

## ПРИМЕНЕНИЕ ФАКТОРИАЛЬНОЙ СИСТЕМЫ ДЛЯ РЕШЕНИЯ КОМБИНАТОРНЫХ ЗАДАЧ

В. И. ПОЛЯКОВ<sup>1</sup>, В. И. СКОРУБСКИЙ<sup>1</sup>, Ю. В. ЭКАЛО<sup>2</sup>

<sup>1</sup>Университет ИТМО, 197101, Санкт-Петербург, Россия

E-mail: v.i.polyakov@mail.ru

<sup>2</sup>ОАО „НИЦ ЭТУ“, 197376, Санкт-Петербург, Россия

Рассматриваются свойства факториальной системы счисления и некоторые новые сферы ее применения для решения ряда комбинаторных задач в программировании и проектировании компьютерных систем. Факториальные числа могут быть однозначно пронумерованы десятичными или двоичными числами и интерпретированы перестановками элементов любого типа. Декодирование двоично-кодируемой записи перестановки в факториальной системе счисления позволяет каждой перестановке однозначно поставить в соответствие размещение дискретных объектов (элементов, ссылок и др.) в некотором фиксированном дискретном пространстве. Перестановки и размещения могут быть применены в задачах автоматизации конструкторского проектирования, коммутации каналов вычислительных систем и в защите данных.

**Ключевые слова:** перестановки, факториальная система счисления, двоичное кодирование факториальных чисел, преобразование двоичного числа в факториальное и обратно, размещения, сочетания элементов.

**Введение.** Среди множества систем счисления в вычислительной технике наиболее используемыми остаются десятичная и двоичная. От большинства других систем счисления [1] пришлось отказаться, так как присущие им недостатки превалировали над достоинствами. Однако не исключено, что с развитием информационных технологий и техники интерес к использованию нетрадиционных систем счисления будет расти. Ряд задач, связанных с размещением элементов конечного множества в конструктиве или с коммутацией каналов и устройств в вычислительных устройствах, носят комбинаторный характер. Факториальная система кодирования чисел [2] позволяет интерпретировать перестановки применительно к различным конечным множествам элементов. В настоящей работе рассматриваются свойства этой системы и приводится пример использования в ряде практических задач, решаемых средствами как программирования, так и схемотехники.

**Факториальная система счисления.** Позиционная система счисления обеспечивает запись натурального числа  $N$  в виде цифрового слова  $F = f_n f_{n-1} \dots f_1$ , состоящего из упорядоченного набора символов  $f_i$ , взятых из множества основных символов  $A$  (алфавита системы счисления). Число 0 записывается в виде цифрового слова 0. В факториальной записи числа каждый символ  $f_i$  записи цифры  $i$ -го разряда числа  $F$  выбирается из множества  $A = \{0, 1, \dots, i - 1\}$ , каждому натуральному числу  $N$  однозначно соответствует запись в факториальной системе

$$N = \sum_{i=1}^n f_i p(i) \quad (1)$$

( $p(i)$  — вес  $i$ -го разряда) и существует рекурсивный алгоритм преобразования

$$\{p(i+1) = \Phi(p(i)), f_i = \Psi(p(i)).$$

В факториальной системе счисления  $p(i+1) = i!$ ,  $0 \leq f_i \leq (i - 1)$ , тогда

$$N = f_n n! + f_{n-1} (n-1)! + \dots + f_1 1! = ((\dots((0 + f_n)n + f_{n-1})(n-1) + \dots + f_2)2 + f_1)1 \quad (2)$$

или по рекурсивной формуле

$$\{S_{i+1} = S_i(n-i) + f_{n-i}, S_0 = 0, i = 1, 2, \dots, n\}.$$

Таким образом, с целью получения последовательности цифр  $f_1, f_2, \dots, f_n$  факториального числа для исходных десятичного или двоичного чисел находим {целая часть  $S_1 = S_0/2 = N/2$  и остаток целочисленного деления  $f_1 = S_0 \% 2 = N \% 2 = 0$  (% — операция получения остатка целочисленного деления)}

$$\{S_i = S_{i-1}/(i+1), f_i = S_{i-1} \% (i+1)\}, S_0 = N, i = 1, 2, \dots, n.$$

Например, если  $N = 100$ , то  $\{f_1 = 100 \% 2 = 0, S_2 = 100 / 2 = 50\}$ ,  $\{f_2 = 50 \% 3 = 2, S_3 = 16\}$ ,  $\{f_3 = 16 \% 4 = 0, S_4 = 4\}$ ,  $\{f_4 = 4 \% 5 = 4, S_5 = 0\}$ .

Следовательно, число 100 в факториальной системе счисления представляется в виде  $F = f_4 f_3 f_2 f_1 = 4020_f$ .

В программе преобразования на языке Си выбранный порядок нумерации для некоторых индексов меняется, так как он определяет порядок доступа к ячейкам памяти, для которых принят унифицированный порядок BigEndian — нулевой индекс массива  $F[0]$ , младший индекс и далее размещаются, последовательно доступны элементы  $F[1], F[2], \dots$  с адресами +1, +2, +3, ... с индексами 1, 2, 3, ...

Приведем фрагмент программы преобразования двоичного числа в факториальное на языке программирования Си.

```
char M=5; //количество разрядов факториального числа
char F[4], N=100; j=3; // j=3,2,1,0 — номер позиции элемента в F[j]
DtoF() //преобразование двоичного числа в факториальное
{char i, j=4;
  for(i=1; i<=M; i++)
  {F[i -1]=N%i; N=N/i; j--;}
}
```

Обратное преобразование числа  $F$  из факториальной системы счисления в десятичную или двоичную форму записи выполняется пересчетом согласно формуле (2). Например, в десятичной системе

$$N = 4020_f = 4 \times 4! + 0 \times 3! + 2 \times 2! + 0 \times 1! = 100.$$

Таким образом, между целыми неотрицательными числами и их представлениями в факториальной системе счисления существует взаимно-однозначное соответствие.

Двоично-кодированная запись числа в факториальной системе счисления может быть использована в некоторых арифметических операциях с положительными числами (+1, -1, +, -).

В частности, результат поразрядного сложения  $\{S_i\}$  двух чисел в факториальной системе счисления с разрядами  $\{X_i\}$  и  $\{Y_i\}$  представляет собой сложение  $X_i + Y_i$  с переносом единицы в следующий разряд, т.е. если  $X_i + Y_i < i + 1$ , то  $S_i = (X_i + Y_i)$  и  $S_{i+1} = S_{i+1}$  и  $S_{i+1} = S_{i+1}$ , иначе  $S_i = (X_i + Y_i) - (i + 1)$  и  $S_{i+1} = S_{i+1}$  и  $S_{i+1} = S_{i+1} + 1$ .

Для двоичного кодирования цифр максимального пятиразрядного факториального числа  $F = X_5 X_4 X_3 X_2 X_1 = 54321_f = 120$  необходимо 11 бит информации (для кодирования 5 и 4 необходимо по три бита, для 2 и 3 — по два и для 1 — один). При доведении до максимального, кратного  $m = \lceil \log_2 n + 1 \rceil$  ( $n = 5, m = 3$ ), числа битов, используемых для кодирования каждой цифры записи 5-разрядного двоично-факториального числа, требуется  $n \times m = 15$  бит информации. Таким образом, оптимальным машинным представлением чисел является двоичное кодирование, поэтому в преобразованиях в факториальную систему счисления используются двоичные числа. Двоичный 8-разрядный код с максимальным значением 255 может определить любую из 5! перестановок, 16-битный код — любую из 8! перестановок.

**Алгоритм формирования перестановки.** В настоящей работе предлагается метод интерпретации факториального числа перестановками конечного множества элементов любого

типа, который ставит в соответствие перестановкам естественный порядок в десятичной системе. На основе этого метода могут быть предложены эквивалентные машинные программы или аппаратные решения практических задач из различных областей с перестановками.

Рассмотрим пример формирования перестановки элементов из пятиэлементного множества символов  $Q = \{q_4, q_3, q_2, q_1, q_0\} = \{a, b, c, d, e\}$ . Общее число перестановок равно  $5! = 120$ . Каждой перестановке соответствует десятичное число от 0 до 119 (номер перестановки), представляющее собой десятичную запись четырехразрядного числа в факториальной системе счисления. Из предыдущего примера выберем перестановку с номером  $N = 100$ .

Преобразуем десятичный номер перестановки в его запись в факториальной системе счисления  $F_f = f_3 f_2 f_1 f_0 = 4020_f$ . Определим перестановку  $R = \{r_4, r_3, r_2, r_1, r_0\}$ , соответствующую номеру  $N = 100$ . Для этого будем последовательно заполнять элементами  $a, b, c, d, e$  множества  $Q$  позиции  $r_4, r_3, r_2, r_1, r_0$  перестановки  $R$  слева направо на основе значений разрядов числа 100, записанного в факториальной системе счисления.

Для контроля выбранных и незанятых позиций перестановки  $R$  в процедуре заполнения используем двоичный вектор  $S = \{s_4, s_3, s_2, s_1, s_0\}$  с начальным значением  $S = \{0, 0, 0, 0, 0\}$ .

Поставим элемент  $\{a\}$  с индексом  $q_4$  в  $Q$  в позицию  $r_4$  в перестановке  $R$  (это соответствует значению старшего разряда  $f_4 = 4$  числа  $4020_f$ ) и получим  $R = \{a, r_3, r_2, r_1, r_0\}$ . Отметим в двоичном векторе  $S$  соответствующий бит  $s_4$ , который обозначает занятую позицию в  $R$ , значением 1. Получим двоичный код размещения  $S = \{1, 0, 0, 0, 0\}$ .

Далее рассмотрим четыре оставшихся свободных места  $\{r_3, r_2, r_1, r_0\}$  в перестановке  $R$ . Для назначения места в перестановке  $R$  элементу  $q_3 = \{b\}$  из множества  $Q$  выберем место  $r_0$ , соответствующее значению следующего разряда  $f_3 = 0$  числа  $4020_f$ . Получим  $R = \{a, r_3, r_2, r_1, b\}$  и двоичный код размещения  $S = \{1, 0, 0, 0, 1\}$ .

Далее для свободных мест  $\{r_3, r_2, r_1\}$  в  $R$  ставим в соответствие три свободные позиции  $\{s_3, s_2, s_1\}$  в  $S$ . Кодировем эти три бита разрядами двоичного кода  $(b_2 b_1 b_0)$ , элементу  $q_2 = \{c\}$  из множества  $Q$  в перестановке  $R$  назначается позиция  $r_3$ , соответствующая коду  $(b_2 b_1 b_0) = (100)$  для  $f_2 = 2$  числа  $4020_f$ . Получим  $\{s_3, s_2, s_1\} = (b_2 b_1 b_0) = (100)$ ,  $R = \{a, c, r_2, r_1, b\}$  и код размещения  $S = \{1, 1, 0, 0, 1\}$ .

Для элемента  $q_1 = \{d\}$  множества  $Q$  в перестановке  $R$  и в  $S$  остаются две свободные позиции  $\{r_2, r_1\}$  и  $\{s_2, s_1\}$ . Для  $f_1 = 0$  выбираем код  $\{s_2, s_1\} = (b_1 b_0) = (01)$ . Получим  $R = \{a, c, r_2, d, b\}$  и код размещения  $S = \{1, 1, 0, 1, 1\}$ .

Последнему элементу  $q_0 = \{e\}$  множества  $Q$  в перестановке  $R$  остается единственное место  $r_2$ . Получим  $R = \{a, c, e, d, b\}$ , код размещения определяет завершение формирования перестановки  $S = \{1, 1, 1, 1, 1\}$ .

Размещение элементов в  $R$  определяет индекс  $i = 4, 3, 2, 1, 0$ . Очевидно, можно принять инверсную нумерацию и размещение с индексами  $i = 0, 1, 2, 3, 4$ . Тогда индекс инвертируется вычитанием из четырех.

Прямая перестановка имеет вид

$$\begin{aligned} k &= 4\ 3\ 2\ 1\ 0 \sim a\ b\ c\ d\ e \\ i &= 4\ 0\ 3\ 1\ 2 \sim a\ c\ e\ d\ b, \end{aligned}$$

инверсная —  $i = 0\ 4\ 1\ 3\ 2 \sim b\ d\ e\ c\ a$ .

Приведем фрагмент программы формирования перестановки на языке Си:

```
char M=5; //количество элементов
char R[5]; //прямая перестановка M=5
char F[5]; //факториальное число 40200
char N=100; //двоичный номер перестановки
char Q[]="abcde"; //исходное размещение
unsigned char S=0;
DtoF() { } //преобразование двоичного числа в факториальное
main()
{char df, j, i=0, k=0, S1;
```

```

DtoF(); //факториальное число F
for(j=4;j>=0;){ //перестановка
  df= F[j]; //чтение j цифры в F
  do{
    if (S&1) {S=S>>1; i++;}; //просмотр i бит в S – пропуск единиц
    if(df) {df=df-1;i++;}; //расчет индекса размещения
  }
  while(df/(S&1));
  if(df==0) {R[i]=Q[k++]; S1=S=S1|(1<<i); }
}
}

```

Для используемой пары индексов  $(k, i)$   $k$  — адрес выбираемого элемента из  $Q$ , а  $i$  — адрес позиции этого элемента в перестановке  $R$ .

Приведем таблицу формирования прямой перестановки  $Q[k] \rightarrow R[i]$  по шагам.

$k$	$i$	$S_k(i)$	$Q[k]$	$j$	$F[j]$	$R_k(i)$
4	4	10000	$a$	3	4	$a0000$
3	0	10001	$b$	2	0	$a000b$
2	3	11001	$c$	1	2	$ac00b$
1	1	11011	$d$	0	0	$ac0db$
0	2	11111	$e$	—	—	$acedb$

Полученная перестановка  $R$  порядка  $m$  может быть представлена квадратной  $m \times m$  бинарной матрицей перестановки и использована при решении математических задач с матрицами [4].

Обратная задача — кодирование конкретных перестановок двоичными кодами — может быть сформулирована как задача восстановления двоично-факториальной записи с последующим преобразованием в двоичный номер.

Свойство *симметрии* перестановки относительно инверсии индексов  $[k,i] \rightarrow [i,k]$  интерпретируем как задачу восстановления исходного размещения элементов. Таким образом, повторное преобразование номера в перестановку позволяет восстановить исходное размещение с инверсным изменением порядка использования индексов. Для примера, прямое преобразование  $Q[5] \rightarrow R[5]$  с парами индексов  $[k, i] = \{[4,4], [3,0], [2,3], [1,1], [0,2]\}$  формирует перестановку  $Q = (a,b,c,d,e) \rightarrow R = (a,c,e,d,b)$ .

Применение инверсных индексов  $[k,i] = \{[4,4], [0,3], [3,2], [1,1], [2,0]\}$  к  $R=(a,c,e,d,b)$  восстанавливает исходное размещение  $Q=(a,b,c,d,e)$ .

Если на множестве  $Q$  из  $M$  элементов выбрано некоторое подмножество из  $1 \leq m < M$  элементов, а остальные представлены нулями, то применяя к ним все возможные перестановки, перечисляем все возможные их размещения  $(A_M^m)$  в  $Q$ . Для того чтобы ограничиться только разными вариантами размещения в алгоритме перестановок с использованием счетчика факториальных чисел, требуется сортировка, при которой исключаются перестановки нулей.

Все возможные сочетания из  $n$  элементов по  $m$  определяются двоичными кодами длиной  $n$ , где каждый разряд соответствует одному из  $n$  элементов конечного множества  $S$ . Мощность системы собственных подмножеств и кодов совпадает —  $2^n$ , включая пустое и единичное множества. Если задать начальное размещение  $Q$  через  $m$  единиц, а остальные — через  $(M - m)$  нулей, то все возможные перестановки единиц, за исключением перестановок нулевых элементов, будут обозначать все возможные сочетания  $(C_M^m)$  и определять как подмножества, состоящие из элементов в позициях, соответствующих единицам в этих перестановках.

**Задачи с перестановками**

1) *Коммутация каналов.*  $R$  можно интерпретировать как отображение  $R:Q \rightarrow R(Q)$ , ставящее в соответствие упорядоченному множеству элементов  $Q$  его перестановку  $R(Q)$ .

Полученную перестановку можно использовать для быстрой последовательной коммутации исходных ссылок  $\{a, b, c, d, e\}$ , размещаемых в  $Q = \{q_4, q_3, q_2, q_1, q_0\}$  входных каналах в выходные каналы  $R = \{r_4, r_3, r_2, r_1, r_0\}$ , выбирая их последовательно для заданной перестановки  $Q \rightarrow R$ . Значения множества  $Q$  можно рассматривать как начальные адреса массивов данных или ссылки на исполняемые функции. Программа, которая контролирует  $R$ , может выбрать соответствующую ссылку для приема массива данных или выполнения заданной функции.

В схемной реализации источники данных в соответствии с предложенным последовательным алгоритмом формирования пар индексов  $(i, j)$ ,  $i \in Q, j \in R$  из  $Q[5]$  подключаются через мультиплексор  $MX_Q$  к шине данных Bus, которая через распределитель  $R[5]$  подсоединяется к входам пяти приемников  $R$  последовательно. Мультиплексор задается индексом  $i$ , распределитель данных по приемникам — индексом  $j$ .

В стандарте преобразования кодов при передаче данных по Bluetooth-интерфейсу [5] используются схемы с парными перестановками на двухвходовых мультиплексорах — для формирования перестановок в  $4 \rightarrow 4$  и  $5 \rightarrow 5$  коммутаторах. При этом используются 14 управляющих сигналов и 14 мультиплексоров. В работе [5] не приведена систематическая процедура построения схемы в общем случае для конкретной перестановки, поэтому схема избыточна. Количество доступных кодов управления для выбора перестановок для четырех каналов  $2^{14} \gg 5!$

Параллельная схема выбора перестановок может быть выполнена из  $n$  мультиплексоров с  $n$  входами. Входы задаются  $\lceil \log_2 n \rceil$  управляющими сигналами. Использование кодов адресов при последовательном выборе входных каналов для  $i = 1, 2, \dots, n$  мультиплексоров исключает повторную выборку, если исключается адрес, использованный в предыдущей выборке. При этом нарушается естественный последовательный порядок перечисления перестановок. Однако адрес  $k$  для  $i$ -го мультиплексора в конкретной перестановке можно визуально определить для известной пары индексов  $(k, i)$  или в алгоритме формирования перестановки по ее факториальному номеру.

2) *Автоматизация проектирования.* Задача конструкторского проектирования топологии печатных плат связана с размещением однотипных электронных компонентов, например микросхем, на многослойной плате, и трассировкой соединений по выбранному критерию.

Алгоритмически задача решается перебором перестановок  $R$ , для каждой из которых вычисляется значение функционала  $F(R)$ , определяющего, например, число электрических переходов с одного слоя платы на другой или суммарную длину электрических соединений, и выбором перестановки  $R_0$ , на которой достигается минимум этого функционала [6, 7].

Здесь также применимы локальные размещения и перестановки с учетом конструктивных ограничений.

3) *Защита информации.* Используется алгоритм преобразования 8-разрядных двоичных кодов, каждый из  $f_i$  разрядов в котором трактуется в факториальной системе как позиция  $i$ -го бита в двоичном коде некоторой перестановки битов в байте.

Для решения задачи защиты доступа, устранения искажений и восстановления информации в системах передачи и хранения данных может быть использована перестановка битов кодов. Кодирование перестановками обеспечивает высокую защищенность символьных данных. Уровень защищенности можно оценить объемом вычислений, необходимых для идентификации кода перестановки. Для повышения надежности и защиты возможно применение массива перестановок, такой подход использовался при защите программ в микроконтроллерах Mcs51FA фирмы Intel [8].

4) *Хеширование в распределении памяти данных в компьютерах.* В высокопроизводительных ЭВМ используется страничная организация памяти с формированием адресов при распределении страниц методом хеширования, предполагающим случайный неповторяющийся выбор с использованием перестановок битов. Этот метод позволяет снизить вероят-

ность попадания в занятую страницу, но требует затрат времени на выбор нового адреса или перезагрузки. С применением предлагаемого метода формирования случайных перестановок битов кода эти затраты времени могут быть минимизированы.

5) *Задачи анализа надежности вычислительных систем.* Предлагаемый алгоритм можно использовать при построении и исследовании устройств реконфигурации отказоустойчивых многофункциональных вычислительных систем на основе перестановок настроек [9, 10].

**Заключение.** В статье рассмотрены свойства факториальной системы счисления, которые могут быть использованы в разных приложениях. Предлагаются алгоритмы основных процедур преобразования целых чисел в факториальные и способы их использования для формирования перестановок и коммутации применительно к заданным исходным размещениям множества элементов.

#### СПИСОК ЛИТЕРАТУРЫ

1. Системы счисления. Факториальная система [Электронный ресурс]: <[http://numsy.ru/education-page/factorial\\_system.html](http://numsy.ru/education-page/factorial_system.html)>.
2. Скорубский В. И., Экало Ю. В. Факториальная система счисления // Изв. вузов. Приборостроение. 1977. Т. 20, № 5. С. 70—73.
3. А. с. № 766011. Коммутатор / М. Л. Александров, Г. Д. Андреев, В. И. Скорубский, Ю. В. Экало. Госреестр изобретений СССР. 28.05.1980.
4. Ланкастер П. Теория матриц. М.: Наука, 1982. 272 с.
5. Specification of the Bluetooth System. 2014. 1081 p.
6. Андреев Г. Д., Экало Ю. В. Алгоритм оптимального и локально-оптимального разбиения печатных соединений на два слоя платы // Обмен опытом в радиопромышленности. 1977. № 2. С. 33—37.
7. Андреев Г. Д., Скорубский В. И., Экало Ю. В. Алгоритм размещения // Применение вычислительных машин в проектировании и производстве печатного монтажа. Л.: ЛДНТП, 1975.
8. Сташин В. В., Урусов А. В., Мологонцева О. Ф. Проектирование цифровых устройств на однокристалльных микроконтроллерах. М.: Энергоатомиздат, 1990. 223 с.
9. Богатырев В. А. О модификации функции „перманент матрицы“ и ее применении в комбинаторных методах анализа надежности вычислительных систем // Информационные технологии. 2002. № 1. С. 5—11.
10. Богатырев В. А. Надежность вычислительных систем с функциональной реконфигурацией на основе перераспределения задач // Информационные технологии. 2001. № 7. С. 22—27.

#### Сведения об авторах

- Владимир Иванович Поляков** — канд. техн. наук, доцент; Университет ИТМО, кафедра вычислительной техники; E-mail: v.i.polyakov@mail.ru
- Владимир Иванович Скорубский** — канд. техн. наук, доцент; Университет ИТМО, кафедра вычислительной техники; E-mail: vlis@km.ru
- Юрий Владимирович Экало** — канд. техн. наук, доцент; ОАО „НИЦ ЭТУ“; генеральный директор; E-mail: yury.ekalo@nicetu.ru

Рекомендована кафедрой  
вычислительной техники

Поступила в редакцию  
30.06.14 г.

**Ссылка для цитирования:** Поляков В. И., Скорубский В. И., Экало Ю. В. Применение факториальной системы для решения комбинаторных задач // Изв. вузов. Приборостроение. 2015. Т. 58, № 6. С. 436—442.

## APPLICATION OF FACTORIAL SYSTEM TO COMBINATORIAL PROBLEMS SOLVING

V. I. Polyakov<sup>1</sup>, V. I. Skorubsky<sup>1</sup>, Yu. V. Ekalo<sup>2</sup><sup>1</sup>ITMO University, 197101, Saint Petersburg, Russia

E-mail: v.i.polyakov@mail.ru

<sup>2</sup>St. Petersburg Electro-Technical University, Research and Engineering Center OJSC,  
197376, Saint Petersburg, Russia

Properties of the factorial number system are analyzed and several new fields of its application to combinatorial problems in programming and computer systems design are considered. Factorial numbers may be univalently enumerated with decimal or binary integers and interpreted as permutations of elements of any type. Decoding of binary-coded record of permutation in the factorial number system makes it possible to relate each permutation to a unique distribution of discrete objects (elements, references, etc.) in a specific discrete space. Permutations and distributions may be used in solving the problems of automation of design and commutation of calculating system channels, as well as in data protection.

**Keywords:** permutations, factorial number system, binary coding of factorial numbers, transformation of binary number into a factorial one and back, distributions, combinations of elements.

**Data on authors**

- Vladimir I. Polyakov** — PhD, Associate Professor; ITMO University, Department of Computer Technique; E-mail: v\_i\_polyakov@mail.ru
- Vladimir I. Skorubsky** — PhD, Associate Professor; ITMO University, Department of Computer Technique; E-mail: vlis@km.ru
- Yury V. Ekalo** — PhD, Associate Professor; St. Petersburg Electro-Technical University, Research and Engineering Center OJSC; Director General; E-mail: yury.ekalo@nicetu.ru

**Reference for citation:** Polyakov V. I., Skorubsky V. I., Ekalo Yu. V. Application of factorial system to combinatorial problems solving // Izvestiya Vysshikh Uchebnykh Zavedeniy. Priborostroenie. 2015. Vol. 58, N 6. P. 436—442 (in Russian).

DOI: 10.17586/0021-3454-2015-58-6-436-442