

15. Ng A.Y., Jordan M.I., Weiss Y. On spectral clustering: analysis and an algorithm, *Proceedings of Neural Information Processing Systems (NIPS)*. 2002. Vol. 2 (3). pp. 857-864.
16. Caelli T., Kosinov S. An eigenspace projection clustering method for inexact graph matching, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2004, Vol. 26 (4), pp. 515-519.
17. Levashov S. Segmentation of a point cloud by data on laser scanning intensities, *Pattern Recognition and Image Analysis*, 2019, Vol. 29 (1), pp. 144-155.
18. van Leuken R.H., Symonova O., Veltkamp R.C., De Amicis R. Complex Fiedler Vectors for Shape Retrieval, *Structural, Syntactic, and Statistical Pattern Recognition*, 2008, Vol. 5342, pp. 167-176.
19. Tsourakakis C.E. Fast counting of triangles in large real networks without counting: Algorithms and laws, *IEEE International Conference on Data Mining (ICDM 2008)*, 2008, Vol. 0, pp. 608-617.
20. Bentley J.M. Multidimensional binary search trees used for associative searching, *Communication of the ACM*, 1975, Vol. 18, pp. 509-517.

Статью рекомендовал к опубликованию д.ф.-м.н., профессор Г.В. Куповых.

**Каркищенко Александр Николаевич** – Южный федеральный университет; e-mail: karkishalex@gmail.com; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 89034610539; кафедра высшей математики; профессор.

**Левашев Сергей Петрович** – e-mail: serzh.levashev@yandex.ru; 347928, г. Таганрог, ул. Чехова, 22; тел.: 89281997778; аспирант.

**Karkischenko Aleksandr Nikolaevich** – Southern Federal University; e-mail: karkishalex@gmail.com; 44? Nekrasovskiy, Taganrog, 347928, Russia; phone: +79034610539; the department of higher mathematic; professor.

**Levashev Sergey Petrovich** – e-mail: serzh.levashev@yandex.ru; 22, Chekhova street, Taganrog, 347928, Russia; phone: +79281997778; graduate student.

УДК 004.021

DOI 10.23683/2311-3103-2019-3-85-96

**В.М. Глушань, А.В. Зубрицкий**

### **ПРИМЕНЕНИЕ ПОЛНОГО И ЧАСТИЧНОГО ПЕРЕБОРА ДЛЯ РЕШЕНИЯ МАЛОРАЗМЕРНЫХ КОМБИНАТОРНЫХ ЗАДАЧ\***

*В настоящее время известно много комбинаторных задач, оптимальное решение которых теоретически можно найти, используя метод полного перебора. Однако количество решений таких задач экспоненциально увеличивается с ростом размерности задачи. Это исключает возможность быстрого нахождения оптимального решения методом полного перебора даже для задач среднего размера. Среди комбинаторных задач имеются и малоразмерные задачи. Это такие задачи, множество элементов в которых составляет несколько десятков. К таким задачам можно отнести задачи о назначениях и рюкзаке, а также задачу формирования турнирных таблиц. Быстрое совершенствование компьютерных технологий позволяет решать некоторые малоразмерные комбинаторные задачи в режиме реального времени. В статье рассматривается несколько подходов оптимального формирования турнирных таблиц, являющихся результатом жеребьевки. Важное значение это имеет для тех видов спорта, в которых принимает участие относительно большое количество игроков из разных регионов. В этих случаях цель жеребьевки состоит в том, чтобы рассеять игроков по разным группам так, чтобы все группы имели примерно одинаковый суммарный рейтинг и региональный фактор. При таком условии задача становится двухкритериальной. Если число участников находится в пределах двух десятков, то задачу еще можно решать полным перебором за приемлемое время. При числе участников турни-*

\* Работа выполнена при поддержке РФФИ (проект № 18-01-00041).

ра за пределами двух десятков полный перебор уже практически не применим. В статье предложено два комбинированных алгоритма формирования турнирных таблиц. Первый основан на эвристическом методе «змейки» и методе полного перебора. Второй алгоритм основан на предварительном формировании нескольких групп частичным перебором, а оставшихся групп – последовательным методом. Разработанные комбинированные алгоритмы позволяют существенно увеличивать число игроков в турнирной таблице при доступном времени их работы. Поскольку предложенные алгоритмы являются приближенными, в статье разработан метод оценки качества результатов их работы относительно оптимального решения. Проведено тестирование представленных алгоритмов, используя как случайно сгенерированные тесты, так и тесты на основе реальных рейтингов, проведено сравнение алгоритмов и даны рекомендации по их использованию.

*Малоразмерная комбинаторная задача; турнирная таблица; частичный и полный перебор; оптимальное решение.*

V.M. Glushan, A.V. Zubritckii

### APPLICATION OF EXHAUSTIVE AND PARTIAL SEARCH FOR SOLVING THE LOW-SIZED COMBINATOR PROBLEMS

*Currently, there are many combinatorial problems for which the optimal solution theoretically can be found using the brute force method. However, the number of solutions for such problems increases exponentially with the size of problems. This excludes the possibility of a quick search for optimal solutions using brute force, even for medium-sized problems. Among combinatorial problems there are also small-sized problems. These are problems with set of elements equal several dozen. These problems include assignment and backpack problems, as well as the problem of creating tournament tables. The rapid improvement of computer technology allows us to solve some small combinatorial problems in real time. The article presents several approaches to the optimal formation of tournament tables, which are the result of the draw. This is important for those sports that involve a relatively large number of players from different regions. In these cases, the purpose of the draw is to separate players into different groups so that groups have approximately the same total rating and regional factor. Under this condition, the problem becomes two-criteria. If the number of participants is within two dozen, then the problem can still be solved by exhaustive search in a reasonable time. With the number of participants in tournament beyond two dozen brute force is almost not applicable. Two combined algorithms for the formation of tournament tables are proposed. The first is based on the heuristic “snake” method and the brute force method. The second algorithm is based on the preliminary formation of several groups by partial enumeration, and the remaining groups by a sequential method. The developed combined algorithms can significantly increase the number of players in the standings with the permissible time of their work. Since the proposed algorithms are approximate, a method for assessing the quality of the results of their work with respect to the optimal solution has been developed. Testing of the presented algorithms was carried out using both randomly generated tests and tests based on real ratings, the algorithms were compared and recommendations on their use were given.*

*Low-sized combinatorial problem; tournament tables; exhaustive and partial search; optimal solution.*

**Введение.** Конец 70-х и начало 80-х гг. прошлого столетия ознаменовались переходом от изложения в учебной и научной литературе традиционных инженерных руководств по разделам вычислительной математики к нетрадиционному изложению комбинаторных задач. Как отмечено в предисловии к [1] это «привело к подлинному возрождению теории графов и комбинаторики, которые за несколько десятков лет трансформировались из разделов досуговой математики в первостепенный инструмент решения огромного числа задач». Это вызвано широким применением ЭВМ в самых различных областях деятельности человека. Теория графов и комбинаторика стали обязательными разделами книг по дискретной математике [2–6]. Следует отметить, что изложение основных положений комбинаторики теперь состоит не только в ответе на извечный вопрос «сколько», т.е. в подсчете

числа комбинаций на комбинаторных множествах. С постоянно растущим быстродействием и объемов внутренней памяти компьютеров появилась возможность решать переборные (перечислительные [7]) и оптимизационные (поисковые) комбинаторные задачи. Как следствие этого, появились исследования по созданию перечислительных алгоритмов (полного и частичного перебора) [1, 8–10].

В докомпьютерную эру, т.е. до того момента, пока компьютеры не стали использоваться повсеместно, реализация перечислительных алгоритмов осуществлялась аппаратными средствами, что объяснялось на порядки более высоким их быстродействием по сравнению с программными средствами. Было разработано значительное число аппаратных средств для реализации перечислительных алгоритмов, некоторые из которых представлены изобретениями<sup>1234</sup>.

Почти все комбинаторные задачи являются NP-трудными, а алгоритмы их решения имеют экспоненциальную сложность [4]. Один из подходов для решения таких комбинаторных задач состоит в разбиении полной последовательности комбинаторных соединений из заданного множества элементов на классы и исследовании этих классов параллельно на нескольких процессорах [11]. Для реализации этого подхода необходимо уметь выполнять нумерацию комбинаторных соединений, т.е. формировать их по заданному номеру из полной последовательности. Методы и алгоритмы для выполнения нумерации рассматриваются в работах [6, 8, 12–17].

В работах [11, 14] процесс нумерации перестановок предлагается использовать для ускорения решения комбинаторных задач путем распараллеливания на  $m$  одинаковых процессоров. При этом, если задача, например задача о назначениях, требует использования всех перестановок из  $n$ -элементного множества, то они разбиваются на блоки размером  $n!/m$ . Каждый такой блок перестановок генерируется соответствующим отдельным процессором параллельно и независимо от других блоков. Экспериментально показано, что время формирования всех перестановок при фиксированном  $n$  в этом случае растет линейно с увеличением числа процессоров  $m$ . Однако в указанных работах упущен учет важнейшего фактора, а именно, как растет абсолютное число перестановок, которое должен формировать каждый процессор с ростом числа элементов  $n$  исходного множества. Элементарные подсчеты показывают, что, например, при  $n=14$ , которые распределяются по 14 процессорам, каждый из них должен формировать по 6 227 020 800 перестановок. Если предположим, что за 1 секунду процессор формирует  $10^6$  перестановок, то ему потребуется 1,73 часа работы, а при распределении на 10 процессоров каждый из них должен формировать 8 717 829 120 перестановок, что потребует уже 2,42 часа. Если мы увеличим число элементов  $n$  в исходном множестве всего на 2, т.е. до 16, то каждый процессор должен будет формировать  $1,3 \cdot 10^{12}$  перестановок, и тогда каждому процессору при том же быстродействии придется работать 361 час.

Из приведенных подсчетов ясно, что простое распараллеливание не приводит к существенному сокращению экспоненциальной сложности комбинаторных задач. Среди задач комбинаторной оптимизации существуют такие, в которых число элементов не превышает нескольких десятков. К ним можно отнести задачу о рюкзаке, задачу о назначениях, задачу формирования турнирных таблиц. Область поиска при решении таких задач достаточно мала, поэтому для них возможен поиск оптимального решения методом частичного и даже полного перебора в режиме реального времени.

<sup>1</sup> Авторское свидетельство СССР на изобретение № 1383381.

<sup>2</sup> Авторское свидетельство СССР на изобретение № 1388887.

<sup>3</sup> Авторское свидетельство СССР на изобретение № 1397933.

<sup>4</sup> Авторское свидетельство СССР на изобретение № 1513467.

В данной работе проводится анализ как существующих многокритериальных методов, так и их модификаций построения турнирных таблиц для групп равного размера. Приводится разработка способа оценки качества решения относительно оптимального.

**Постановка задачи.** В статье [17] рассмотрены основные методы, используемые для распределения участников по группам при проведении спортивных соревнований. Большинство существующих на сегодняшний день методов являются эвристическими и предназначены для решения однокритериальных задач, то есть они, как правило, производят распределение игроков только по критерию рейтинга. Однако часто проводятся турниры, в которых каждая ассоциация может быть представлена несколькими игроками одновременно. В таких случаях необходимо вводить ассоциативный критерий, отвечающий за равномерность распределения участников от каждой ассоциации по турнирной таблице. Введение данного критерия позволяет снизить вероятность встреч между представителями одной ассоциации на начальных этапах соревнований. Таким образом, задача становится двухкритериальной, следовательно, усложняются способы построения и оценки её решений.

Для построения турнирных таблиц используются перечислительные процессы формирования упорядоченных (по принципу формирования, но неупорядоченные по использованию) комбинаторных соединений – разбиений, каждое из которых состоит из нескольких выборок с равным числом элементов (игроков). Порядок игроков в выборках также упорядочен по принципу формирования, но по применению порядок значения не имеет.

Рассматриваемые в статье методы ориентированы на использование перечислительных алгоритмов полного перебора разбиений [16] и частичного перебора разбиений [18].

**Описание сравниваемых методов. Алгоритм 1.** В статье [19] предложен алгоритм, который последовательно распределяет всех участников по группам. На первом этапе список участников упорядочивается по убыванию рейтинга. Затем в каждую группу распределяют по одному участнику с начала списка, то есть в первую группу участник с наибольшим рейтингом, во вторую второй и так далее. На следующем этапе происходит распределение участников согласно формулам:

$$\tilde{r} = \frac{(R - g_{i-1})m}{N - m(i-1)},$$

$$R = \sum_{i=1}^n r_i / m,$$

где  $R$  – средний суммарный рейтинг,  $r_i$  – значение рейтинга  $i$ -го игрока,  $m$  – число групп,  $N$  – общее количество игроков,  $g_{i-1}$  – сумма рейтингов игроков, распределенных в рассматриваемую группу на шаге  $i-1$ . Также учитывается число ассоциаций, уже распределенных в предполагаемую группу, т.е., если количество представителей данной ассоциации будет превышать некоторое пороговое значение таких же ассоциаций в предполагаемой группе, то участник будет распределен в другую группу.

**Алгоритм 2.** В статье [20] предложен лексикографический алгоритм формирования турнирных таблиц. Он основан на методе полного перебора и теории  $g(n)$ -разбиений. Алгоритм позволяет последовательно без повторений получить все возможные варианты распределения участников турнира на группы одинакового размера, а каждое распределение является упорядоченным  $g(n)$ -разбиением. Он позволяет выбрать из полученных решений рассматриваемой комбинаторной за-

дачи одно точное или серию решений, исходя из предъявляемых требований. В этом методе, лицо принимающее решение, может учитывать не только критерии рейтинга и ассоциации, но и другие. Данный лексикографический алгоритм предназначен для решения только малоразмерных задач с максимальным количеством участников не более 24, так как количество возможных вариантов решения экспоненциально растет с увеличением числа участников. Для определения оптимального решения программа ЭВМ использует двухкритериальную аддитивную целевую функцию. То есть для каждого варианта турнирной таблицы вычисляется значение целевой функции, а ее наименьшее значение соответствует оптимальному.

**Алгоритм 3.** В статье [21] предложен алгоритм сужения области поиска при формировании турнирных таблиц. В основе данного алгоритма лежит идея случайного поиска, то есть рассматриваются только те области решений, в которых вероятность нахождения оптимального решения является наибольшей. Алгоритм позволяет варьировать точность решения, путем изменения параметров глубины и ширины поиска. Параметр глубины влияет на количество областей, которые будут исследованы, а параметр ширины влияет на ширину каждой из этих областей. Увеличение этих параметров ведет к увеличению точности решения, а также к увеличению количества вычислений. Необходимо отметить, что при определенных значениях параметров результат работы данного алгоритма будет совпадать с результатами лексикографического алгоритма формирования турнирных таблиц, то есть будет являться полным перебором.

**Алгоритм 4.** Лексикографический алгоритм построения турнирных таблиц, приведенный в [6, 20], позволяет получить все возможные варианты таблиц. На практике часто встречаются ситуации, когда некоторая часть участников выделяется на фоне остальных участников, то есть значения рейтингов этих участников значительно превосходят значение среднего рейтинга. В таких случаях для получения наиболее оптимальной турнирной таблицы необходимо компенсировать вклад выделяющихся участников в значение рейтингового критерия. Для этого целесообразно построить предварительное распределение только для участников с наибольшими показателями рейтинга с помощью популярного эвристического алгоритма «змейка». Это позволит уменьшить область поиска решений, исключив из нее большое количество не оптимальных решений. Данный эвристический алгоритм позволяет учитывать только один критерий, то есть предварительное распределение не учитывает ассоциации участников, поэтому для окончательного формирования оптимальной турнирной таблицы необходимо использовать алгоритм, учитывающий оба критерия.

Для реализации этой идеи разработан комбинированный алгоритм, основанный на эвристическом алгоритме «змейка» и алгоритме полного перебора (алгоритм 2). Сначала необходимо выбрать количество участников, которое будет предварительно распределено. Для наибольшей эффективности предварительного распределения необходимо, чтобы это число было кратно числу групп, не превышало половину от числа всех участников, а число участников в каждой группе предварительного распределения было четным. Эти ограничения позволяют обеспечить достаточно хорошее усреднение участников с большими рейтингами. Далее нужно упорядочить всех участников по величине рейтинга по убыванию и теперь с помощью алгоритма «змейки» можно провести предварительное распределение. Для этого последовательно распределяем по одному участнику в группу, начиная с первой. После того, как в последнюю группу был распределен участник, необходимо сменить порядок распределения участников, то есть следующие участники также будут последовательно распределяться по одному в группу, но теперь начиная с последней и заканчивая первой. Если предварительное распределе-

ние не завершено, то следует менять порядок каждый раз, когда распределение производится в первую или последнюю группу. Теперь необходимо оптимальным образом распределить оставшихся участников в группы, частично сформированные после предварительного распределения. Для решения этой задачи воспользуемся лексикографическим алгоритмом формирования турнирных таблиц. Это позволит учитывать не только критерий распределения по рейтингу, но и по ассоциациям.

**Алгоритм 5.** Рассмотренные ранее алгоритмы основаны на последовательном распределении участников по группам, поэтому был разработан алгоритм последовательного формирования групп. На основе списка участников можно априори рассчитать идеальные условия распределения участников, то есть средний рейтинг для каждой группы и максимальное количество представителей от каждой ассоциации. Если сложить рейтинги всех игроков и разделить полученное значение на количество групп, получится идеальный средний рейтинг. Если для каждой ассоциации разделить число её представителей на количество групп и округлить до целого числа в большую сторону, получим ограничения количества участников от каждой ассоциации в одной группе.

Теперь необходимо построить все возможные варианты формирования первой группы и выбрать оптимальный, для этого нужно получить все возможные сочетания количества игроков одной группы из всех участников. Оптимальным будет являться тот вариант, для которого выполняются ограничения на максимальное количество представителей от ассоциации и суммарное значение рейтингов участников полученной группы является наиболее близким к оптимальному. Аналогичным образом из оставшихся участников можно формировать оставшиеся группы, однако при этом в последние группы могут попадать игроки из одной и той же ассоциации, а значение средних рейтингов для этих групп может быть сильно удалено от идеального. Поэтому был разработан комбинированный алгоритм, который предварительно формирует несколько групп, а затем используется лексикографический алгоритм для того, чтобы достроить турнирную таблицу. Это позволяет увеличить точность, а также снизить время получения решения.

**Метод оценки решений.** Для решения и оценки однокритериальных задач существует развитый математический аппарат, который давно успешно используется. Однако в случае многокритериальных задач необходимо принимать во внимание также физическую природу и размерность частных критериев. В рассматриваемой задаче ассоциативный критерий и критерий рейтинга являются независимыми, поэтому можно представить целевую функцию как суперпозицию частных критериев с весовыми коэффициентами. Такой вид целевой функции позволяет добавлять дополнительные критерии, которые могут быть важны при оценке решений. Весовые коэффициенты дают возможность эксперту или лицу, принимающему решение, изменить важность критериев. Это необходимо, например, когда количество участников от каждой ассоциации относительно мало, например, 1–2 от каждой ассоциации. Тогда целевая функция примет следующий вид:

$$F = \alpha \frac{A}{A_{\max}} + \beta \frac{P}{P_{\max}} \rightarrow \min, \quad (1)$$

где  $A$  и  $P$  – это значение ассоциативного и рейтингового критериев соответственно для заданного распределения,  $A_{\max}$  и  $P_{\max}$  – это теоретически наибольшие значения критериев из всех возможных для заданных участников,  $\alpha$  и  $\beta$  – это весовые коэффициенты, в данном случае они принимают значение 0,5, придавая критериям одинаковую степень важности. Значения  $A$  и  $P$  вычисляются по следующим формулам:

$$A = \sum_i \sum_j (a_{ij})^2,$$

$$P = \frac{b_{\max} - b_{\min}}{b_{\text{average}}}, \quad (2)$$

где  $a_{ij}$  – число вхождений  $i$ -й ассоциации в  $j$ -ю группу,  $b_{\max}$  и  $b_{\min}$  – сумма рейтингов участников для группы с наибольшим общим рейтингом и наименьшим соответственно,  $b_{\text{average}}$  – среднее значение сумм рейтингов.

Необходимо отметить, что критерии имеют различную природу, откуда следует, что они обладают различной размерностью и областью изменения значений. Для установления связи между частными критериями в таких случаях следует использовать принцип равенства. Он предполагает, что каждый частный критерий в нормированном виде должен изменять свои численные значения в пределах от 0 до 1 и быть безразмерным. Поэтому в формуле (1) производится нормирование частных критериев к своим максимальным значениям [22].

Чтобы вычислить нормировочный коэффициент для рейтингового критерия, необходимо смоделировать наихудшее распределение по рейтингу. В таком распределении в одной группе должны быть участники с наибольшими значениями рейтингов, в другой с наименьшими, тогда  $b_{\max}$  принимает наибольшее значение из возможных, а  $b_{\min}$  соответственно наименьшее. Подставив полученные значения в формулу (2), найдем наибольшее возможное значение рейтингового критерия для заданных участников.

Чтобы вычислить нормировочный коэффициент для ассоциативного критерия, необходимо чтобы группы состояли только из представителей одной ассоциации. Если невозможно распределить участников данным образом, тогда следует максимизировать количество представителей от одной ассоциации в каждой группе. Следуя описанной выше логике, получим формулу для вычисления наибольшего значения ассоциативного критерия.

$$A_{\max} = \sum_i \left[ (a_i \operatorname{div} n) \cdot n^2 + (a_i \operatorname{mod} n)^2 \right],$$

где  $a_i$  – количество представителей от  $i$ -й ассоциации,  $n$  – число участников в одной группе,  $\operatorname{div}$  – оператор получения целой части от деления,  $\operatorname{mod}$  – оператор получения остатка от деления.

Из приведенных выше формул следует, что значения частных критериев нормированы, то есть изменяются от 0 до 1, следовательно, и значения целевой функции (1) должны находиться в пределах от 0 до 1, но не обязательно принимают эти крайние значения. Для удобства анализа результатов работы алгоритмов проведем нормировку целевой функции к её наибольшему (наихудшему) значению, тогда в наихудшем случае её значение будет равняться 1.

Если для оценки результатов работы алгоритмов использовать абсолютную разность значений целевых функций, то она не в полной мере отразит степень отступления решения друг от друга [22]. Важным также является и то, какую долю эта разность составляет от диапазона изменения целевых функций. Максимальное значение целевой функции равняется 1, так как мы провели её нормирование, а минимальное будет равняться значению целевой функции при решении

задачи с помощью лексикографического алгоритма, так как он позволяет получить точное решение. Следовательно, ухудшение решения определяется по следующей формуле:

$$dF = (F_i - F_2)/(1 - F_2). \quad (3)$$

$F_i$  и  $F_2$  – значение целевых функций для анализируемого алгоритма и лексикографического алгоритма соответственно.

**Результаты экспериментов.** Для проведения тестирования алгоритмов целесообразно ограничить число участников 16-ю, а число групп – 4-мя. Это вызвано необходимостью находить оптимальное распределение участников за ограниченный период времени. С увеличением количества участников экспоненциально увеличивается количество возможных решений для алгоритма 2, следовательно, увеличивается время вычислений. При данных параметрах количество возможных решений, получаемых алгоритмом 2, превышает 2 миллиона, но время работы при этом является приемлемым.

Таблица 1

**Результаты тестов**

№ теста	F1	F2	F3	F4	F5	F6	F7	F8
1	0,066	0,004	0,423	0,273	0,209	0,099	-	-
2	0,039	0,008	0,457	0,329	0,248	0,035	-	-
3	0,108	0,007	0,304	0,212	0,119	0,088	-	-
4	0,318	0,199	0,498	0,408	0,291	0,328	0,414	0,305
5	0,313	0,200	0,501	0,434	0,316	0,344	0,412	0,328
6	0,305	0,253	0,553	0,407	0,360	0,355	0,435	0,317
7	0,457	0,385	0,529	0,501	0,409	0,712	0,639	0,451
8	0,356	0,295	0,683	0,503	0,394	0,350	0,465	0,385
Число комбинаций	1	2627625	56	392	12320	454	3178	74456

Таблица 2

**Ухудшение качества решений относительно оптимального**

№ теста	F1	F2	F3	F4	F5	F6	F7	F8
1	6%	0%	42%	27%	21%	10%	-	-
2	3%	0%	45%	32%	24%	3%	-	-
3	10%	0%	30%	20%	11%	8%	-	-
4	15%	0%	37%	26%	11%	16%	27%	13%
5	14%	0%	38%	29%	15%	18%	27%	16%
6	7%	0%	40%	21%	14%	14%	24%	9%
7	12%	0%	23%	19%	4%	53%	41%	11%
8	9%	0%	55%	29%	14%	8%	24%	13%



В табл. 1 представлены абсолютные значения целевой функции и число получаемых вариантов распределения участников для каждого алгоритма, а в табл. 2 ухудшение качества решения относительно оптимального, рассчитываемое по формуле (3).  $F_1$  – значение целевой функции для алгоритма 1,  $F_2$  – значение целевой функции для алгоритма 2,  $F_3, F_4, F_5$  – значение целевой функции для алгоритма 3 с параметрами глубины и ширины  $d=1$  и  $w=0$ ,  $d=1$  и  $w=3$ ,  $d=2$  и  $w=0$  соответственно,  $F_6$  – значение целевой функции для алгоритма 4,  $F_7, F_8$  – значение целевой функции для алгоритма 5 с предварительным распределением двух и одной группы соответственно.

В первых трех тестовых заданиях решалась однокритериальная задача построения турнирных таблиц только по критерию рейтинга, то есть без учета ассоциативного критерия. В качестве значений рейтингов использовались случайные величины от 0 до 300. Алгоритм 5 в представленном виде не позволяет решать однокритериальные задачи в отличие от остальных алгоритмов, поэтому в табл. 1 и табл. 2 нет значений целевой для этих тестовых заданий. В остальных тестовых заданиях решалась двухкритериальная задача и использовались реальные данные, чтобы максимально приблизить задачу к реальной.

Из полученных данных, очевидно, что алгоритм 2 позволяет получить наилучший результат, при этом качество получаемых решений в большинстве случаев значительно лучше по сравнению с другими алгоритмами. Также данный алгоритм позволяет вносить изменения в вид целевой функции, например, изменять и добавлять различные критерии оптимальности, что позволяет решать широкий класс различных задач. Так как данный алгоритм основан на методе полного перебора, то к его недостаткам относится избыточность и большое время работы.

Результаты работы алгоритма 1 на 3–14 % хуже оптимальных. Он обладает высокой скоростью построения решений, что позволяет решать задачи большой размерности за короткое время с очень хорошим качеством решений. Необходимо отметить, что главным недостатком алгоритма 1 является неравномерность получаемых решений, то есть значение среднего рейтинга последней группы может значительно отличаться от аналогичных значений остальных групп. Это значит, что результаты работы алгоритма сильно зависят от исходных данных.

Результаты алгоритма 3 напрямую зависят от выбора параметров ширины и глубины поиска. Увеличение любого из них способствует увеличению точности решения, но вместе с этим быстро растет и количество возможных вариантов решения. Данный алгоритм показывает плохие результаты при решении однокритериальных задач по сравнению с остальными алгоритмами. В случае двухкритериальных задач изменение параметра ширины с 0 до 3 позволяет улучшить качество решений на 2–26 %, увеличивая время вычислений в 7 раз, а изменение параметра глубины с 1 до 2 позволяет улучшить качество решений на 19–41 %, увеличивая время вычислений в 220 раз. Данный алгоритм с параметрами глубины  $d=2$  и ширины  $w=0$  дает ухудшение решения 4–15 % для двухкритериальных задач.

Алгоритм 4 позволяет получить достаточно хорошие результаты при ухудшении качества решения на 3–18 % относительно оптимального. Но при этом в тестовом задании №7 значение целевой функции для полученного решения слишком далеко от оптимального, то есть ухудшение получилось очень большим. Это связано с тем, что в результате предварительного распределения эвристическим методом «змейки» в одни группы попали большое количество участников из одинаковых ассоциаций.

Результаты работы алгоритма 5 показали, что при увеличении количества предварительно формируемых групп только ухудшает качество получаемых решений. Также как и в алгоритме 1 в данном алгоритме наблюдается неравномерность получаемых решений, то есть последняя группа дает очень большой вклад в значение целевой функции. В случае одной предварительно формируемой группы ухудшение составляет 9–16 % от оптимального, а также возможных решений равняется 74456.

**Рекомендации по использованию.** Для решения малоразмерных комбинаторных задач, таких как построение турнирных таблиц с количеством участников не более 20 целесообразно использовать алгоритм 2. Количество всех возможных вариантов решений не превышает нескольких миллионов, что позволяет быстро получать оптимальное решение на современных ПК.

Для построения турнирных таблиц средней размерности с количеством участников не более 50 необходимо использовать алгоритм 1, 3 и 4. Алгоритм 1 хорошо подходит в ситуациях, когда значения рейтингов участников монотонно убывают и ассоциации распределены равномерно в списке участников. Алгоритм 4 является наилучшим, когда в списке участников можно выделить явных фаворитов, у которых значение рейтинга намного выше среднего. В остальных случаях можно воспользоваться алгоритмом 3 с параметрами глубины и ширины  $d = 1$  и  $w = 0$ , так как с увеличением числа участников растет количество квазиоптимальных решений, следовательно, увеличивается вероятность нахождения решения близкого к оптимальному.

Для построения турнирных таблиц большой размерности с количеством участников свыше 50 в режиме реального времени позволяет только алгоритм 1, так как он имеет самую низкую временную сложность среди рассмотренных алгоритмов.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Липский В.* Комбинаторика для программистов: пер. с польск. – М.: Мир, 1988. – 213 с.
2. *Новиков Ф.А.* Дискретная математика: учебник для вузов. – 2-е изд. Стандарт третьего поколения. – СПб.: Питер, 2014. – 432 с.
3. *Хаггарти Р.* Дискретная математика для программистов. – 2-е изд., испр.: пер. с англ. / под ред. С.А. Кулешова. – М.: Техносфера, 2012. – 400 с.
4. *Андерсен Дж.* Дискретная математика и комбинаторика: пер. с англ. – М.: ООО «И.Д. Вильямс», 2016. – 960 с.
5. *Гладков Л.А., Курейчик В.В., Курейчик В.М.* Дискретная математика: учебник. – М.: Физматлит, 2014. – 496 с.
6. *Хаггард Г., Шлифф Дж., Уайтсайдс С.* Дискретная математика для программистов: учеб. пособие: пер. с англ. – М.: БИНОМ. Лаборатория знаний, 2010. – 627 с.
7. *Стенли Р.* Перечислительная комбинаторика. Деревья, производящие функции и симметрические функции // *Enumerative Combinatorics. Vol. 2.* – М.: Мир, 2009. – 767 с.
8. *Курейчик В.М., Глушань В.М., Щербаков Л.И.* Комбинаторные аппаратные модели и алгоритмы в САПР. – М.: Радио и связь», 1990. – 216 с.
9. *Глушань В.М., Пупков М.И., Щербаков Л.И.* Алгоритмы формирования упорядоченных  $r$ -выборок // *Кибернетика.* – 1988. – № 1. – С. 129-131.
10. *Глушань В.М., Левин И.П., Щербаков Л.И.* Аппаратурное формирование случайной последовательности сочетаний единичных сигналов // *Известия вузов. Приборостроение.* – 1987. – Т. XXX, № 4. – С. 37-42.
11. *Тимошевская Н.Е.* Разработка и исследование параллельных комбинаторных алгоритмов // *Прикладная дискретная математика.* – 2009. – № 2 (4). – С. 97-103.
12. *Тимошевская Н.Е.* О нумерации перестановок и сочетаний для организации параллельных вычислений в задачах проектирования управляющих систем // *Известия Томского политехнического университета.* – 2004. – Т. 307, № 6. – С. 18-20.
13. *Герасимов В.А.* Генерация случайных сочетаний. Генерация сочетаний по его порядковому номеру // *RSDN Magazine.* – 2010. – № 3.
14. *Тимошевская Н.Е.* Параллельная генерация сочетаний и перестановок // *Вторая Сибирская школа-семинар по параллельным вычислениям / под ред. проф. А.В. Старченко.* – Томск: Изд-во Том. ун-та, 2004. – 108 с.

15. Рябко Б.Я. Быстрая нумерация комбинаторных объектов // Дискретная математика. – 1998. – Т. 10. – С. 101-119.
16. Глушань В.М., Зубрицкий А.В. Алгоритм разбиения множества по его номеру на совокупность равномошных подмножеств // Известия ЮФУ. Технические науки. – 2018. – № 4 (198). – С. 59-65.
17. Зубрицкий А.В. Развитие средств информационной поддержки принятия решений в социально-технологических системах // Сб. трудов XIV Всероссийской научной конференции молодых ученых, аспирантов и студентов «Информационные технологии, системный анализ и управление» (ИТСАУ-2016). – Таганрог: Изд-во ЮФУ, 2016. – Т. 1. – 339 с.
18. Глушань В.М., Зубрицкий А.В. Теоретическое обоснование алгоритма формирования упорядоченных разбиений с равномошными неповторными выборками // Тр. конгресса по интеллектуальным системам и информационным технологиям «IS&IT'17». Научное издание в 3-х т. Т. 2. – Таганрог: Изд-во Ступина С.А. 2017. – С. 104-112.
19. Глушань В.М., Кажаров А.А., Пономарев В.К., Данильченко В.И. Развитие информационного обеспечения региональных и международных спортивных соревнований // Тр. конгресса по интеллектуальным системам и информационным технологиям «AIS-IT'14». Научное издание в 4-х т. Т. 3. – М.: Физматлит, 2014.
20. Glushan V., Zubritckii A., Lozovoy A. Formation of proximate two-criterion tournament tables // Advances in Computer Science Research (ACSR). – Vol. 72. – P. 160-163.
21. Глушань В.М., Зубрицкий А.В. Сужение области поиска решений для некоторых многокритериальных комбинаторных задач // Тр. конгресса по интеллектуальным системам и информационным технологиям «IS&IT'18». Научное издание в 3-х т. Т. 1. – Таганрог: Изд-во Ступина А.С., 2018. – С. 272-279.
22. Брахман Т.Р. Многокритериальность и выбор альтернативы в технике. – М.: Радио и связь, 1984. – 288 с.

#### REFERENCES

1. Lipskiy V. Kombinatorika dlya programistov [Combinatorics for programmers]. Moscow: Mir, 1988, 213 p.
2. Novikov F.A. Diskretnaya matematika: uchebnik dlya vuzov [Discrete mathematics: Textbook for university students]. 2<sup>nd</sup> ed. 3<sup>rd</sup> generation standart]. Saint Petersburg: Piter, 2014, 432 p.
3. Haggarti R. Diskretnaya matematika dlya programistov [Discrete mathematics for programmers]. 2<sup>nd</sup> edition, corrected: translated from English, ed. by Kuleshov S.A. Moscow: Tehnosfera, 2012, 400 p.
4. Andersen J. Diskretnaya matematika i kombinatorika [Discrete mathematics and Combinatorics]. Moscow: ООО «I.D. Williams», 2016, 960 p.
5. Gladkov L.A., Kureychik V.V., Kureychik V.M. Diskretnaya matematika: uchebnik [Discrete mathematics: textbook]. Moscow: Phymathlib, 2014, 496 p.
6. Haggard G., Shlipf J., Uaytsayds S. Diskretnaya matematika dlya programistov: uchebnoe posobie [Discrete mathematics for programmers: textbook]. Moscow: BINOM. Laboratoriya znaniy, 2010, 627 p.
7. Stenli R. Perechislitel'naya kombinatorika. Derev`ya, proizvodyashchie funktsii i simmetricheskie funktsii [Enumerative Combinatorics. Trees generating functions and symmetric functions], *Enumerative Combinatorics*. Vol. 2. Moscow: Mir, 2009, 767 p.
8. Kureychik V.M., Glushan V.M., Shcherbakov L.I. Kombinatornye apparatnye modeli i algoritmy v SAPR [Combinatorial hardware models and algorithms in CAD]. Moscow: Radio i svyaz', 1990, 216 p.
9. Glushan V.M., Pupkov M.I., Shcherbakov L.I. Algoritmy formirovaniya uporyadochennykh r-vyborok [Algorithms for the formation of ordered r-samples], *Kibernetika* [Cybernetics], 1988, No. 1, pp. 129-131.
10. Glushan V.M., Levin I.P., Shcherbakov L.I. Apparturnoe formirovanie sluchaynoy posledovatel'nosti sochetaniy edinichnykh signalov [Apparatus formation of a random sequence of combinations of single signals], *Izvestiya vuzov. Priborostroenie* [Bulletin of universities. Instrument making], 1987, Vol. XXX, No. 4, pp. 37-42.
11. Timoshevskaya N.E. Razrabotka i issledovanie parallel'nykh kombinatornykh algoritmov [Development and research of parallel combinatorial algorithms], *Prikladnaya diskretnaya matematika* [Applied discrete mathematics], 2009, No. 2 (4), pp. 97-103.

12. *Timoshevskaya N.E.* О нумерации перестановок и сочетаний для организации параллельных вычислений в задачах проектирования управляющих систем [On the numbering of permutations and combinations for the organization of parallel computations in problems of designing computer systems] *Izvestiya Tomskogo politekhnicheskogo universiteta* [Bulletin of the Tomsk polytechnic university], 2004, Vol. 307, No. 6, pp. 18-20.
13. *Gerasimov V.A.* Generatsiya sluchaynykh sochetaniy. Generatsiya sochetaniy po ego poryadkovomu nomeru [Generating random combinations. Generation of combinations by its serial number] *RSDN jurnal* [RSDN Magazine], 2010, Vol. 3.
14. *Timoshevskaya N.E.* Parallel'naya generatsiya sochetaniy i perestanovok [Parallel generation of combinations and permutations], *Vtoraya Sibirskaya shkola-seminar po parallel'nykh vychisleniyam* [Second Siberian school-workshop on parallel computing]. Tomsk: Tomsk university publishing, 2004, 108 p.
15. *Ryabko B.Ya.* Bystraya numeratsiya kombinatornykh ob'ektov [Fast numbering of combinatorial objects], *Diskretnaya matematika* [Discrete mathematics], 1998, Vol. 10, pp. 101-119.
16. *Glushan V.M., Zubritskiy A.V.* Algoritm razbiveniya vnogetstva po ego nomeru na sovokupnost' ravnomoshchnykh podmnozhestv [An algorithm for dividing a set by its number into a collection of equipotential subsets], *Izvestiya YuFU. Tehnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2018, No. 4, (198), pp.59-65.
17. *Zubritskii A.V.* Razvitie sredstv informatsionnoy podderzhki prinyatiya resheniy v social'no-tehnologicheskikh sistemakh [Development of information support decision-making tools in social and technological systems], *Sbornik trudov 14 Vserossiyskoy nauchnoy konferentsii molodykh uchenykh, aspirantov i studentov «Informatsionnye tekhnologii, sistemnyy analiz i upravlenie»* [Proceedings of the XIV All-Russian Scientific Conference of young scientists, graduate students and students "Information technology, systems analysis and management"]. Taganrog: SFEDU, 2016, Vol. 1, 339 p.
18. *Glushan V.M., Zubritskiy A.V.* Teoreticheskoe obosnovanie algoritma formirovaniya uporyadochenntkh razbieniy s ravnomoshchnymi bespovtornymi vyborkami [Theoretical substantiation of the algorithm of formation of ordered divisions with equipped unbeatable selections], *Trudt kongressa po intellektual'nykh sistemam i informatsionnym tekhnologiyam «IS&IT'17»*. Nauchnoe izdanie v 3 t. [Proceedings of the Congress on Intelligent Systems and Information Technologies «IS&IT'17». Scientific publication in 3 vol. Vol. 2]. Taganrog: Izdatel'stvo Stupina S.A., 2017, pp. 104-112.
19. *Glushan V.M., Kazharov A.A., Ponomarenko V.K., Danil'chenko V.I.* Razvitie sredstv informatsionnogo obespecheniya regional'nykh i mezhdunarodnykh sportivnykh sorevnovaniy [Development of information support for regional and international sports competitions], *Trudy kongressa po intellektual'nykh sistemam i informatsionnym tekhnologiyam «AIS-IT'14»*. Nauchnoe izdanie v 4 t. [Proceedings of the Congress on Intelligent Systems and Information Technology «AIS-IT'14». Scientific publication in 4 vol. Vol. 3]. Moscow: Physmatlit, 2014.
20. *Glushan V., Zubritskiy A., Lozovoy A.* Formation of proximate two-criterion tournament tables, *Advances in Computer Science Research (ACSR)*, Vol. 72, pp. 160-163.
21. *Glushan V.M., Zubritskiy A.V.* Suzhenie oblasti poiska resheniy nekotorykh mnogokriterial'nykh kombinatornykh zadach [Narrowing of search area of solutions for some multi-criteria combinatorial problems], *Trudy kongressa po intellektual'nykh sistemam i informatsionnym tekhnologiyam «IS&IT'18»*. Nauchnoe izdanie v 3 t. [Proceedings of the Congress on Intelligent Systems and Information Technologies «IS&IT'18». Scientific publication in 3 vol. Vol. 1]. Taganrog: Izd-vo Stupina S.A., 2018, pp. 272-279.
22. *Brahman T.R.* Mnogokriterial'nost' i vybor al'ternativy v tekhnike [Multiple criteria and choice of alternatives in engineering]. Moscow: Radio I svyaz', 1984, 288 p.

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

**Глушань Валентин Михайлович** – Южный федеральный университет; e-mail: gluval07@rambler.ru; 347928, г. Таганрог, пер. Некрасовский, 44; кафедра САПР; профессор.

**Зубрицкий Артур Валериевич** – e-mail: zubr\_artur@bk.ru; кафедра САПР; аспирант.

**Glushan Valentin Mihailovich** – Southern Federal University; e-mail: gluval07@rambler.ru; 44, Nekrasovskiy, Taganrog, 347928, Russia; the department of computer aided design; professor.

**Zubrickii Artur Valerievich** – e-mail: zubr\_artur@bk.ru; the department of computer aided design; postgraduate student.