# РАЦИОНАЛЬНЫЙ МЕТОД ГЕНЕРАЦИИ СОЧЕТАНИЙ ДЛЯ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ В НЕКОТОРЫХ КОМБИНАТОРНЫХ ЗАДАЧАХ

**Дубинин И. С., Арапов С. Ю., Тягунов А. Г.**

*УрФУ имени первого Президента России Б.Н.Ельцина*

**Целью данной статьи являлась разработка нового способа нумерации сочетаний. Его особенностью является отсутствие в алгоритме циклов и ветвлений, что позволяет эффективно использовать его в многопоточном режиме. Для выполнения поставленной задачи были произведены разработка алгоритма, нахождение оптимального способа вычисления требуемых величин и оптимизация под многопоточные системы. По сравнению с классическими, данный метод показывает заметное увеличение производительности даже не в самых благоприятных условиях. Особенностью полученного алгоритма является независимое выполнение разных потоков вычисления, что очень важно при выполнении программы на процессорах *SIMD* архитектуры. Таким образом, разработанный способ нумерации имеет очевидные преимущества и может быть использован в задачах, решаемых методами комбинаторной оптимизации.**

Ключевые слова: сочетания, комбинаторная оптимизация, параллельные вычисления.

# RATIONAL METHOD OF GENERATION OF COMBINATIONS FOR PARALLEL CALCULATIONS IN SOME COMBINATORIAL PROBLEMS

**Dubinin I.S., Arapov S.Y., Tjagunov A.G.**

*Ural Federal University named after the first President of Russia B.N.Yeltsin*

**The purpose of this paper is development of a new method of numbering combinations. Its feature is the absence of loops and branches in the algorithm, which allows using it effectively multithreaded. For this purpose the following tasks have been produced: development of an algorithm, finding an optimal method for calculating the required values and optimization for multi-threaded system. Compared to classical ones, proposed method showed a significant performance increase even in adverse conditions. Thus, the developed algorithm of numbering has obvious advantages, and can be used in tasks, solved by methods of combinatorial optimization.**

Keywords: algorithm, combinatorial optimization, parallel computing.

## Введение

Разрабатывалась система автоматизированного составления производственного расписания [2,3]. Для этого использовался метод локального поиска. Выбор метода обусловлен его высокой эффективностью для данного класса задач, что было показано в классической работе [6]. Для повышения практической ценности программы было решено использовать в качестве аппаратной базы более распространенные и доступные решения для параллельных вычислений – видеокарты (GPU). Особенностью процессоров графических карт является их многоядерная структура. Разместить в одном кристалле до нескольких тысяч ядер позволило некоторое их упрощение. Видеопроцессор способен показывать значительную производительность на арифметических действиях, которые выполняются параллельно и независимо, но более сложные операции, например, операции ветвления, сводят преимущество на нет.

Следовательно, для эффективного использования видеопроцессора требуется метод получения перестановок и сочетаний, максимально свободный от указанных ограничений.

**Цель исследования**

В данной работе стояла задача разработать метод генерации сочетаний, использующий только алгебраический способ получения номеров объектов из индекса нумерации.

**Решение**

На рис. 1 приведено графическое представление сочетаний по 2 из 5 и 6. Строки соответствуют первому элементу сочетания, столбцы – второму. Так как выбор элемента для сочетания возможен только один раз, столбцы соответствуют элементам со сдвигом в единицу. Например, индекс 8 на рисунке 1а находится в третьей строке и третьем столбце. Это расшифровывается следующим образом: данному объекту соответствует выбор третьего (третья строка) и четвертого (третий столбец) элементов исходного массива.

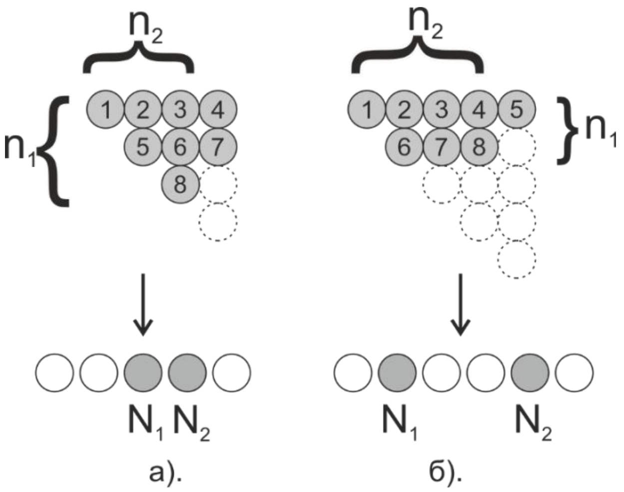


**Рис. 1. Графическое представление выборок: а) по 2 из 5, б) по 2 из 6.**

Проблема традиционного подхода нумерации объектов заключается в том, что для определения номеров элементов необходимо знать общее их количество. На рисунке 1 видно, что сочетания под одинаковыми индексами из разного количества объектов подразумевают очевидно разные объекты.

Чтобы избежать данной особенности, предлагается использовать нумерацию «с другой стороны». На рисунке 2 изображено графическое представление предлагаемого порядка перечисления сочетаний из конечного набора элементов по три. Предложенная методика хороша тем, что при любом количестве элементов в исходном наборе, индекс нумерации всегда будет строго соответствовать сочетанию.

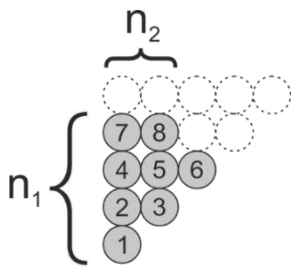


**Рис. 2. Сочетания по 2 из 6 с использованием предложенного алфавитно-графического порядка.**

Для лучшего понимания принципа работы можно провести геометрическую аналогию. Так, каждому сочетанию будет соответствовать шарик, а множество всех возможных сочетаний можно представить в виде n-мерной пирамиды основанием вверх (Рис. 3). По мере нарастания индекса нумерации пирамида будет заполняться шариками начиная с вершины. В этом случае, для нахождения сочетания, соответствующей индексу интересующего нас шарика, нужно будет последовательно вычислять её координаты. Начало координат находится в вершине пирамиды.

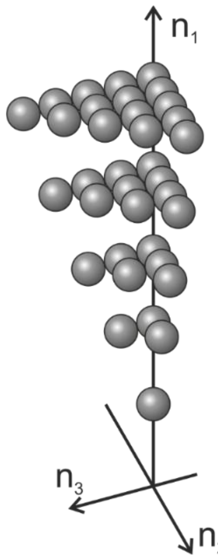


**Рис. 3 Графическое представление сочетаний трех объектов. Трем координатам каждого шарика соответствуют три заказа.**

Таким образом, индекс сочетания по три определяется формулой:

$$I = \frac{n_1(n_1+1)(n_1+2)}{6} + \frac{n_2(n_2+1)}{2} + n_3, \quad (1)$$

где

$$N_1 = n_1 + 2, \qquad N_2 = n_2 + 1, \qquad N_3 = n_3 \quad (2)$$

Решать уравнение (1) следует в следующем порядке. Сначала необходимо решить уравнение:

$$x_1(x_1+1)(x_1+2) = 6I, \quad (3)$$

и округлить полученный результат в меньшую сторону до ближайшего целого. Полученное значение и будет являться основой $n_1$ для получения индекса сочетания $N_1$ с помощью выражений (2):

$$n_1 = fix(x_1), \text{или } n_1 = floor(x_1). \tag{4}$$

Далее необходимо найти разность:

$$I_1 = I - n_1, \tag{5}$$

и решить следующее уравнение:

$$x_2(x_2 + 1) = 2I_1, \tag{6}$$

откуда аналогичным образом получаем $n_2$ :

$$n_2 = fix(x_2), \text{или } n_2 = floor(x_2). \tag{7}$$

Разность

$$n_3 = I_1 - n_2, \tag{8}$$

даст последнее недостающее значение $n_3$ . После этого с помощью выражений (2) могут быть получены выборочные индексы.

Решение уравнения (6) не представляет никакой сложности:

$$x_2 = \frac{\sqrt{I_1+1}-1}{2}, \tag{9}$$

второй корень квадратного уравнения (6) необходимо исключить.

Решение кубического уравнения (3) несколько сложнее предыдущего:

$$x_1 = \sqrt[3]{3I + \sqrt[2]{9I^2 - 1/27}} + \frac{1}{3\sqrt[3]{3I+\sqrt[2]{9I^2-1/27}}} - 1, \tag{10}$$

это единственный положительный корень, остальные являются комплексными числами и могут быть исключены из дальнейшего рассмотрения.

**Результат исследования**

Проверка эффективности описанного алгоритма проводилась путем его применения к программе составления производственного расписания, написанной на языке *MATLAB*. Для сравнения использовалась программа, использующая принцип работы, описанный в [1,4,5].

Исследование производительности осуществлялось на компьютере с двумя процессорами *Intel Xeon X*5670 (12 вычислительных потоков). Тестирование производилось на одинаковом наборе исходных данных. Было осуществлено по 100 запусков каждой из программ, для сравнения было взято среднее арифметическое времени выполнения.

Программе с описанным алгоритмом для нахождения решения требуется 9,6 сек, тогда как программа, использующая классический метод нумерации выполняется за 13 сек. Таким образом, прирост производительности составил примерно 35%.

**Выводы**

В ходе работы был получен метод генерации сочетаний, превосходящий по быстродействию используемый ранее. Тестирование алгоритма в версии программы, выполняемой на *CPU* показало его эффективность. При реализации метода с использованием *SIMD* архитектуры графического ускорителя можно ожидать улучшения результата, пропорционального росту количества вычислительных ядер системы.

**Список литературы**

1. Герасимов В.А. Генерация случайных сочетаний. Генерация сочетания по его порядковому номеру // RSDN Magazine. — 2010. — № 3.

2. Некоторые допустимые приближения в задаче составления оптимальных производственных расписаний для полиграфических предприятий / Арапов С.Ю. [и др.] // Известия высших учебных заведений. Проблемы полиграфии и издательского дела. – 2010. – № 4. – С. 89–95.

3. Применение параллельных вычислений к решению задачи оптимизации производственных расписаний для полиграфического предприятии / Арапов С.Ю. [и др.] // Известия высших учебных заведений. Проблемы полиграфии и издательского дела. – 2010. – № 5. – С. 82–90.

4. Тимошевская Н.Е. О нумерации перестановок и сочетаний для организации параллельных вычислений в задачах проектирования управляющих систем // Известия Томского политехнического университета. – 2004. – Том 307. – № 6. – С. 18 – 20.

5. Тимошевская Н.Е. Разработка и исследование параллельных комбинаторных алгоритмов // Прикладная дискретная математика –2009. – № 2(4). С. 96–103.

6. An Effective Heuristic Algorithm for the Traveling-Salesman Problem / S Lin., B.W. Kernighan // Operations Research. – 1973. – № 2. – Vol. 21. – P. 498–516.