватель A готов работать по каждому из своих шести проектов с четырьмя студентами, при этом он понимает, что некоторые из проектов останутся без исполнителей и его это устраивает. То есть у него нет четкого понятия квоты на все или несколько из проектов – ему все равно сколько и куда запишется студентов и его выбор будут диктовать личные предпочтения и соответствие общей квоте. Таким образом, в алгоритм интегрируется понятие неопределенной квоты. Одновременно с этим один или несколько других проектов могут иметь заданную квоту.

$q = 4$					
$P_1 (1)$	$P_2 (?)$	$P_3 (?)$	$P_4 (?)$	$P_5 (?)$	$P_6 (?)$

Такое распределение квот говорит о том, что на проект P_1 преподавателю A необходимо не более одного студента, количество студентов на каждом из других проектов не имеет значения (A предлагает разнообразный выбор, что позволит заинтересовать больше студентов).

Разница между последними двумя случаями заключается в том, как система реагирует на возможность выбора при записи на проект P_1 , например, трех студентов. В первом случае, согласно концепции «все равно» и общей квоте, преподаватель не имеет права отказаться ни от одного из них.

$q = 4$					
$P_1 (?)$	$P_2 (?)$	$P_3 (?)$	$P_4 (?)$	$P_5 (?)$	$P_6 (?)$
a					
b					
c					

Во втором варианте ему доступен выбор только одного из трех студентов независимо от общей квоты. При желании увеличить квоту по проекту P_1 , это будет трактоваться как желание повысить общую квоту, и при применении этих изменений список ожидания будет оставаться в адекватном состоянии, поскольку изменения коснутся только указанных преподавателем проектов.

$q = 4$					
$P_1 (?)$	$P_2 (?)$	$P_3 (?)$	$P_4 (?)$	$P_5 (?)$	$P_6 (?)$
a					
b					
c					

В случае, когда все же у преподавателя есть желание увеличить квоту по проекту без увеличения общей квоты, стоит задуматься не означает ли это то, что по факту определенное количество студентов на этом проекте не имеет значения в отличие от того, кто именно из студентов это будет. И в случае положительного ответа приводит к возможности «снять» заданную квоту, поставив ей статус «не важно».

На этом моменте, возвращаясь к отложенному случаю на первый взгляд «неадекватной» квоты, где было $1+1+3=4$, прослеживается явное совпадение с последним примером. Теперь этот вопрос можно решить установлением специальной квоты «не важно» на

один или несколько проектов, которые действительно соответствуют такому понятию с позиции преподавателя.

В результате, рассматривая проблемы задания квот, решения составлялись таким образом, чтобы не нарушать помимо ранее обозначенных (и не отмененных) ограничений, еще одно интуитивно-полученное: сумма заданных квот по проектам не должна превышать общую. При этом общая концепция не запрещает задание квот по каждому из проектов, но в этом случае требует равенства суммы проектных квот общей.

Особенности распределения студентов без предпочтений

Задача минимизации ручного маппинга нераспределенных студентов в условиях, когда большая часть нераспределенных студентов формируется из тех, кто не составил свои списки распределения (на примере распределения 2018-2019 учебного года – 15% студентов из общего числа не воспользовались возможностью задать списки и 5% студентов не были распределены при их наличии) сводится к решению проблемы заполненности списков предпочтений студентов. При этом, решение, основанное на увеличении степени работы со студентами с целью повышения количества составленных списков, не сможет принести достаточных результатов – опыт показывает, что процент таких студентов присутствует всегда и в большей степени зависит от состава групп выпускников, чем от дополнительной проработки вопросов распределения. В связи с этим вопрос перемещается в плоскость обработки таких случаев алгоритмически.

Следует отметить, что в результате таких распределений оставались преподаватели со значительно незаполненной квотой. Поскольку описанную модель поведения студентов справедливо трактовать как «кто из преподавателей будет назначен для меня – не важно», было предложено решение заполнения списка предпочтений этих студентов автоматически, причем таким образом, чтобы преимущественно отдавать предпочтение тем преподавателям, у которых потенциально образуется «свободная» квота (определяется по количеству студентов указавших каждого из преподавателей в списке предпочтений – «популярности»). Апробация такого подхода в процессе последнего распределения показала, что все студенты с автоматически заполненными списками были распределены в процессе выполнения алгоритма и не потребовали ручного маппинга.

При применении логики, включающей в себя проектный подход, автоматические списки формируются с применением понятия «популярности» для проектов.

4. Заключение

Необходимость перехода к проектному подходу основывается на подтверждении улучшения качества распределения, путем формирования более адекватных ожиданий у всех участников процесса, упрощает получение необходимой информации в процессе, позволяет ускорить процесс распределения, и может быть

использован в других предметных областях, например для распределения сотрудников по проектам.

Алгоритм распределения не может быть перенесен один к одному с понятия распределения по преподавателям на проекты, для этого требуется: введение отдельных понятий квот проекта и преподавателя; возможность не указывать проектную квоту, введение ограничений на квоты: суммарное значение заданных квот по проектам не должно превышать общее, снижение значение любой квоты – недопустимо; реализация возможности выбора преподавателя «сквозным» образом по всем его проектам с учетом одновременного соблюдения обеих квот; поддержка для ответственного управления квотами только на уровне общей квоты преподавателя при двухуровневом управлении квотами для преподавателя.

В сложившемся понимании бизнес-процесса, модификация отдельных частей архитектуры программного комплекса распределения студентов по руководителям ВКР несет в себе разрушающие эволюционное развитие изменения этих частей. В связи с этим было принято решение полной переработки этих модулей с наследованием только сохранившихся методик и подходов предыдущей версии системы.

Библиографический список

- [1] Автоматизация распределения студентов по руководителям выпускных квалификационных работ с применением модели двустороннего матчинга / А.Г. Подвесовский, Д.Г. Лагерев, И.Г. Егорова. – DOI 10.25559/SITITO.2017.4.514. – Текст: электронный // Современные информационные технологии и ИТ-образование. – 2017. – Т.14, № 3. – С. 147-157.
- [2] Система программной поддержки распределения студентов по руководителям выпускных квалификационных работ: архитектура и особенности реализации / И.Г. Егорова, Е.А. Антюхова. – Текст.: электронный // Молодежь и современные информационные технологии: сборник трудов XVI Международной научно-практической конференции студентов, аспирантов и молодых ученых (Томск, 3–7 декабря 2018 г.) / Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2019. – С. 245–246. – URL: http://msit.tpu.ru/assets/digestArticles/msit_2018.zip (дата обращения: 10.03.2020).
- [3] Two algorithms for the Student-Project Allocation problem / D.J. Abraham, R.W. Irving, and D.F. Manlove, – DOI 10.1016/j.jda.2006.03.006. – Текст.: электронный // Journal of Discrete Algorithms. – 2007, March. – Vol. 5, Iss. 1. – P. 73-90.

Об авторах

Антюхова Екатерина Андреевна – магистрант кафедры «Информатика и программное обеспечение» Брянского государственного технического университета. E-mail: katya.antyukhova@mail.ru

Лагерев Дмитрий Григорьевич – к.т.н., доц., доцент кафедры «Информатика и программное обеспечение» Брянского государственного технического университета. E-mail: LagerevDG@mail.ru