Раздел I. Моделирование процессов и систем

УДК 681.3.068:796.01 DOI 10.18522/2311-3103-2022-6-6-21

В.М. Глушань

ПРИМЕНЕНИЕ ГРАФОВЫХ МОДЕЛЕЙ ПРИ РЕШЕНИИ ПРАКТИЧЕСКИХ ЗАДАЧ ОБРАЗОВАНИЯ И СПОРТА

Статья является обобщением некоторых работ автора, подготовленных им как индивидуально, так и в соавторстве с коллегами. В статье показано, как теория графов может применяться, казалось бы, в таких различных областях как образование и спорт. В первом случае графовая модель использована для формулировки и решения задачи составления оптимальных тестовых заданий (билетов). Задача составления оптимальных тестовых заданий формулируется как задача разрезания (разбиения) графа $G(N,R)$ *на подграфы* $G_i(N_i,R_i)$. *Исходный граф G(N, R) разбивается на заданное число K подграфов* $G_i(N_i,R_i)$, *где* $i \in K$, *так, чтобы сложность каждого задания была одинакова и в каждом задании находилось минимальное число вопросов из одной и той же темы. Особенности формулировки приведенной задачи порождают множество эвристических алгоритмов ее решения. В статье рассматривается следующая эвристика: каждое тестовое задание формируется последовательно, а каждый очередной вопрос помещается в текущее тестовое задание, если его оценка является ближайшей к относительной величине разности средней сложности тестовых заданий и суммарной сложности тех вопросов, которые уже включены в данное тестовое задание, к сумме вопросов, которое осталось включить в задание. Представлены алгоритмы и результаты их программных реализаций, с помощью которых проведены исследования по оптимальному формированию тестовых заданий, предназначенных для контроля знаний обучаемых. Анализируются различные эвристики, позволяющие осуществлять оптимизацию тестовых заданий. Во втором случае показано, что турнирные таблицы для проведения спортивных мероприятий также могут быть представлены графовыми моделями. Формально задача жеребьевки, как и в случае формирования тестовых заданий, сводится к задаче разбиения графа на подграфы, каждый из которых будет соответствовать одной из групп в турнирной таблице. При этом каждая вершина графа соответствует рейтингу определенного участника турнира. Ребрами графа отображаются отношения между участниками – наличие ребра говорит о том, что соответствующие участники являются представителями одной ассоциации или клуба. Опираясь на эти модели, приводятся описания разработанных алгоритмов и результаты их программных реализаций по оптимальному формированию турнирных таблиц, используемых при проведении соревнований на примере настольного тенниса. Анализируются эвристики для одно и двухкритериальной оптимизации построения турнирных таблиц. Общность и преемственность в алгоритмах формирования турнирных таблиц и последовательного распределения вопросов в тестовых заданиях проявляется в использовании не только графовых моделей, но и одних и тех же аналитических соотношений, для формализации используемых эвристик.*

Графовые модели; тестовые задания; жеребьёвка; турнирная таблица.

V.M. Glushan

APPLICATION OF GRAPH MODELS IN SOLVING PRACTICAL PROBLEMS OF EDUCATION AND SPORT

The article is a generalization of some of the author's works, prepared by him both individually and in collaboration with colleagues. The article shows how graph theory can be applied, it would seem, in such diverse areas as education and sports. In the first case, the graph model is used to formulate and solve the problem of compiling optimal test tasks (tickets). It is formulated as a problem of cutting (partitioning) the graph G (N, R) into subgraphs. The original graph G(N, R) is divided into a given number K of subgraphs where the same topic. The peculiarities of the formulation of the given problem give rise to many heuristic algorithms for its solution. The following heuristic is considered in the article: each test task is formed sequentially, and each next question is placed in the current test task if its score is the closest to the relative value of the difference between the average complexity of test tasks and the total complexity of those questions that are already included in this test task, to the amount of questions left to include in the task. Algorithms and results of their software implementations are presented, with the help of which studies were carried out on the optimal formation of test tasks designed to control the knowledge of trainees. Various heuristics are analyzed that allow optimization of test tasks. In the second case, it is shown that tournament tables for sporting events can also be represented by graph models. Formally, the task of drawing lots, as in the case of the formation of test tasks, is reduced to the task of splitting the graph into subgraphs, each of which will correspond to one of the groups in the tournament table. In addition, each vertex of the graph corresponds to the rating of a certain participant in the tournament. The edges of the graph show the relationship between the participants – the presence of an edge indicates that the corresponding participants are representatives of the same association or club. Based on these models, descriptions of the developed algorithms and the results of their software implementations for the optimal formation of tournament tables used in competitions on the example of table tennis are given. Heuristics are analyzed for one- and two-criteria optimization of tournament tables construction. The commonality and continuity in the algorithms for the formation of tournament tables and the sequential distribution of questions in test tasks is to use not only graph models, but also the same analytical relationships to formalize the heuristics used.

Graph models; test tasks; draw; tournament table.

Введение. Теория графов находит широкое применение в алгоритмизации самых разнообразных задач [1–3]. Данная статья является попыткой некоторого обобщения работ автора, в которых строятся алгоритмы, опирающиеся на графовые модели, для решения оптимизационных задач из области образования и спорта. Существенный подъем применения информационных технологий в области образования отмечается с конца XIX-го начала XX-го веков. В это время особенно интенсивно разрабатываются компьютерные средства обучения [4] и программно-аппаратные средства для контроля знаний обучаемых [5–8].

Статьи [9–11] посвящены разработке и исследованию алгоритмов для оптимизации тестовых заданий, используемых для контроля знаний обучаемых. В статьях [12–15] рассматриваются вопросы автоматизированного построения турнирных таблиц. На некоторые программы, реализующих алгоритмы, получены соответствующие свидетельства [16, 17].

Алгоритмы построения тестовых заданий. Контроль и оценка знаний – важное мероприятие любого процесса обучения. Для его реализации, как правило, составляются тестовые задания (билеты), которые предъявляются обучаемым. Каждое задание состоит из нескольких вопросов и соответствующим образом сформированных ответов на каждый вопрос задания. При составлении заданий возникает множество психолого-педагогических и дидактических проблем. Одной из них является проблема соблюдения принципа «справедливости». Эта проблема возникает, как правило, по двум причинам: 1) в связи с различной сложностью вопросов, изучаемых практически в любом предмете и 2) с необходимостью рассредоточения вопросов по заданиям, т.е. в каждом задании нежелательно поме-

щать вопросы из одного и того же раздела изучаемого предмета. Для решения этой проблемы предлагаются различные алгоритмы. В [9] предложен и исследован алгоритм, основанный на графовой модели, формулировка которой приведена ниже.

Пусть имеется множество вопросов N = {1, 2, 3, …, n}, по которым должны контролироваться знания обучаемых, задана сложность каждого вопроса, выраженная в некоторой ξ-бальной системе Ξ = {1, 2, 3, …, ξ}. Отношение принадлежности вопросов к одному и тому же разделу предмета задается матрицей смежности некоторого графа G(N, R), в котором N – множество вопросов (множество вершин графа), R – множество всех подмножеств ребер графа. Каждое подмножество отображает принадлежность вопроса к какой-либо одной теме (если вопросы принадлежат к одной теме, то в целом графе G(N, R) они отображаются полносвязным подграфом).

Сформулируем задачу составления оптимальных тестовых заданий (билетов) как задачу разрезания (разбиения) графа G(N, R) на подграфы $G_i(N_i, R_i)$. Исходный граф G(N, R) необходимо разбить на заданное число *K* подграфов $G_i(N_i, R_i)$, где $i \in K$ так, чтобы сложность каждого задания (билета) была одинакова и в каждом задании (билете) находилось минимальное число вопросов из одной и той же темы. При этом должны выполняться условия:

$$G_i(N_i,R_i) \neq \emptyset; \quad (1)$$

$$G_i(N_i,R_i) \cap_{i=j} G_j(N_j,R_j) = 0; \quad (2)$$

$$\cup_{i=1}^{K} G_i(N_i,R_i) = G(N,R); \quad (3)$$

$$\sum_{i=1}^{K} N_i = N. \quad (4)$$

Условие (1) означает, что каждый подграф не должен быть пустым (в терминах вопросов и заданий это означает, что каждый вопрос должен быть распределен в какое-то задание); условие (2) означает, что пересечение подграфов должно быть равным нулю, т.е. ни одна вершина не может принадлежать более чем одному подграфу (каждый вопрос должен входить только в одно задание); условие (3) означает, что объединение всех подграфов есть сам граф и, наконец, условие (4) означает, что все вопросы должны быть распределены по заданиям.

Существует большое разнообразие формулировок графовых задач. Однако формулировка каждой задачи содержит и специфические различия, определяющих алгоритмическую сложность решения индивидуальной задачи. Формулировка приведенной задачи обладает двумя особенностями: 1) сложность каждого задания (билета) должна быть одинакова или, по крайней мере, иметь минимальную дисперсию, 2) в каждое задание должны входить вопросы из разных тем или, по крайней мере, таких вопросов должно быть минимальное количество.

Особенности формулировки приведенной задачи порождают множество эвристических алгоритмов ее решения. Здесь рассматривается следующая эвристика: каждое тестовое задание формируется последовательно, а каждый очередной вопрос помещается в текущее тестовое задание, если его оценка является ближайшей к относительной величине разности средней сложности тестовых заданий и суммарной сложности тех вопросов, которые уже включены в данное тестовое задание, к сумме вопросов, которое осталось включить в задание.

Приведенную эвристику представим в формализованном виде. Все тестовые задания заполняются последовательно вопросами из множества вопросов $V = \{v_1, v_{2,}, \ldots, v_n\}$. Каждому вопросу соответствует бальная оценка из множества оценок $C = \{c_1, c_2, \ldots, c_n\}$. В каждое задание включается одинаковое число во-

просов, равное величине | V |/|T|, где T ={t_1, t_2,…, t_m} – множество тестовых заданий. В основу выбора очередного вопроса в тестовое задание t_i положена следующая эвристика. Для всех еще нераспределенных к i-му шагу вопросов $v \in T_i = |T| \setminus |T_{i-1}|$ выбирается такой вопрос t_i, бальная сложность которого равна или меньше всего отличается от величины

$$\tilde{c}_i = \frac{(Q - g_{i-1})m}{n - m(i-1)},$$

где Q – расчетная средняя бальная оценка сложности одного задания, определяемая по формуле

$$Q = \frac{\sum_{i=1}^{n} c_i}{m},$$

g_{i-1} – сумма бальных оценок вопросов, уже вошедших в задание и рассчитываемая по формуле

$$g_{i-1} = \sum_{c_j \in C_{i-1}} c_j.$$

Таким образом, значение величины $\tilde{c}$ определяет тот вопрос, бальная оценка которого наиболее благоприятна для включения в формируемое на i-м шаге тестовое задание.

Для экспериментального исследования первого варианта последовательного алгоритма, блок-схема которого приведена в [9], был разработан программный код. С его помощью были проведены три серии экспериментов по формированию тестовых заданий (билетов), включающих 5, 8 и 10 вопросов соответственно. Бальная оценка вопросов варьировалась в двух диапазонах: от 1 до 5 и от 1 до 10, которая устанавливалась случайно в заданных диапазонах. Некоторые результаты исследований приведены ниже в табл. 1. В качестве наглядного примера на рис. 1 приведена, построенная по экспериментальным данным диаграмма распределений 43 вопросов по 5 билетам при суммарной оценке всех вопросов в 235 баллов и по 8 вопросов в каждом билете, а на рис. 2 – диаграмма распределений тех же вопросов по 21 билету и по 2 вопроса в каждом билете.

Таблица 1

Результаты формирования заданий

Цена вопроса / Количество вопросов	Дисперсия = 0			
	Количество заданий	5	8	10
	Номер эксперта	Номер и цена задания 1 2 3 4 5	Номер и цена задания 1 2 3 4 5 6 7 8	Номер и цена задания 1 2 3 4 5 6 7 8 9 10
(1-5) / 45	1	25 27 25 26 24	14 15 14 14 14 15 14 12	11 12 11 12 9 11 10 11 12 12
	2	25 27 24 27 24	14 17 18 18 17 18 18 16	10 8 12 11 12 8 12 8 10 11
	3	26 24 26 25 25	15 15 14 13 16 14 13 16	12 13 11 13 14 13 14 13 14 13
(1-10) / 45	1	44 44 44 43 44	-	-
	2	44 42 47 42 41	-	-
	3	42 39 40 43 39	-	-
(1-10) / 90	1	105 99 104 101 98	-	-
	2	97 95 97 96 94	-	-
	3	98 98 94 99 92	-	-

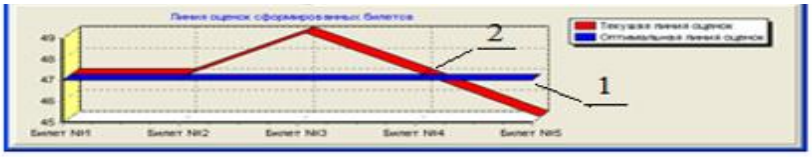

Рис. 1. Диаграмма распределений вопросов в 5 билетов

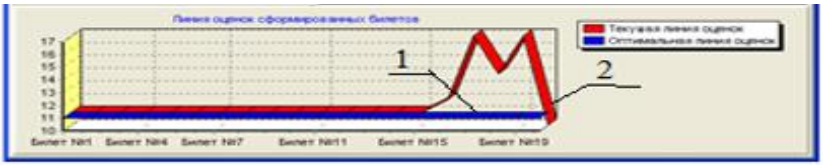

Рис. 2. Диаграмма распределений вопросов в 21 билет

Из приведенных на рисунках диаграмм можно сделать вывод, что при распределении вопросов на малое число билетов с достаточно большим числом вопросов в каждом билете (рис. 1) отклонение линии диаграммы (отмеченной цифрой 2) от оптимального равномерного распределения (отмеченного цифрой 1) не превышает двух баллов или 4,3% (билеты 3 и 5). Из рис. 2 видно, что при распределении вопросов на большое число билетов (в данном случае на 21 и лишь с 2-мя вопросами в билете) оценки сложности билетов от 1-го до 14-го совпадают с теоретически оптимальным значением. Однако среди сформированных билетов в конце работы программы имеются такие (билеты 16 и 18), которые отличаются от теоретически оптимальных на 6 баллов, что составляет 54%. Такой разброс (дисперсия) тестовых заданий связан с негативной особенностью последовательного алгоритма и может быть объяснено тем, что вопросы с небольшими значениями оценок в базе данных закончились на 14-м билете, поэтому алгоритм был вынужден выбрать вопросы с оставшимися большими бальными значениями оценок.

Отмеченный недостаток первого варианта последовательного алгоритма привел к необходимости его устранения. В связи с этим был разработан модифицированный вариант последовательного алгоритма [10] и две модификации параллельного алгоритма [11]. В первой модификации параллельного алгоритма все тестовые задания предлагается формировать одновременно (параллельно). В этом случае на каждом шаге работы такого алгоритма все тестовые задания будут получать по одному вопросу одновременно. Тогда к концу процесса формирования тестовых заданий в неблагоприятной ситуации окажется не одно, два последних задания, а все задания. Но такая неблагоприятная ситуация окажется сглаженной, распределенной по всем заданиям. В конечном итоге, это позволит, если не полностью избежать, то, по крайней мере, уменьшить неравномерность распределения вопросов по заданиям.

Сущность второй модификации состоит в том, чтобы сложность $\tilde{C}_i$ следующего вопроса, который должен войти в формируемое тестовое задание, не рассчитывать, как это делается в последовательном алгоритме, а брать вопрос наибольшей сложности из еще нераспределенных вопросов. Отсюда очевидно, что параллельное распределение вопросов следует начинать с самых сложных вопросов и заканчивать самыми легкими.

Действительно, распределив сначала самые сложные вопросы по тестовым заданиям, затем будет легче варьировать самыми легкими вопросами (т.е. с самой низкой бальной оценкой) для достижения сбалансированности суммарной сложности задания. В процессе распределения вопросов целесообразно также учитывать, в какое задание нужно распределить самый сложный вопрос на *n*-м шаге, а после этого целесообразно отсортировать порядок распределения по заданиям оставшихся вопросов.

Результаты экспериментальных исследований алгоритмов. Для установления более объективной оценки рассмотренных алгоритмов проведено их исследование для различных исходных данных. В качестве исходных данных в исследованиях использовалось различное число вопросов, различное число билетов, различные оценки сложности вопросов, а также различное число изучаемых тем. Проведено 3 серии экспериментов. В каждой серии сложности вопросов генерировались случайно, поэтому в приведенных ниже таблицах указывается лишь математическое ожидание сложности всех участвующих в эксперименте тестовых заданий. Проведение экспериментов преследовало две цели. Первая цель состояла в установлении дисперсии сложности сформированных тестовых заданий тремя предложенными алгоритмами. Целью второй серии экспериментов является установление временной сложности алгоритмов.

Для достижения первой цели в экспериментах варьировались условия их проведения в каждой серии. Эксперименты выполнялись в соответствии с заданными условиями табл. 2, 3 и 4, а результаты экспериментов представлены соответственно на рис. 3, 4 и 5.

Таблица 2

Данные к первой серии экспериментов

Кол-во заданий	Кол-во вопросов в задании	Макс. сложность вопроса	Кол-во тем	Мат. ожидание	Дисперсия результатов для алгоритмов		
					Последовательный	Параллельный	Оптимизированный
5	5	5	1	14,4	1,84	0,24	0,24
5	5	5	1	12,4	0,64	0,64	0,24
5	5	5	1	11,4	2,24	0,24	0,24
5	5	5	1	14,6	1,84	0,64	0,24
5	5	5	1	13,6	0,24	0,24	0,24

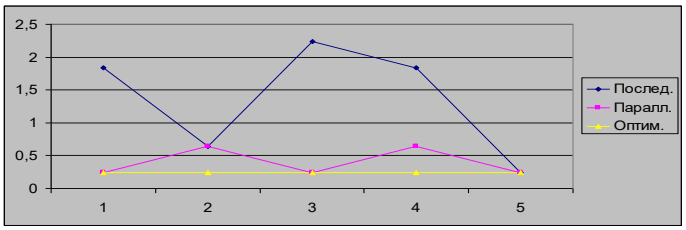

Рис. 3. Дисперсия результатов по данным табл. 2

Таблица 3

Данные ко второй серии экспериментов

Кол-во заданий	Кол-во вопросов в задании	Макс. сложность вопроса	Кол-во тем	Мат. ожидание	Дисперсия результатов для алгоритмов		
					Последовательный	Параллельный	Оптимизированный
5	5	5	5	10,8	2,16	1,36	0,16
5	5	5	5	9,8	0,56	0,16	0,16
5	5	5	5	12,6	1,84	0,24	0,24
5	5	5	5	14,2	2,96	0,96	0,16
5	5	5	5	14,6	1,04	0,64	0,24

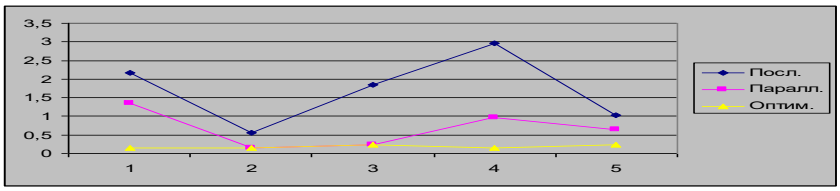

Рис. 4. Дисперсия результатов по данным табл. 3

Таблица 4

Данные к третьей серии экспериментов

Кол-во билетов	Кол-во вопросов в билете	Макс. сложность вопроса	Кол-во тем	Мат. ожидание	Дисперсия результатов для алгоритмов		
					Последовательный	Параллельный	Оптимизированный
15	10	10	5	49,13	4,51	3,44	0,11
15	10	10	5	50,2	4,63	3,22	0,16
15	10	10	5	57	5,86	3,46	0
15	10	10	5	47,8	2,96	2,82	0,16
15	10	10	5	57	3,73	3,2	0

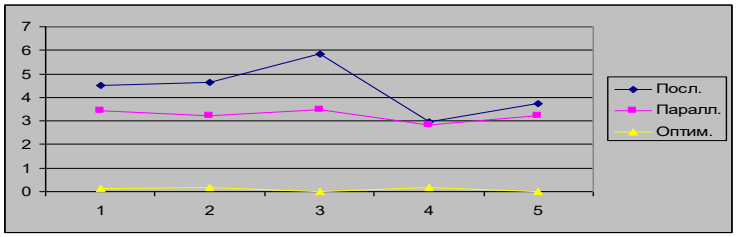

Рис. 5. Дисперсия результатов по данным табл. 4

Для достижения второй цели в код программы был инжектирован модуль оценки временной сложности алгоритма, с помощью которого фиксировалось время работы алгоритма при увеличении количества заданий и количества вопросов в них, максимально возможной сложности задания и принадлежности вопросов определенным темам. Серия экспериментов состояла из нескольких групп, а каждая группа формировала тестовые задания последовательным, параллельным и оптимизированным алгоритмами. Результаты экспериментов обрабатывались с помощью пакета программ MathCad. Ниже приведены две наиболее репрезентативные группы из второй серии экспериментов.

Количество заданий в первой группе экспериментов изменялось от 20 до 30, количество вопросов в задании изменялось с 10 до 25, максимальная сложность вопроса оценивалась в 10 баллов, деление на темы отсутствовало. По оси абсцисс откладывалось количество заданий, а по оси ординат – абсолютное время в миллисекундах. При этом в диапазоне изменения количества заданий от 20 до 30 строились реальные зависимости, а в диапазонах от 0 до 20, и от 30 до 70 зависимости предсказывались путем экстраполяции пакетом программ MathCad. Такой подход был использован, с одной стороны, с целью сокращения объема экспериментальных исследований, а с другой стороны, с целью выявления тенденции изменения временной сложности алгоритмов в зависимости от соотношения числа заданий и

числа вопросов в них. В связи с этим на приведенных ниже рис. 6–11 выделяются три облака точек. Пунктирная кривая является усредненным значением тенденции изменения сложности соответствующего алгоритма в зависимости от соотношения числа заданий и числа вопросов.

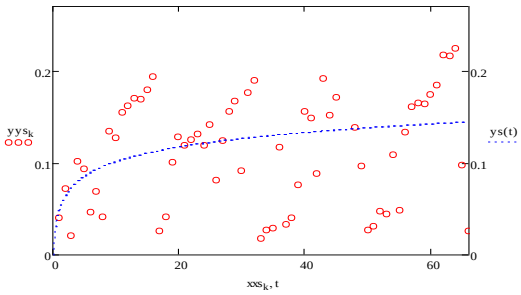

Рис. 6. Последовательный алгоритм без деления на темы

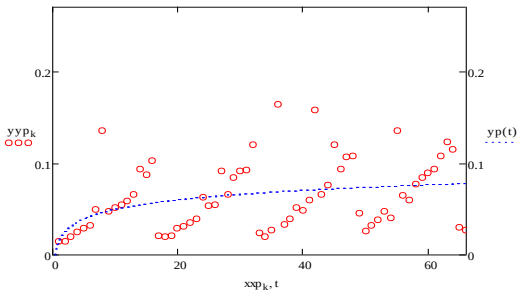

Рис. 7. Параллельный алгоритм без деления на темы

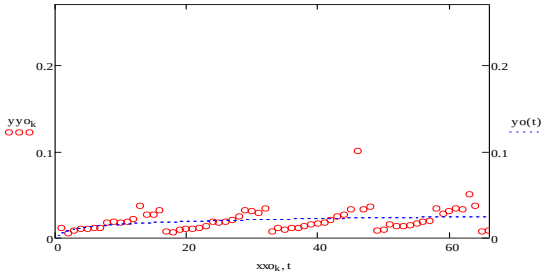

Рис. 8. Оптимизированный алгоритм без деления на темы

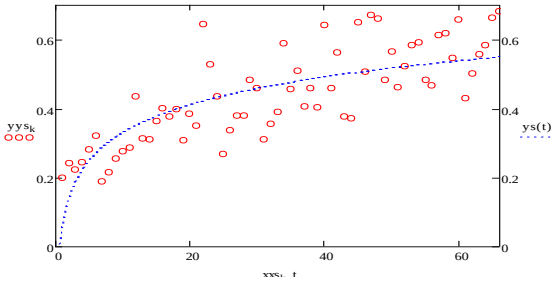

Рис. 9. Последовательный алгоритм, деление вопросов на 10 тем

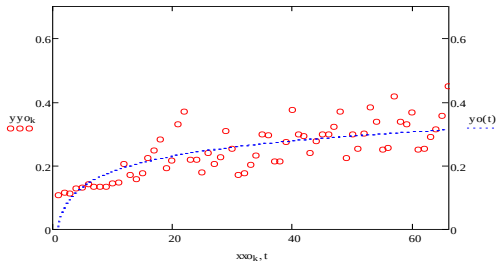

Рис. 10. Параллельный алгоритм, деление вопросов на 10 тем

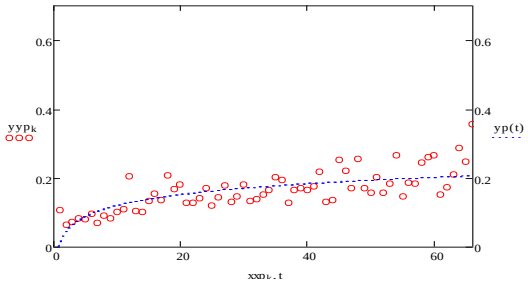

Рис. 11. Оптимизированный алгоритм, деление вопросов на 10 тем

Проведенные исследования показывают, что последовательный алгоритм распределения вопросов по билетам является, как и следовало ожидать, достаточно быстрым. Но результат его работы приводит к непредсказуемо большой дисперсии сложности билетов. Получение более качественных результатов относительно дисперсии сложности билетов возможно с помощью параллельного алгоритма. Однако она является все еще достаточно высокой. Оптимизированный алгоритм обладает значительно меньшей дисперсией, позволяющей формировать тестовые задания с практически равной сложностью при удовлетворительных временных характеристиках. Это стало главной предпосылкой для использования оптимизированного алгоритма в основе модуля формирования тестовых заданий, являющегося важной составной частью автоматизированного рабочего места преподавателя [18, 19].

Алгоритмы построения турнирных таблиц. Информационные технологии получили чрезвычайно широкое распространение в виде создания веб-сайтов. Не стали исключением в этом смысле и спортивные соревнования. Примеры таких веб-сайтов представлены в [20, 21]. В первом из них представлено программное обеспечение для самых разнообразных спортивных соревнований. Наиболее известной в нашей стране является компания «Исток системы» [21], которая занимается как созданием, так и распространением программ компьютерной поддержки спортивных мероприятий. Так, например, программа «Турнир НТ» создана для помощи судьям и организаторам турниров по настольному теннису. Она позволяет при проведении турнира обойтись без бумажных таблиц. За счет встроенных формул и макросов она автоматизирует многие операции, как при распределении игроков, так и при заполнении протоколов турнира. Однако в этой системе практически не используются методы оптимизации автоматизированного формирования турнирных таблиц.

В то же время, любому спортивному мероприятию по игровым видам спорта предшествует кропотливый и достаточно трудоемкий подготовительный этап проведения жеребьевки. Цель жеребьевки – формирование турнирной таблицы. Особое значение это имеет для тех видов спорта, в которых принимает участие большое количество игроков из разных регионов. В этих случаях цель жеребьевки со-

стоит в том, чтобы рассеять игроков по разным группам таким образом, чтобы все группы имели примерно одинаковый суммарный рейтинг, т.е. были равными по силам, и при этом в каждую группу входили игроки из разных регионов. Ситуация значительно усложняется, когда от одного региона заявки на участие в турнире подали несколько его представителей. Таких участников необходимо распределить в различные отборочные группы. Таким образом, задача становится двухкритериальной: первый критерий – суммарный рейтинг групп (он должен быть по возможности одинаковым), второй критерий – распределение участников турнира по группам из одного региона (оно должно быть равномерным). Решение такой задачи уже требует специфичных алгоритмов. Среди видов спорта, в которых такая ситуация типична, является настольный теннис.

Формально задачу жеребьевки можно свести к известной математической задаче разбиения графа на подграфы, каждый из которых будет соответствовать одной из групп в турнирной таблице. Каждая вершина графа имеет вес, характеризующий рейтинг определенного участника. Ребрами графа отображаются отношения между участниками – наличие ребра говорит о том, что соответствующие участники являются представителями одной ассоциации или клуба. Пример графовой модели данной задачи приведен на рис. 12.

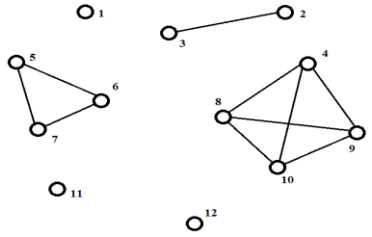

Рис. 12. Пример графовой модели для задачи жеребьевки

На приведенном рисунке цифрами обозначены номера участников турнира. В данном случае их 12, объединенных в 6 компонент, число которых отображает число различных ассоциаций (клубов):

$$s_1 = \{1\},\ s_2 = \{2,3\},\ s_3 = \{5,6,7\},\ s_4 = \{4,8,9,10\},\ s_5 = \{11\},\ s_6 = \{12\}.$$

Такой граф является симметрическим, а каждая компонента является кликой графа, т.е. полно связным подграфом, и при этом выполняется условие транзитивности: если *A* и *B* – одноклубники, *A* и *C* – одноклубники, то *B* и *C* также являются одноклубниками.

Математическую модель задачи можно представить либо в виде матрицы смежности графа $D = \|d_{ij}\|$, либо в виде матрицы инциденций $S = \|s_{ik}\|$ и вектором $P = \{p_1, p_2, ..., p_n\}$, где p_i – рейтинг *i*-го участника. В первом случае матрица смежности *D* является квадратной с размерностью $n \times n$ элементов, а $d_{ij} = 1$, если участник с номером *i* является одноклубником с участником под номером *j*, иначе $d_{ij} = 0.$ Во втором случае матрица инициденций *S* имеет меньшую прямоугольную размерность $n \times k.$

Цель такой задачи разбиения состоит в получении непересекающихся подграфов (т.е. когда одна вершина не может принадлежать более, чем одному подграфу), чтобы число внутренних связей между вершинами было минимально или, соответственно, число внешних связей было максимально.

Рассмотрим подходы и приемы, которые сейчас используются в реальных случаях ручного проведения жеребьевки для построения эффективных алгоритмов автоматизированного формирования турнирных таблиц.

Самый распространенный способ ручной жеребьевки основан на так называемом посеве змейкой. Он осуществляется следующим образом. Сначала все игроки упорядочиваются по рейтингу. Затем, начиная с игрока с наивысшим рейтингом, последовательно по мере убывания рейтинга игроков их атрибутами (данными об игроках) заполняются первые *m* мест, начиная с 1-ой по *m*-ю группу. Следующими по рейтингу *m* игроками продолжают заполнение вторых мест во всех группах, начиная с *m*-ой группы и заканчивая 1-ой группой. Очередными *m* игроками продолжают заполнение третьих мест во всех группах, снова начиная с 1-ой группы и заканчивая *m*-ой группой. Такой посев змейкой продолжают до тех пор, пока не заполнятся все места во всех группах. Качественно этот процесс представлен табл. 5 и 6.

Таблица 5

Направления посева змейкой

Гр.1	Гр.2	Гр.3	Гр.4	Гр.5
→	→	→	→	→
←	←	←	←	←
→	→	→	→	→
←	←	←	←	←

Таблица 6

Пример посева змейкой

20	19	18	17	16
11	12	13	14	15
10	9	8	7	6
1	2	3	4	5
42	42	42	42	42

Табл. 5 демонстрирует процесс посева змейкой, а табл. 6 – равенство сумм всех чисел в каждом столбце, если каждое последующее число в таблице отличается от предыдущего на постоянную величину. Наибольшая применимость посева змейкой объясняется именно тем, что он формирует примерно одинаковые по силе группы.

Формально, используя понятие инверсных (обратных) чисел, нетрудно показать, что если рейтинги участников понижаются от максимального до минимального на постоянную величину и при четном числе строк, то суммарные рейтинги всех групп будут одинаковыми. Для простоты будем считать, что имеется некоторая ограниченная убывающая последовательность натуральных чисел $R=\{r_1, r_2, ..., r_k\}$. Для любого числа этой последовательности существует инверсное число $r \in R$, определяемое следующим образом

$$\overline{r} = r_1 + r_k - r. \quad (5)$$

Каждую пару строк табл. 2 с записанными в их ячейках рейтингами в общем виде развернем в одну строку удвоенной длины, где *m* это число групп. Тогда для первой пары строк мы получим развертку – рис. 13.

r	*r* - Δ	*r* – 2*r*	*r* – 3*r*	. . .	*r* – (2*m*-2) Δ	*r* – (2*m*-2) Δ

Рис. 13. Развертка двух соседних строк при посеве змейкой

В соответствии с рис. 13 для каждого из первых *m* чисел инверсными будут последние *m* чисел, но в обратном порядке. Исходя из представления инверсного числа формулой (5), ясно, что суммы всех прямых и инверсных чисел одинаковы. Причем первое прямое число и инверсное ему последнее число входят в первый столбец табл. 1, второе прямое число и предпоследнее число входят во второй столбец и т.д. Развернув аналогично вторую пару строк табл. 2, мы получим также одинаковые суммы соответствующих прямых и инверсных чисел. В конечном итоге, сумма чисел во всех столбцах будет одной и той же.

Жеребьевка с посевом змейкой находит широкое применение в организации турниров, в которых от каждого региона заявлено только по одному участнику. Ситуация значительно усложняется, когда от одного региона заявки на участие в турнире подали несколько его представителей. Необходимость учета регионального фактора вытекает из того, что часто в списке участников соревнований находятся игроки, которые имеют потенциально общие интересы. Например, какие-то участники представляют один клуб или одну школу в городских соревнованиях; один город в областных соревнованиях; один регион во всероссийских соревнованиях; одну страну в международных соревнованиях. Чтобы избежать любых попыток к сговору, таких, например, как: участник, который потерял шансы на выход из отборочной группы намеренно проигрывает своему одноклубнику, чтобы повысить его шансы на выход из отборочной группы. Таких участников необходимо распределить в различные отборочные группы [22]. Таким образом, задача становится двухкритериальной: первый критерий – суммарный рейтинг групп (он должен быть по возможности одинаковым), второй критерий – распределение участников турнира по группам из одного региона (оно должно быть равномерным). Решение такой задачи уже требует специфичных алгоритмов.

В [12] предложен и проанализирован числовой критерий равномерности распределения участников турнира по группам, который имеет вид

$$K_p = \sum_{i=1}^{l} \sum_{j=1}^{m} (a_{ij})^2 / m,$$

где m – число групп, l – число участников в группах, a_{ij} – число участников из i-го региона в j-й группе. В цитированной работе также построена и исследована компромиссная аддитивная целевая функция. Она имеет следующий вид

$$F = \alpha_1 \frac{K_{p.тек}}{K_{p.min}} + \alpha_2 \frac{\Delta}{0{,}1\overline{R}} \to \min,$$

где $K_{p\ min}$ и $K_{p\ тек}$ – соответственно минимальный и текущий (расчетный) критерии равномерности, α_1 и α_2 – весовые экспертные оценки, в сумме равные единице, Δ – разность между наибольшим и наименьшим суммарным рейтингом двух групп.

В статье [13] предложен один из возможных вариантов последовательного алгоритма автоматизированного построения турнирных таблиц с оптимизацией по двум критериям. Блок-схема этого алгоритма приведена на рис. 14. В приведенной блок-схеме A – турнирная таблица ($A_{i\,j}$ – j-й игрок i-й группы), S = S\S$_d$ – исключение d-го игрока из S, |S| - мощность множества, т.е. количество игроков в нем, x *mod* y – операция деления x по модулю y, |i-1| – модуль значения i-1. По приведенному алгоритму был разработан программный код на языке *C#* в среде разработки Microsoft Visual Studio 2012.

На рис. 15 приведен один из результатов работы алгоритма.

В программу заложены дополнительные возможности добавления в таблицу при необходимости или удаление из нее по одному игроку, а также рандомизация ввода исходных данных. По предварительным оценкам компетентных членов судейской коллегии разработанный программный продукт удовлетворяет их требованиям по качеству автоматически созданных турнирных таблиц и позволяет многократно сократить время их формирования [22].

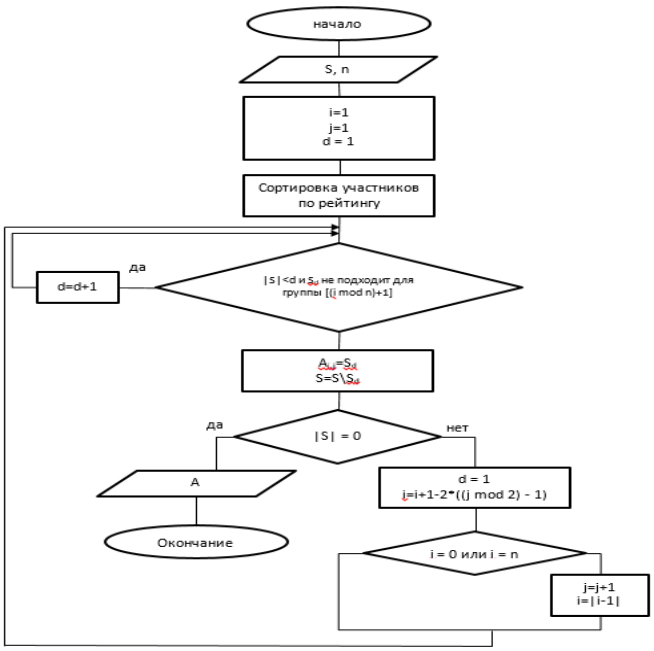

Рис. 14. Блок-схема алгоритма жеребьевки

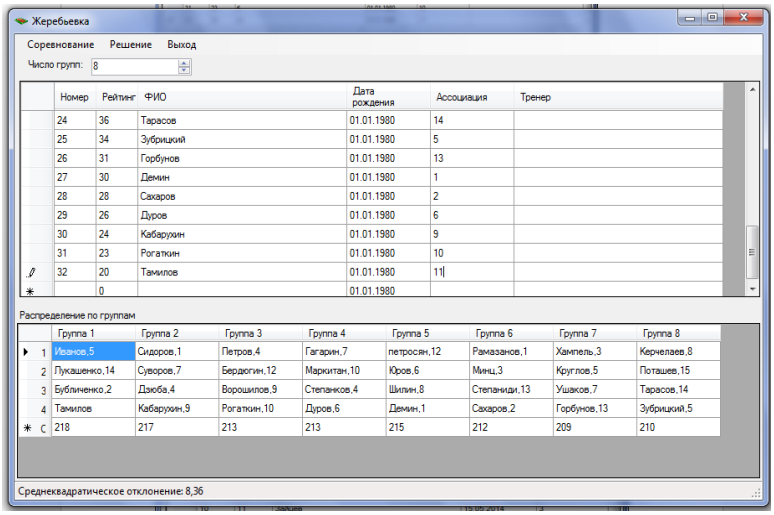

Рис. 15. Пример работы алгоритма жеребьевки

Общие выводы и заключение. Из приведенных на рис. 1 и 2 диаграмм распределений вопросов по билетам вытекает вывод, что последовательный алгоритм обладает существенным недостатком. Он состоит в том, что в случае распределения вопросов на большое число билетов, но с малым числом вопросов в билете, в конце работы программы могут формироваться билеты со значительной суммарной сложностью вопросов. С целью устранения этого недостатка разработан модифицированный вариант последовательного алгоритма и две модификации параллельного алгоритма наполнения билетов вопросами. Графические результаты

данных экспериментов, приведенные на рис. 3, 4 и 5, дают основание утверждать, что оптимизированный вариант параллельного алгоритма формирования билетов обладает наименьшей дисперсией и формирует билеты с практически равной сложностью. Подтверждающие результаты экспериментов, приведенные на рис. 6–11, показывают, что оптимизированный алгоритм обладает и меньшей временной сложностью. Выяснилось также, что при разбиении вопросов по темам временная сложность исследуемых алгоритмов увеличивается. Кроме того, выявилась четкая тенденция зависимости временной сложности алгоритмов от соотношения числа билетов и числа вопросов в них. При отсутствии распределения вопросов по темам для последовательного и параллельного алгоритмов наблюдается незначительный линейный рост временной сложности (не учитывая начальный участок). Оптимизированный же алгоритм в этом случае дает практически постоянные результаты. Таким образом, в данных исследованиях оптимизированный алгоритм проявил абсолютно лучшие результаты.

Общность и преемственность в алгоритмах формирования турнирных таблиц и последовательного распределения вопросов по билетам состоит не только в использовании графовых моделей, но и одних и тех же соотношений, приведенных на странице 5 данной статьи.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Кристофидес К.* Теория графов. Алгоритмический подход. – М.: Изд-во «Мир», 1978. – 432 с.
2. *Оре О.* Теория графов. – 2-е изд. – М.: Наука, Глав. ред. Физмат. лит-ры, 1980. – 336 с.
3. *Нечепуренко М.И., Попков В.К., Майнагашев С.М. и др.* Алгоритмы и программы решения задач на графах и сетях. – Новосибирск: Сиб. отд-ние, 1990. – 515 с.
4. *Башмаков А.И., Башмаков И.А.* Разработка компьютерных учебников и обучающих систем. – М.: Информационно-издательский дом «Филинь», 2003. – 616 с.
5. *Майоров А.Н.* Теория и практика создания тестов для системы образования. – М.: Интеллект-центр, 2001. – 296 с.
6. *Радчиков Н.П.* Тесты проверки знаний: этапы разработки. – Минск: РИВШ, 2007. – 30 с.
7. *Ерохин А.Л., Кольченко А.В., Патрах Т.Е., Чикина В.А.* Программный комплекс для генерации компьютерных тестирующих систем // Образование и виртуальность: Сб. научных трудов 6-й Международной конференции Украинской ассоциации дистанционного образования. – Харьков – Ялта: УАДО, 2002. – С. 323-327.
8. *Ерохин А.Л. Кольченко А.В., Струкова А.В.* Использование тестирующих систем в учебных заведениях и практических подразделениях ОВД // Право і безпека. – 2002. – № 3. – С. 128-130.
9. *Глушань В.М., Липало Н.Н, Малютин В.А.* Оптимизация тестовых заданий при контроле знаний // Вестник Таганрогского государственного педагогического института. Естественные науки. – 2007. – № 1. – С. 72-76.
10. *Глушань В.М., Афанасьев А.Ю.* Алгоритмы оптимизации тестовых заданий // Вестник Таганрогского государственного педагогического института. Физико-математические и естественные науки. – 2010. – № 1. – С. 128-136.
11. *Афанасьев А.Ю., Глушань В.М., Карелин В.П.* Исследование алгоритмов оптимизации тестовых заданий // Известия ЮФУ. Технические науки. – 2010. – № 12. – С. 203-211.
12. *Глушань В.М.* Критериальные особенности оптимального формирования турнирных таблиц // Тр. Конгресса по интеллектуальным системам и информационным технологиям «IS&IT'14». Научное издание в 4-х т. Т. 1. – М.: Физматлит, 2014. – С. 446-453.
13. *Глушань В.М, Кажаров А.А., Пономарев В.К*. Методы оптимального автоматизированного формирования турнирных таблиц // Программные продукты и системы. Международный научно-практический журнал. – 2015. – № 3 (111). – С. 226 -232.
14. *Глушань В.М., Зубрицкий А.В.* Полноперебoрный вариант формирования турнирных таблиц // Тр. Конгресса по интеллектуальным системам и информационным технологиям «IS&IT'16». Научное издание в 3-х т. Т. II. – Таганрог: Изд-во ЮФУ 2016. – С. 193-199.

15. *Глушань В.М., Зубрицкий А.В.* Алгоритм разбиения множества по его номеру на совокупность равномощных подмножеств // Известия ЮФУ. Технические науки. – 2018. – № 4 (198). – С. 59-65.
16. *Глушань В.М., Афанасьев А.Ю.* Оптимальный распределитель тестовых заданий. Свидетельство о государственной регистрации для ЭВМ № 2010612908.
17. *Глушань В.М., Кажаров А.А.* Программа оптимального размещения объектов в дискретном прямоугольном пространстве. Свидетельство о государственной регистрации программы для ЭВМ № 2015618325.
18. *Глушань В.М., А.Ю. Афанасьев, Н.И. Лященко*. Концепция создания автоматизированного рабочего места (АРМ) преподавателя // Вестник Таганрогского педагогического института. Физико-математические и естественные науки. – 2011. – № 1. – С. 108-111.
19. *Глушань В.М., А.Ю. Афанасьев, Н.И. Лященко.* Оптимизация вузовского образовательного процесса в среде автоматизированного обучения и контроля знаний // Вестник Таганрогского педагогического института. Физико-математические и естественные науки. – 2011. – № 1. – С. 111-116.
20. http://www.ystok.ru/tournament/ (дата посещения: 25.08.22).
21. https://www.qrz.ru/software/category/8.html (дата посещения: 25.08.22).
22. *Пенов Г.Г, Александров А.В., Зубарь Я.С., Кизилов А.В., Мазаев К.М., Пономарев В.К.* Настольный теннис. Сборник материалов для судей. – М., 2012.

REFERENCES

1. *Kristofides K.* Teoriya grafov. Algoritmicheskiy podkhod [Graph theory. Algorithmic approach]. Moscow: Izd-vo «Mir», 1978, 432 p.
2. *Ore O.* Teoriya grafov [Graph theory]. 2nd ed. Moscow: Nauka, Glav. red. Fizmat. lit-ry, 1980, 336 p.
3. *Nechepurenko M.I., Popkov V.K., Maynagashev S.M. i dr.* Algoritmy i programmy resheniya zadach na grafakh i setyakh [Algorithms and programs for solving problems on graphs and networks]. Novosibirsk: Sib. otd-nie, 1990, 515 p.
4. *Bashmakov A.I., Bashmakov I.A.* Razrabotka komp'yuternykh uchebnikov i obuchayushchikh system [Development of computer textbooks and training systems]. Moscow: Informatsionno-izdatel'skiy dom «Filin'», 2003, 616 p.
5. *Mayorov A.N.* Teoriya i praktika sozdaniya testov dlya sistemy obrazovaniya [Theory and practice of creating tests for the education system]. Moscow: Intellekt-tsentr, 2001, 296 p.
6. *Radchikov N.P.* Testy proverki znaniy: etapy razrabotki [Knowledge testing tests: stages of development]. Minsk: RIVSh, 2007, 30 p.
7. *Erokhin A.L., Kol'chenko A.V., Patrakh T.E., Chikina V.A.* Programmnyy kompleks dlya generatsii komp'yuternykh testiruyushchikh sistem [A software package for generating computer testing systems], *Obrazovanie i virtual'nost': Sb. nauchnykh trudov 6-y Mezhdunarodnoy konferentsii Ukrainskoy assotsiatsii distantsionnogo obrazovaniya* [Education and virtuality: A collection of scientific papers of the 6th International Conference of the Ukrainian Association of Distance Education]. Khar'kov – Yalta: UADO, 2002, pp. 323-327.
8. *Erokhin A.L. Kol'chenko A.V., Strukova A.V.* Ispol'zovanie testiruyushchikh sistem v uchebnykh zavedeniyakh i prakticheskikh podrazdeleniyakh OVD [The use of testing systems in educational institutions and practical departments of the Department of Internal Affairs], *Pravo i bezpeka* [Law and security], 2002, No. 3, pp. 128-130.
9. *Glushan' V.M., Lipalo N.N, Malyutin V.A.* Optimizatsiya testovykh zadaniy pri kontrole znaniy [Optimization of test tasks in the control of knowledge], *Vestnik Taganrogskogo gosudarstvennogo pedagogicheskogo instituta. Estestvennye nauki* [Bulletin of the Taganrog State Pedagogical Institute. Natural sciences], 2007, No. 1, pp. 72-76.
10. *Glushan' V.M., Afanas'ev A.Yu.* Algoritmy optimizatsii testovykh zadaniy [Algorithms for optimization of test tasks], *Vestnik Taganrogskogo gosudarstvennogo pedagogicheskogo instituta. Fiziko-matematicheskie i estestvennye nauki* [Bulletin of the Taganrog State Pedagogical Institute. Physical, mathematical and natural sciences], 2010, No. 1, pp. 128-136.
11. *Afanas'ev A.Yu., Glushan' V.M., Karelin V.P.* Issledovanie algoritmov optimizatsii testovykh zadaniy [Research of algorithms for optimization of test tasks], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2010, No. 12, pp. 203-211.

12. *Glushan' V.M.* Kriterial'nye osobennosti optimal'nogo formirovaniya turnirnykh tablits [Criteria features of optimal formation of tournament tables], *Tr. Kongressa po intellektual'nym sistemam i informatsionnym tekhnologiyam «IS&IT'14». Nauchnoe izdanie v 4-kh t. T. 1* [Proceedings of the Congress on Intelligent Systems and Information Technologies "IS&IT'14". Scientific edition in 4 vol. Vol. 1]. Moscow: Fizmatlit, 2014, pp. 446-453.
13. *Glushan' V.M, Kazharov A.A., Ponomarev V.K.* Metody optimal'nogo avtomatizirovannogo formirovaniya turnirnykh tablits [Methods of optimal automated formation of tournament tables], *Programmnye produkty i sistemy. Mezhdunarodnyy nauchno-prakticheskiy zhurnal* [Software products and systems. International Scientific and Practical Journal], 2015, No. 3 (111), pp. 226 -232.
14. *Glushan' V.M., Zubritskiy A.V.* Polnoperebornyy variant formirovaniya turnirnykh tablits [A full-fledged version of the formation of tournament tables], *Tr. Kongressa po intellektual'nym sistemam i informatsionnym tekhnologiyam «IS&IT'16». Nauchnoe izdanie v 3-kh t. T. II* [Proceedings of the Congress on Intelligent Systems and Information Technologies "IS&IT'16". Scientific edition in 3 vol. Vol. II]. Taganrog: Izd-vo YUFU 2016, pp. 193-199.
15. *Glushan' V.M., Zubritskiy A.V.* Algoritm razbieniya mnozhestva po ego nomeru na sovokupnost' ravnomoshchnykh podmnozhestv [Algorithm for splitting a set by its number into a set of equally powerful subsets], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2018, No. 4 (198), pp. 59-65.
16. *Glushan' V.M., Afanas'ev A.Yu.* Optimal'nyy raspredelitel' testovykh zadaniy. Svidetel'stvo o gosudarstvennoy registratsii dlya EVM № 2010612908 [The optimal distributor of test tasks. Certificate of state registration for computers No. 2010612908].
17. *Glushan' V.M., Kazharov A.A.* Programma optimal'nogo razmeshcheniya ob"ektov v diskretnom pryamougol'nom prostranstve. Svidetel'stvo o gosudarstvennoy registratsii programmy dlya EVM № 2015618325 [A program for optimal placement of objects in a discrete rectangular space. Certificate of state registration of the computer program No. 2015618325].
18. *Glushan' V.M., Afanas'ev A.Yu., Lyashchenko N.I.* Kontseptsiya sozdaniya avtomatizirovannogo rabochego mesta (ARM) prepodavatelya [The concept of creating an automated workplace (APM) of a teacher], *Vestnik Taganrogskogo pedagogicheskogo instituta. Fiziko-matematicheskie i estestvennye nauki* [Bulletin of the Taganrog Pedagogical Institute. Physical, mathematical and natural sciences], 2011, No. 1, pp. 108-111.
19. *Glushan' V.M., Afanas'ev A.Yu., Lyashchenko N.I.* Optimizatsiya vuzovskogo obrazovatel'nogo protsessa v srede avtomatizirovannogo obucheniya i kontrolya znaniy [Optimization of the university educational process in the environment of automated learning and knowledge control], *Vestnik Taganrogskogo pedagogicheskogo instituta. Fiziko-matematicheskie i estestvennye nauki* [Bulletin of the Taganrog Pedagogical Institute. Physical, mathematical and natural sciences], 2011, No. 1, pp. 111-116.
20. Available at: http://www.ystok.ru/tournament/ (accessed 25August 22).
21. Available at: https://www.qrz.ru/software/category/8.html (accessed 25August 22).
22. *Penov G.G, Aleksandrov A.V., Zubar' YA.S., Kizilov A.V., Mazaev K.M., Ponomarev V.K.* Nastol'nyy tennis. Sbornik materialov dlya sudey [Table tennis. Collection of materials for judges]. Moscow, 2012.

Статью рекомендовал к опубликованию д.т.н., профессор Я.Е. Ромм.

Глушань Валентин Михайлович – Южный федеральный университет; e-mail: gluval07@rambler.ru; г. Таганрог, Россия; кафедра САПР; профессор.

Glushan Valentin Mihailovich – Southern Federal University; e-mail: gluval07@rambler.ru; Taganrog, Russia; the department of computer aided design; professor.