20. *Glushko A.A.* Priborno-tekhnologicheskoe modelirovanie v sisteme TCAD Sentaurus. Metodicheskie ukazaniya k vypolneniyu laboratornykh rabot po distsipline "Avtomatizatsiya proektirovaniya elektronnykh sredstv" [Instrument-technological modeling in TCAD Sentaurus system. Methodical instructions to performance of laboratory works on discipline" automation of design of electronic means"]. Moscow, 2015.

Статью рекомендовал к опубликованию д.т.н., профессор И.Б. Фоминых.

Зинченко Людмила Анатольевна – МГТУ им. Н.Э. Баумана; e-mail: lzinchenko@bmstu.ru; 105005, г. Москва, ул. 2-я Бауманская, 5; тел. 84992636553; д.т.н.; профессор.

Никитин Илья Владимирович – студент

Верстов Владимир Александрович – e-mail: v.verstov@gmail.com; ассистент.

Гусев Михаил Владимирович – студент.

Бачурин Антон Сергеевич – студент.

Дмитриев Виктор Евгеньевич – студент.

Сорокин Борис Сергеевич – МГУ им. М.В. Ломоносова; e-mail: me@mralin.ru; 119991, г. Москва, Ленинские горы, 1; аспирант.

Zinchenko Lyudmila Anatol'evna – Bauman Moscow State Technical University; e-mail: lzinchenko@bmstu.ru; 105005, Moscow, 2nd Baumanskaya, 5; phone: +74992636553; dr. of eng. sc.; professor.

Nikitin Ilya Vladimirovich – student.

Verstov Vladimir Alexandrovich – e-mail: v.verstov@gmail.com; assistant.

Gusev Michail Vladimirovich – student.

Bachurin Anton Sergeevich – student.

Dmitriev Victor Evgen'evich – student.

Sorokin Boris Sergeevich – Moscow State University, e-mail: me@mralin.ru; 119991, Leninskie Gory, 1, Moscow; postgraduate student.

УДК 004.021 DOI 10.23683/2311-3103-2018-4-58-66

В.М. Глушань, А.В. Зубрицкий

АЛГОРИТМ РАЗБИЕНИЯ МНОЖЕСТВА ПО ЕГО НОМЕРУ НА СОВОКУПНОСТЬ РАВНОМОЩНЫХ ПОДМНОЖЕСТВ*

Представлены результаты разработки алгоритма формирования (r, n)-разбиений по их номерам в лексикографической последовательности всех возможных комбинаций. На сегодняшний день существуют различные задачи комбинаторной оптимизации, для которых комбинаторное соединение типа (r, n)-разбиения может являться оптимальным решением. Теоретически оптимальное решение можно найти полным перебором всех (r, n) -разбиений, но количество возможных вариантов экспоненциально растет с увеличением числа элементов. Следовательно, для задач большой размерности необходимо сократить время вычислений, то есть сократить число возможных решений или организовать параллельный вычислительный процесс. В статье приведено обоснование формулы, позволяющей вычислить точное количество всех возможных неповторяющихся (r, n)-разбиений. Опираясь на данную формулу, предложен алгоритм построения разбиений по заданному номеру. Он позволяет проанализировать область решений и использовать методы случай-

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

ного поиска решений. Работа алгоритма рассмотрена на примере. Разработан лексикографический алгоритм построения (r, n)-разбиений, который является модифицированной комбинацией двух алгоритмов: предложенного ранее рекурсивно-лексикографического алгоритма и алгоритма построения разбиений по заданному номеру. Он предназначен для решения задач оптимизации на комбинаторных объектах с сужаемыми областями поиска для тех случаев, когда мы заранее можем предсказать области оптимальных решений. Таким образом, алгоритм позволяет использовать частичный перебор вместо полного, то есть проводить исследования перспективных областей на наличие оптимального или близкого к оптимальному решения. Данный алгоритм, позволяет решать задачи большей мощности разбиваемого множества, чем рекурсивно-лексикографический алгоритм. Используя приведенную формулу для подсчета числа разбиений, можно разбить область решений на необходимое число подобластей и для каждой из них параллельно использовать предложенный алгоритм, реализовав таким образом параллельный вычислительный процесс.

Множества; комбинаторное множество; лексикографическая последовательность; оптимизация.

V.M. Glushan, A.V. Zubritckii

AN ALGORITHM FOR DIVIDING A SET BY ITS NUMBER INTO A COLLECTION OF EQUIPOTENTIAL SUBSETS

This article presents the results of the development of the algorithm for formation (r, n)-partitions by their numbers in the lexicographic sequence of all possible combinations. Today, there are various combinatorial optimization problems for which a combinatorial compound of the (r, n)-partition type can be the optimal solution. Theoretically, the optimal solution can be found by exhaustive search of all (r, n)-partitions, but the number of possible variants exponentially increasing with an increase in the number of elements. Consequently, for large-scale problems, it is necessary to reduce the computation time, that is, to reduce the number of possible solutions, or to organize a parallel computational process. The article presents the justification of the formula, which allows to calculate the exact number of all possible non-repeating (r, n) -partitions. Based on this formula, an algorithm for constructing partitions according to a given number is proposed. It allows us to analyze the scope of solutions and to use the methods of random search for solutions. The operation of the algorithm is considered on example. A lexicographic algorithm for constructing (r, n) -partitions has been developed, which is a modified combination of two algorithms: the previously proposed recursive-lexicographic algorithm and the algorithm for constructing partitions according to a given number. It is intended for solving optimization problems on combinatorial objects with narrowing search areas for those cases where we can predict the areas of optimal solutions in advance. Thus, the algorithm allows to use the partial search instead of the exhaustive one, that is, to conduct research of promising areas for the presence of an optimal or close to optimal solution. This algorithm allows solving problems of greater power of the set than a recursive-lexicographic algorithm. Using the above formula to count the number of partitions, we can split the solution domain into the required number of subdomains and use the proposed algorithm for each of them in parallel, thus implementing a parallel computing process.

Sets; kombinatorisches Set; lexicographic sequence; optimization.

Введение. Многие практические задачи формулируются в терминах комбинаторной оптимизации [1, 2]. К такому классу задач относятся все задачи автоматизированного конструкторского проектирования электронно-вычислительной аппаратуры (ЭВА) [3–5], в которых число элементов исчисляется миллионами и даже миллиардами [6, 7]. Найти оптимальное решение любой комбинаторной задачи, то есть получить её точное решение теоретически можно полным перебором всех возможных вариантов. Но, к сожалению, в подавляющем большинстве случаев это невозможно, так как при числе элементов за пределами второго десятка число вариантов становится астрономически большим, перебрать которые за приемлемое время невозможно даже на самом мощном компьютере. Такой подход в [8] автор называет «грубой силой и невежеством» – ГСН. Именно поэтому при автоматизированном проектировании ЭВА используются специфичные эвристические алгоритмы, которые позволяют за приемлемое время получать квазиоптимальные решения.

Однако это не означает, что полный перебор нужно исключить навсегда из подходов, применяемых для оптимизации. Существуют такие задачи, множество комбинаторных элементов в которых составляет всего два-три десятка. Одной из таких задач является задача формирования оптимальных турнирных таблиц. Задача о рюкзаке, ставшая в некотором роде тестовой, имеет такую же размерность. Но даже и для этих задач разрабатываются эвристические алгоритмы, поскольку полный перебор еще возможен при числе перебираемых элементов, не превышающих 20.

И все же полный перебор может оказаться полезным даже в тех случаях, когда комбинаторное множество достаточно велико. В таких случаях может использоваться случайный направленный поиск, предложенный и разработанный Растригиным Л.А [9]. В обзоре [10] приведена классификация, так называемых метаэвристических алгоритмов. Одна из таких эвристик, особо выделяемая автором, состоит в эффективном исследовании пространства поиска с целью его сужения. В суженном пространстве уже можно использовать полный перебор для нахождения решения, более близкого к оптимальному. Другой случай возможного использования полного перебора связан с организацией параллельных вычислений [11, 12] на комбинаторных множествах. В этом случае так же, как и при случайном поиске, необходимо уметь формировать комбинаторные соединения элементов по их номерам на множестве всех комбинаций. Разбив область поиска решения на некоторое количество равных подобластей, обработкой каждой из которых будет заниматься отдельное вычислительное устройство, можно получить выигрыш по времени вычислений. Однако параллельные вычисления как по экономическим (нужно использовать много вычислительных устройств), так и по программно-техническим причинам наталкиваются на ограничения, обусловленные законом Амдала [13–15].

Наиболее полное построение алгоритмов формирования различных комбинаторных соединений приведено в работах [16, 17], а статьи [18–20] посвящены исключительно вопросу генерации некоторых комбинаторных соединений по их номерам. Однако существуют такие комбинаторные соединения, формирование которых по номеру не нашли должного отражения в научной и практической литературе. Такими комбинаторными соединениями являются (r,n)-разбиения. В предлагаемой статье приводятся результаты разработки такого алгоритма, который ориентирован для формирования турнирных таблиц с двумя критериями оптимизации.

1. Построение формулы для подсчета числа разбиений. Сначала покажем, что множество всех бесповторных (r, n)-разбиений определяется ниже приведенной формулой

$$Q(n,r)=\prod_{i=1}^{r-1}(C_{k(i+1)-1}^{k-1})_i,$$

где n – общее число элементов в исходном множестве, r – число выборок в разбиении, k – число элементов в каждой выборке.

Рассмотрим работу формулы на примере 9,3-разбиения. Данное разбиение имеет вид: (1,2,3)(4,5,6)(7,8,9). Процесс формирования разбиений начинается с предпоследней выборки, так как перестановка элементов в последней выборке между собой не дает новое разбиение. Первый элемент в выборке не увеличивается, когда все предшествующие элементы достигли своих максимальных значений, а увеличивается на 1 последний элемент в предшествующей выборке. Таким образом, на данном этапе в перестановке задействованы 5 элементов: 5,6,7,8,9. Цифра 4 будет оставаться на своем месте до тех пор, пока не будут получены все комбинации по два элемента из пяти возможных. Количество вариантов такого перебора определяется числом сочетаний из 5 по 2, т.е. C_5^2. После перебора всех указанных сочетаний, в рассмотрение будут добавлены два элемента из первой выборки. Теперь рекурсивно-лексикографический перебор будет уже осуществляться на множестве 8 элементов, аналогично предыдущему случаю первый элемент не учитывается. Количество вариантов этого перебора дает число сочетаний из 8 по, 2, т.е. C_8^2. Но при каждом варианте из числа C_8^2 будут осуществляться все переборы во второй выборке, число которых есть величина C_5^2. С учетом комбинаторного принципа умножения общее число (9,3)-разбиений составит величину $C_8^2 \cdot C_5^2$. Распространяя данный подход на общий случай, мы получим формулу, приведенную выше.

Таким образом, данная формула позволяет вычислить полное количество всех не повторяющихся разбиений. Для случая (9,3)-разбиений их будет 280. С её помощью можно также рассчитать количество разбиений, которое будет сделано между сменой элемента в заданной выборке, например, между сменой элементов в первой выборке. Для этого необходимо представить, что мы имеем на одну выборку меньше в нашем разбиении и вычислить количество всех (6,2)-разбиений. В результате получим, что между разбиением (1,2,3)(4,5,6)(7,8,9) и (1,2,4)(3,5,6)(7,8,9) существует еще 9 разбиений с первой выборкой (1,2,3).

2. Алгоритм построения разбиения по его номеру. Количество всех бесповторных разбиений экспоненциально увеличивается с ростом числа элементов в разбиении, и может достигать астрономических значений даже при относительно малом количестве элементов (менее 30). Рекурсивно-лексикографический алгоритм позволяет последовательно получить все возможные разбиения, но это не всегда необходимо, а в некоторых случаях заранее можно предсказать, что необходимое разбиение находится между определенными номерами решений. Чтобы сократить объем вычислений в подобных случаях был разработан алгоритм, позволяющий построить разбиение согласно заданному номеру.

На рис. 1 представлена блок-схема алгоритма. На начальном этапе задаются: число элементов разбиения, число выборок в разбиении и номер необходимого разбиения. Вводится параметр j, отвечающий за номер выборки рассматриваемой на данном этапе. Затем, используя заданные параметры, строится первое разбиение в соответствии с лексикографическом порядком. Далее вводится параметр X, который показывает количество последовательных разбиений от текущего разбиения до необходимого.

Теперь необходимо вычислить, сколько смен элементов следует совершить в первой выборке. Для этого сначала вычислим количество разбиений, которое будет получено при неизменной первой выборке. Иначе говоря, воспользуемся формулой для подсчета всех разбиений, то есть в соответствии с формулой уменьшим количество выборок на единицу и вычтем из количества всех элементов количество элементов в одной выборке. Подставим полученные значения в формулу подсчета количества разбиений и запишем результат в переменную K. Тогда целая часть от деления X_0 на K будет количеством смен элементов в первой выборке, а остаток от деления записывается в X_1. Аналогичным образом подсчитываются количество смен элементов для оставшихся выборок до тех пор, пока остаток от деления станет равным нулю или количество не распределенных элементов будет умещаться в одну выборку.

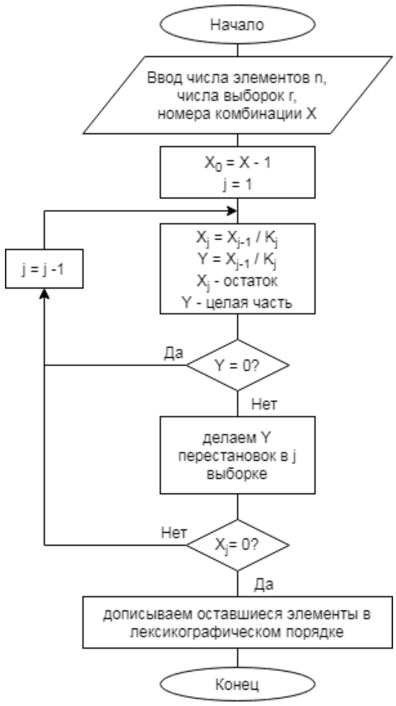

Рис. 1. Блок-схема алгоритма построения разбиения по номеру

Рассмотрим работу алгоритма на реальном примере. Пусть имеется 9 элементов, которые распределяются на 3 выборки и необходимо получить (9,3)-разбиение под номером 75, т.е. *n=9, r=3, X=75*. Тогда (1,2,3)(4,5,6)(7,8,9) будет первым разбиением и остается пропустить $X_0 = 74$ разбиений. Вычислим количество разбиений между сменой элемента в первой выборке, получим $K_1 = 10$, тогда $X_1 = 4$ и $Y = 7$. Следовательно, нужно сделать 7 последовательных смен элементов в первой выборке, получим новое разбиение (1,3,4)(2,5,6)(7,8,9). Таким образом, мы получили окончательную первую выборку (1,3,4). Теперь проведем аналогичные действия для вычисления второй выборки. Количество разбиений между сменой элемента во второй выборке $K_2 = 1$, тогда $X_2 = 0$ и $Y = 4$. Теперь сделаем 4 последовательные смены элементов во второй выборке, тогда искомое (9,3)-разбиение под номером 75 примет следующий вид (1,3,4)(2,6,7)(5,8,9).

3. Лексикографический алгоритм формирования разбиений. Алгоритм на рис. 2 является модификацией рекурсивно-лексикографического алгоритма, представленного в статье [21]. Исходный алгоритм позволяет получить все возможные бесповторные выборки для заданного числа элементов и выборок. Стоит отметить, что в некоторых ситуациях нет необходимости в получении всех возможных вариантов, а нужно получить определенное количество разбиений последовательно следующих друг за другом. Главным отличием данного алгоритма является возможность задания начального разбиения и числа последовательных разбиений, которые необходимо получить. То есть он позволяет выбрать область поиска решений, тем самым сокращая временные затраты на поиск решения, близкого к оптимальному.

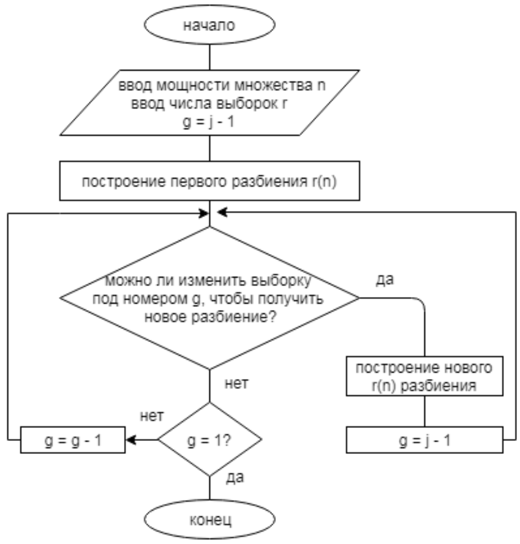

Рис. 2. Блок-схема лексикографического алгоритма

Заключение. Представленные в данной работе алгоритмы расширяют возможности при поиске оптимального решения на комбинаторных множествах. Сочетание рассмотренных алгоритмов позволяет проводить предварительную оценку множества всех решений, например, с помощью анализа разбиений, получаемых при смене элементов в первой или второй выборке. Таким образом, появляется возможность сузить область поиска, для которой можно получить набор разбиений. Это очень важно при решении задач с большим количеством элементов, так как количество разбиений увеличивается экспоненциально в зависимости от числа всех элементов.

В случаях, когда не удается выделить отдельные области для поиска или в силу других причин и необходимо получить все решения комбинаторной задачи оптимизации, можно воспользоваться распределенными вычислительными устройствами. При помощи алгоритма построения разбиения по номеру, мы можем разбить задачу получения всех решений на меньшие задачи и вычислять решения параллельно. Это позволит получить большой выигрыш во времени вычислений для задач большой размерности.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Сергиенко И.В., Шило В.П.* Задачи дискретной оптимизации: проблемы, методы решения, исследования. – Киев: Наук. Думка, 2003. – 264 с.
2. *Семенова Н.В, Колечкина Л.М.* Векторные задачи дискретной оптимизации на комбинаторных множествах: методы исследования и решения: монография. – Киев: Наукова думка, 2009. – 266 с.
3. *Гладков Л.А., Гладкова Н.В., Бричеева Ю.В.* Решение задачи компоновки схем ЭВА на основе выделения клик // Труды конгресса по интеллектуальным системам и информационным технологиям «IS– IT’13». Научное издание в 4-х т. Т. 3. – М.: Физматлит, 2013. – С. 151-153.
4. *Гладков Л.А., Шкурко В.А.* Решение задачи разбиения графа на подграфы на основе генетического алгоритма // Труды конгресса по интеллектуальным системам и информационным технологиям «IS – IT’14». Научное издание в 4-х т. Т. 3. – М.: Физматлит, 2014. – С. 142-146.

5. *Курейчик В.М., Кажаров А.А.* Роевой интеллект в решении графовых задач // Труды Международной конференции по мягким вычислениям и измерениям (SCM'2013). – СПб.: 2013. – С. 31-34.
6. *Глушань В. М., Лаврик П. В.* Распределенные САПР. Архитектура и возможности: монография. – Старый Оскол: ТНТ, 2014. – 188 с.
7. Тест и обзор: Intel Core i7-4770K и Core i5-4670K – новое поколение Haswell. – М., 2013. – Режим достпа: http://www.hardwareluxx.ru/index.php/artikel/hardware/prozessoren/26018-haswell-test-intel-core-i7-4770k-core-i5.html, свободный.
8. *Горбатов В.А* Основы дискретной математики: учеб. пособие для студентов вузов. – М.: Высш. шк., 1986. – 311 с.
9. *Растригин Л.А.* Теория и применение случайного поиска. – Рига: Зинатне, 1969. – 309 с.
10. *Щербина О.А.* Метаэвристические алгоритмы для задач комбинаторной оптимизации (обзор) // Таврический вестник информатики и математики. – 2014. – № 1 (24). – С.56-71.
11. *Воеводин В.В., Воеводин Вл. В.* Параллельные вычисления. – СПб.: БХВ-Петербург, 2004. – 608 с.
12. *Воеводин Вл.В.* Распределенная обработка данных // Вторая Сибирская школа-семинар по параллельным вычислениям / под ред. проф. А.В. Старченко. – Томск: Изд-во Том. ун-та, 2004. – 108 с.
13. Закон Амдала. – URL:https://dic.academic.ru/dic.nsf/ruwiki/433588.
14. Закон Амдала и будущее многоядерных процессоров. – URL:https://www.osp.ru/os/2009/04/9288815.
15. *Глушань В.М., Лаврик П.В.* Ограничение быстродействия вычислительных систем в результате совмещения закона Амдала и гипотезы Минского // Труды Конгресса по интеллектуальным системам и информационным технологиям «IS-IT'13». Научное издание в 4-х т. Т. 1. – М.: Физматлит, 2013. – С. 129-137.
16. *Липский В.* Комбинаторика для программистов: пер. с польск. – М.: Мир, 1988. – 213 с.
17. *Курейчик В.М, Глушань В.М., Щербаков Л.И.* Комбинаторные аппаратные модели и алгоритмы в САПР. – М.: Радио и связь, 1990. – 216 с.
18. *Герасимов В.А.* Генерация случайных сочетаний. Генерация сочетаний по его порядковому номеру // RSDN Magasine. – 2010. – № 3.
19. *Дубинин И.С., Арапов С.Ю., Тягунов А.Г.* Рациональный метод генерации сочетаний для параллельных вычислений в некоторых комбинаторных задачах // Международная конференция студентов, аспирантов и молодых ученых "Информационные технологии, телекоммуникации и системы управления": Сб. докладов. – Екатеринбург: УрФУ, 2015. – С. 174-178.
20. *Тимошевская Н.Е.* О нумерации перестановок и сочетаний для организации параллельных вычислений в задачах проектирования вычислительных систем // Известия Томского политехнического университета. – 2004. – Т. 307, № 6. – С. 18-20.
21. *Глушань В.М., Зубрицкий А.В.* Теоретическое обоснование алгоритма формирования упорядоченных разбиений с равномощными бесповторными выборками // Труды 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.

REFERENCES

1. *Sergienko I.V., Shilo V.P.* Zadachi diskretnoy optimizatsii: problemy, metody resheniya, issledovaniya [Problems of discrete optimization: problems, methods of solution, research]. Kiev: Scientific thought, 2003, 264 p.
2. *Semenova N.V., Kolechkina L.M.* Vektornye zadachi diskretnoy optimizatsii na kombinatornykh mnozhestvakh: metody issledovaniya i resheniya [Vector problem of discrete optimization on combinatorial sets: research methods and solutions], *Monografiya* [Monograph]. Kiev: Scientific thought, 2009, 266 p.
3. *Gladkov L.A., Gladkova N.V., Bricheeva U.V.* Reshenie zadach kompanovki skhem EVA na osnove vydeleniya klik [Solving the problem of layout of eva based on the selection of cliques], *Trudy kogressa po intellektual'nym sistemam i informatsionnym tekhnologiyam «IS&IT'13». Nauchnoe izdanie v 4 t. T. 3.* [Proceedings of the Congress on Intelligent Systems and Information Technologies «IS&IT'13». Scientific publication in 4 vol. Vol. 3]. Moscow: Fizmatlit, 2013, pp. 151-153.

4. *Gladkov L.A., Shkurko V.A.* Reshenie zadach razbieniya grafa na podgrafy na osnove geneticheskogo algoritma [Solution of the problem of partitioning a graph into subgraphs based on a genetic algorithm], *Trudy kogressa po intellektual'nym sistemam i informatsionnym tekhnologiyam «IS&IT'14». Nauchnoe izdanie v 4 t. T. 3* [Proceedings of the Congress on Intelligent Systems and Information Technologies «IS&IT'14». Scientific publication in 4 vol. Vol. 3]. Moscow: Fizmatlit, 2014, pp. 142-146.
5. *Kureychik V.M., Kazharov A.A.* Roevoy intellect v reshenii grafovykh zadach [Root intelligence in solving graph tasks], *Trudy mezhdunarodnoy konferentsii po myagkim vichisleniyam i izmereniyam (SCM'2013)* [Proceedings of the International Conference on Soft Computing and Measurement (SCM'2013)]. Saint Petersburg, 2013, pp. 31-34.
6. *Glushan V.M., Lavrik P.V.* Raspredelennie SAPR. Arkhitektura i vozmozhnosti [Distributed CAD. Architecture and capabilities], *Monografiya* [Monograph]. Stariy Oskol: TNT, 2014, 188 p.
7. Test i obzor: Intel Core i7-4770K i Core i5-4670K – novoe pokolenie Haswell [Test and review: Intel Core i7-4770K and Core i5-4670K – new generation]. M., 2013. Available at: http://www.hardwareluxx.ru/index.php/artikel/hardware/prozessoren/26018-haswell-test-intel-core-i7-4770k-core-i5.html.
8. *Gorbatov V.A* Osnovy diskretnoy matematiki: *ucheb. posobie dlya studentov vuzov* [Basics of Discrete Mathematics: Textbook for university students]. Moscow: High School, 1986, 311 p.
9. *Rastrigin L.A. Teoriya i primenenie sluchaynogo poiska* [Theory and application of random search], Riga: Zinatne, 1969, 309 p.
10. *Shcherbina O.A.* Metaevristicheskie algoritmy dlya zadach kombinatornoy optimizatsii (obzor) [Metaheuristic algorithms for combinatorial optimization problems (review)], *Tavricheskiy vestnik informatiki i matematiki* [Tavrichesky messenger of computer science and mathematics], 2014, No. 1 (24), pp. 56-71.
11. *Voevodin V.V., Voevodin Vl.V.* Parallelnye vychisleniya [Parallel computing]. Saint Petersburg: BHV-Peterburg, 2004, 608 p.
12. *Voevodin Vl.V.* Raspredelennaya obrabotka dannykh [Distributed data processing], *Vtoraya Sibirskaya shkola seminar po parallel'nym vychisleniyam* [The Second Siberian School-Seminar on Parallel Computing], ed. by prof. A.V. Starchenko. Tomsk: Tomsk university, 2004, 108 p.
13. Zakon Amdala [Amdahl's law]. Available at: https://dic.academic.ru/dic.nsf/ruwiki/433588.
14. Zakon Amdala i budushchee mnogoyadernih processorov [Amdahl's Law and the Future of Multi-Core Processors]. Available at: https://www.osp.ru/os/2009/04/9288815.
15. *Glushan V.M., Lavrik P.V.* Ogranichenie bystodeystviya vychislitel'nykh system v rezultate sovmeshcheniya zakona Amdala i gipotezi Minskogo [The limitation of the speed of computing systems as a result of the combination of the Amdahl law and the Minskiy hypothesis], *Trudy kogressa po intellektual'nym sistemam i informatsionnym tekhnologiyam «IS&IT'13». Nauchnoe izdanie v 4 t. T. 1.* [Proceedings of the Congress on Intelligent Systems and Information Technologies «IS&IT'13». Scientific publication in 4 vol. Vol. 1]. Moscow: Fizmatlit, 2013, pp. 129-137.
16. *Lipskiy V.* Kombinatorika dlya programistov [Combinatorics for programmers]. Moscow: Mir, 1988, 213 p.
17. *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.
18. *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 Magasine], 2010, Vol. 3.
19. *Dubinin I.S., Arapov S.U., Tyagunov A.G.* Ratsionalnyy metod generatsii sochetaniy dlya parallel'nykh vychisleniy v nekotorykh kombinatornykh zadachah [Rational method of generating combinations for parallel computations in some combinatorial problems], S*bornik dokladov Mezhdunarodnoy konferentsii studentov, aspirantov i molodykh uchenykh "Informatsionnye tehnologii, telekommunikatsii i sistemy upravleniya"* [Collection of reports of the International conference of students, graduate students and young scientists "Information technology, telecommunications and management systems"]. Ekaterinburg: UrFU, 2015, pp. 174-178.

20. *Timoshevskaya N.E.* O numeratsii perestanovok i sochetaniy dlya organizatsii parallel'nykh vichisleniy v zadachakh proektirovaniya vicheslitel'nykh system [On the numbering of permutations and combinations for the organization of parallel computations in problems of designing computer systems] *Izvestiya Tomskogo politekhnicheskogo uniersiteta* [Bulletin of the Tomsk polytechnic university], 2004, Vol. 307, No. 6, pp. 18-20.
21. *Glushan V.M., Zubrittskii A.V.* Teoreticheskoe obosnovanie algoritma formirovaniya uporyadochennykh razbieniy s ravnomoshchnymi bespovtornimi viborkami [Theoretical substantiation of the algorithm of formation of ordered divisions with equipped unbeatable selections], *Trudi kogressa po intellekrual'nykh sistemam i informatsionnym tekhnologiyam «IS&IT'17». Nauchnoe izdanie v 3 t. T. 2* [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.

Статью рекомендовал к опубликованию д.т.н., профессор В.П. Карелин.

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.

УДК 004.896 DOI 10.23683/2311-3103-2018-4-66-74

В.М. Курейчик, И.Б. Сафроненкова

МЕТОД ФОРМИРОВАНИЯ КЛАСТЕРОВ ЗАДАЧ КОМПОНОВКИ*

Стремительное развитие современных средств вычислительной техники обуславливает необходимость в разработке новых методов интеллектуальной обработки информации. В статье показана актуальность автоматической разработки онтологии предметной области, которая представляет собой «каркас» базы знаний в интеллектуальной системе поддержки принятия решений (ИСППР). Проблема сравнительного анализа инженерного программного обеспечения с целью подбора под тип задачи и вычислительные ресурсы является актуальным направлением в исследованиях отечественных и зарубежных ученных. Решение данной задачи во многих случаях осуществляется с использованием ИСППР. Ввиду большого разнообразия формулировок задач компоновки, кластеризация – необходимый этап автоматической разработки онтологии задач компоновки. Возникает проблема автоматической кластеризации задач компоновки в связи с необходимостью комплексного сравнения данных, представленных в виде структур разной размерности. Целью настоящей работы является разработка метода формирования кластеров задач компоновки конструктивных узлов на основе унификации матриц инцидентности. В качестве представления схемы выбрана гиперграфовая модель, формализована постановка задачи компоновки конструктивных узлов, рассмотрен случай кластеризации матриц инцидентности различной размерности. Новизна предложенного метода формирования кластеров задач компоновки конструктивных узлов заключается во введении процедуры унификации матриц различной размерности в типовой алгоритм кластеризации. Проведены эксперименты по формированию кластеров задач компоновки конструктивных узлов из представленной выборки. На настоящем этапе исследования можно сделать вывод о невысокой вычислительной сложности производимых расчетов и возможности эффективной обработки информации на ЭВМ. Принципиальным отличием разработанного метода от типовых методов кластерного анализа является возможность кластеризации задач компоновки

* Работа выполнена при финансовой поддержке РФФИ (проекты: № 18-07-00050, № 18-29-22019).