Journal of the Nigerian Association of Mathematical Physics
Volume **16 (May, 2010), pp 585 - 590**

A Fast Algorithm for Generating Permutation Distribution of Ranks in a K-Sample Experiment.

**F. Ewere and S.M. Ogbonmwan*
Department of Mathematics, University of Benin,
Benin City, Nigeria.
*Corresponding author: **e-mail:** frewere2@yahoo.com Tel.+2348058504893

Abstract

An algorithm for generating permutation distribution of ranks in a k-sample experiment is presented.The algorithm is based on combinatorics in finding the generating function of the distribution of the ranks. This further gives insight into the permutation distribution of a rank statistics. The algorithm is implemented with the aid of the computer algebra system Mathematica.

Key words: Combinatorics, generating function, permutation distribution, rank statistics, partitions, computer algebra.

1.0 Introduction

An exhaustive permutation distribution of a test statistic is necessary in the construction of exact tests of significance, thus controlling the risk in decision making. The unconditional exact permutation approach turns out to be the only possible way of constructing exact tests of significance for a general class of problems especially when complete enumeration is possible, see 11], [13] and [15]. But, generating the associated permutation sample space in order to apply the permutation test has always been a problem especially for fairly large sample sizes. This difficulty is mainly due to the logical and computational requirement necessary to develop and implement exact permutation scheme. Hence other approaches have been developed over the years. For detailed discussion along these lines, see [4], [5], [6] ad [12] on Monte Carlo methods, [2] and [17] on Bayesian and Likelihood approaches. The aim of this paper is to provide an algorithm which circumvents the difficulty associated with generating the permutation sample space, thus offering possibility of constructing exact test of significance of a rank statistic. Combinatorial problems which are very essential in finding the distribution of ranks in a k-sample experiment are clearly stated. The generating functions for the distribution of the ranks are obtained and it is shown how these calculations can be performed with the computer algebra system Mathematica. It transpires that the use of computer algebra opens new horizons for nonparametric statistics. Instead of time consuming calculations with recurrences, exact distributions can be found very fast from generating functions with the aid of a computer algebra system.

This paper is organized as follows. S ection 2 gives a brief discussion on integer partition. Two combinatorial problems are stated in section 3. We link these combinatorial problems to the partitions of integers. This forms the basis of the proposed algorithm. Section 4 provides the Mathematica implementation of the algorithm. In section 5, the computational efficiency of the proposed algorithm is investigated. The conclusion of this paper is given in section 6. Throughout this paper, command statements appear in a separate font and are written in terms of Mathematica 6.0 code. All calculations were performed on an intel Pentium M computer with a processor speed of $1.73GH_z$.

2.0 Integer Partitions

A partition of a positive integer n is defined as away of writing n as the sum of positive integers. Let $P(n)$ denote the number of partitions of n. For example, $P(5)=7$. An explicit formula for $P(n)$ valid for all positive integers n was discovered by [14], but since it is a complicated infinite series and is not needed for the purpose of this paper, it will be omitted here. However, there exist a simple generating function for $P(n)$, that is, a function which when expanded into a power series $\sum_{n=0}^{\infty} C_n x^n$ has its general coefficient $C_n = P(n)$. This generating

function is given by:

$$\frac{1}{\prod_{n=1}^{\infty}(1-x^n)} = \sum_{n=0}^{\infty} P(n)x^n = 1 + x + 2x^2 + 3x^3 + 5x^4 + \ldots \quad (2.1)$$

See [7]. The function $P(n)$ is referred to as the number of unrestricted partitions of n, to make clear that no restrictions are imposed upon the way in which n is partitioned into parts. A very interesting and perhaps the most interesting part of the theory of partitions concerns restricted partitions, that is, partitions in which some kind of restrictions is imposed upon the parts. For further discussions on unrestricted partitions, see ([1] and [10]) and in the case of restricted partitions, we refer to([8] and [18]).

Now, let $P(n,k)$ represents the number of ways of writing n as a sum of exactly k terms. $P(n,k)$ can be computed from the recurrence relation

$$P(n,k) = P(n-1,k-1) + P(n-k,k) \quad (2.2)$$

With $P(n,k)=0$ for $k>n, P(n,n)=1$ and $P(n,0)=0$, see [16]

3.0 Combinatorial Problems

A large class of problems in Combinatorial Mathematics is concerned with computing the number of ways in which some well-defined operation can be performed. The notions of combinations and permutations are the simplest and yet most fundamental concepts in the study of the theory of enumeration. Other enumerative techniques include generating functions, recurrence relations, the principle of inclusion and exclusion, Polya's theory of counting. For detailed discussion of these concepts, see [9]. Usually, a generating function that gives the number of combinations or permutations is called an enumerator.

Many nonparametric test statistics are of a combinatorial nature, especially those based on ranks. With this remark, it becomes obvious that a knowledge of the combinatorics of the permutation distribution of ranks in k-sample experiment offers useful insight into the exact permutation distribution of a rank statistic. For example, the distribution of the Wilcoxon Rank Sum statistic $W_{m,n}$ can be linked to partitions of integers which has a combinatorial interpretation.

Under $H_O : F = G$, all rank orders in the combined sample are equiprobable. Thus,

$$\Pr(W_{m,n} = k) = \frac{P(m,n,k)}{\binom{m+n}{m}}, \quad (3.1)$$

$P(m,n,k)$ denotes the number of ways a subset of {0, 1, 2, …, n } can be chosen with m elements such that the elements of this subset add up to k. In combinatorial terminology, $P(m,n,k)$ is nothing but the number of partitions of k with at most m non-zero blocks of maximal size n, see [1] and [3]. This connection was already noted by [19], but is hardly used in the statistical literature.

In what follows, two combinatorial problems which give essential idea in finding the permutation distribution of ranks in a k-sample experiment are stated.

Problem 1

Suppose there n observations which are ranked 1, 2, 3,…, n. In how many different ways is it possible to divide these n observations among k samples such that the ith sample T_i contains n_i observations and the sum of the ranks of these n_i observations in sample T_i is r_i with $n = n_1 + n_2 + ... + n_k$ and $r = r_1 + r_2 + ... + r_k = \frac{1}{2} n(n+1)$? Let the number be:

$$P[nlist, rlist] := P[\{n_1, n_2, ..., n_k\}, \{r_1, r_2 ..., r_k\}] \quad (3.2)$$

$P[nlist, rlist]$ can be calculated by counting the relevant partitions.

There are $\frac{(n_1 + n_2 + ... + n_k)!}{n_1! n_2! ..., n_k!}$ possible permutations of the n variates of the k samples of sizes

$n_{i,} i = 1,2,...,k$ which are equally likely with probability $\left(\frac{n!}{n_1! n_2! ..., n_k!} \right)^{-1}$. The number $P[nlist, rlist]$ can be obtained easily for small n and k by counting the relevant partitions, for example, $P[\{3,2\}, \{8,7\}] = 2$ which requires only 10 distinct arrangements (partitions). However, when n and k are not as small as in the above example, this method of obtaining $P[nlist, rlist]$ fails because of the large associated permutation sample spaces. For instance, when $n_1 = 10, n_2 = 7, n_3 = 3$, there are 22,170,720 distinct arrangements of the ranks. Admittedly, it is very difficult to carry out this enumeration manually in order to compute $P[nlist, rlist]$.

To overcome this problem, the generating function for the number $P[nlist, rlist]$ is obtained. Let $x[i]$ be a variable governing the number of observations in the ith sample and $y[i]$ be a variable governing the sum of the ranks of the observations in the ith sample. With this remark, the generating function for the number $P[nlist, rlist]$ is

$$p[n,k] = \prod_{j=1}^{n} \left(\sum_{i=1}^{k} x[i] y[i]^j \right)$$

(3.3)

However, this method of enumeration is not as fast as one would expect due to the fact that the number of terms of the generating function in (3.3) are of order k^n which is not too small even if n and k are not very large.

To improve on the computational efficiency of (3.3), let $nlist = \{n_1, n_2, ..., n_k\}$. In this case, the new generating function $p[nlist]$ for the number $P[nlist, rlist]$ have number of terms whose order is only $Multinomial[n_1, n_2, ..., n_k]$ which is smaller than k^n. Clearly, this new generating function $p[nlist]$ are the coefficients of $\prod_{i=1}^{k} x[i]^{n_i}$ of the generating function $p[n,k]$. To speed up computations, the generating function $p[nlist]$ is defined recursively as:

$$p[nlist] = p[\{n_1, n_2, ..., n_k\}] = \sum_{i=1}^{k} y[i]^n \, p[\{n_1, n_2, ..., n_i - 1, ..., n_k\}]$$

(3.4)

Problem 2

Suppose again that there are n observations numbered in such a way that d_1 of these observations are ranked m_1, d_2 have ranks $m_2, \ldots,$ and d_l have ranks m_l so that $n = d_1 + d_2 + \ldots + d_l$. One can find the different ways that is possible to divide these n observations among k samples such that the ith sample contains n_i observations and the sum of the ranks of these n_i observations is r_i with

$n = n_1 + n_2 + \ldots + n_k = d_1 + d_2 + \ldots + d_l$

$r = r_1 + r_2 + \ldots + r_k = d_1 m_1 + d_2 m_2 + \ldots + d_k m_k$

Let this number be denoted by:

$$Q[dlist, mlist, nlist, rlist] := Q[\{d_1, d_2, \ldots, d_l\}, \{m_1, m_2, \ldots, m_k\}, \{n_1, n_2, \ldots, n_k\}, \{r_1, r_2, \ldots, r_k\}] \quad (3.5)$$

Once again, it is difficult to calculate (3.5) manually if n and k are not small. To handle the problem that arises for fairly large n and k, the generating function for the number $Q[dlist, mlist, nlist, rlist]$ is derived. This generating function is given by:

$$q[dlist, mlist, k] = \prod_{j=1}^{l} \left(\sum_{i=1}^{k} x[i] y[i]^{m_j} \right)^{d_j} \quad (3.6)$$

The number of terms in (3.6) is still of order k^n and following same argument in problem 1, a new generating function $q[dlist, mlist, nlist]$ is introduced. This generating function improves on the computational efficiency of (3.6) as the number of terms is only of order $Multinomial[n_1, n_2, \ldots, n_k]$. This generating function is given and defined recursively as:

$$q[dlist, mlist, nlist] = q[\{d_1, d_2, \ldots, d_l\}, \{m_1, m_2, \ldots, m_l\}, \{n_1, n_2, \ldots, n_k\}] =$$
$$\sum_{i=1}^{k} y[i]^{m_l} q[\{d_1, d_2, \ldots, d_l - 1\}, \{m_1, m_2, \ldots, m_l\}, \{n_1, n_2, \ldots, n_i - 1, \ldots, n_k\}] \quad (3.7)$$

Obviously, the number $Q[dlist, mlist, nlist, rlist]$ is obtained by selecting the coefficient of $\prod_{i=1}^{k} y[i]^{r_i}$ of the generating function in (3.7). In what follows, the Mathematica procedures for obtaining the generating functions in (3.3), (3.4), (3.6) and (3.7) is provided. With these generating functions, the numbers in (3.2) and (3.5) are easily obtained.

4.0 Mathematica Session

In this section, the Mathematica procedures with numerical examples for obtaining the generating functions and the numbers $P[nlist, rlist]$, $Q[dlist, mlist, nlist, rlist]$ given above is presented.

```
xlist[k_] := Table[x[i], {i, 1, k}];
ylist[k_] := Table[y[i], {i, 1, k}];
p[1, k_] := xlist[k].ylist[k];
p[n_, k_] := p[n, k] = p[n - 1, k] * xlist[k].ylist[k] ^ n
q[{d_}, {m_}, k_] := (xlist[k].ylist[k] ^ m) ^ d;
q[dlist_, mlist_, k_] := q[dlist, mlist, k] =
        q[{First[dlist]}, {First[mlist]}, k] q[Rest[dlist], Rest[mlist], k]
```

```
Expand[p[5, 2]]
```

```
x[1]^5 y[1]^15 + x[1]^4 x[2] y[1]^14 y[2] + x[1]^4 x[2] y[1]^13 y[2]^2 + x[1]^4 x[2] y[1]^12 y[2]^3 + x[1]^3 x[2]^2 y[1]^12 y[2]^3 + x[1]^4 x[2] y[1]^11 y[2]^4 + x[1]^3 x[2]^2 y[1]^11 y[2]^4 +
 x[1]^4 x[2] y[1]^10 y[2]^5 + 2 x[1]^3 x[2]^2 y[1]^10 y[2]^5 + 2 x[1]^3 x[2]^2 y[1]^9 y[2]^6 + x[1]^2 x[2]^3 y[1]^9 y[2]^6 + 2 x[1]^3 x[2]^2 y[1]^8 y[2]^7 + x[1]^2 x[2]^3 y[1]^8 y[2]^7 +
 x[1]^3 x[2]^2 y[1]^7 y[2]^8 + 2 x[1]^2 x[2]^3 y[1]^7 y[2]^8 + x[1]^3 x[2]^2 y[1]^6 y[2]^9 + 2 x[1]^2 x[2]^3 y[1]^6 y[2]^9 + 2 x[1]^2 x[2]^3 y[1]^5 y[2]^10 + x[1] x[2]^4 y[1]^5 y[2]^10 +
 x[1]^2 x[2]^3 y[1]^4 y[2]^11 + x[1] x[2]^4 y[1]^4 y[2]^11 + x[1]^2 x[2]^3 y[1]^3 y[2]^12 + x[1] x[2]^4 y[1]^3 y[2]^12 + x[1] x[2]^4 y[1]^2 y[2]^13 + x[1] x[2]^4 y[1] y[2]^14 + x[2]^5 y[2]^15
```

```
Expand[q[{2, 3}, {1, 2}, 3]]
```

```
x[1]^5 y[1]^8 + 2 x[1]^4 x[2] y[1]^7 y[2] + 3 x[1]^4 x[2] y[1]^6 y[2]^2 + x[1]^3 x[2]^2 y[1]^6 y[2]^2 + 6 x[1]^3 x[2]^2 y[1]^5 y[2]^3 + 3 x[1]^3 x[2]^2 y[1]^4 y[2]^4 + 3 x[1]^2 x[2]^3 y[1]^4 y[2]^4 +
 6 x[1]^2 x[2]^3 y[1]^3 y[2]^5 + x[1]^2 x[2]^3 y[1]^2 y[2]^6 + 3 x[1] x[2]^4 y[1]^2 y[2]^6 + 2 x[1] x[2]^4 y[1] y[2]^7 + x[2]^5 y[2]^8 + 2 x[1]^4 x[3] y[1]^7 y[3] + 2 x[1]^3 x[2] x[3] y[1]^6 y[2] y[3] +
 6 x[1]^3 x[2] x[3] y[1]^5 y[2]^2 y[3] + 6 x[1]^2 x[2]^2 x[3] y[1]^4 y[2]^3 y[3] + 6 x[1]^2 x[2]^2 x[3] y[1]^3 y[2]^4 y[3] + 6 x[1] x[2]^3 x[3] y[1]^2 y[2]^5 y[3] + 2 x[1] x[2]^3 x[3] y[1] y[2]^6 y[3] +
 2 x[2]^4 x[3] y[2]^7 y[3] + 3 x[1]^4 x[3] y[1]^6 y[3]^2 + x[1]^3 x[3]^2 y[1]^6 y[3]^2 + 6 x[1]^3 x[2] x[3] y[1]^5 y[2] y[3]^2 + 6 x[1]^3 x[2] x[3] y[1]^4 y[2]^2 y[3]^2 + 3 x[1]^2 x[2]^2 x[3] y[1]^4 y[2]^2 y[3]^2 +
 3 x[1]^2 x[2] x[3]^2 y[1]^4 y[2]^2 y[3]^2 + 12 x[1]^2 x[2]^2 x[3] y[1]^3 y[2]^3 y[3]^2 + 3 x[1]^2 x[2]^2 x[3] y[1]^2 y[2]^4 y[3]^2 + 6 x[1] x[2]^3 x[3] y[1]^2 y[2]^4 y[3]^2 + 3 x[1] x[2]^2 x[3]^2 y[1]^2 y[2]^4 y[3]^2 +
 6 x[1] x[2]^3 x[3] y[1] y[2]^5 y[3]^2 + 3 x[2]^4 x[3] y[2]^6 y[3]^2 + x[2]^3 x[3]^2 y[2]^6 y[3]^2 + 6 x[1]^3 x[3]^2 y[1]^5 y[3]^3 + 6 x[1]^2 x[2] x[3]^2 y[1]^4 y[2] y[3]^3 + 12 x[1]^2 x[2] x[3]^2 y[1]^3 y[2]^2 y[3]^3 +
 12 x[1] x[2]^2 x[3]^2 y[1]^2 y[2]^3 y[3]^3 + 6 x[1] x[2]^2 x[3]^2 y[1] y[2]^4 y[3]^3 + 6 x[2]^3 x[3]^2 y[2]^5 y[3]^3 + 3 x[1]^3 x[3]^2 y[1]^4 y[3]^4 + 3 x[1]^2 x[3]^3 y[1]^4 y[3]^4 + 6 x[1]^2 x[2] x[3]^2 y[1]^3 y[2] y[3]^4 +
 3 x[1]^2 x[2] x[3]^2 y[1]^2 y[2]^2 y[3]^4 + 3 x[1] x[2]^2 x[3]^2 y[1]^2 y[2]^2 y[3]^4 + 6 x[1] x[2] x[3]^3 y[1]^2 y[2]^2 y[3]^4 + 6 x[1] x[2]^2 x[3]^2 y[1] y[2]^3 y[3]^4 + 3 x[2]^3 x[3]^2 y[2]^4 y[3]^4 +
 3 x[2]^2 x[3]^3 y[2]^4 y[3]^4 + 6 x[1]^2 x[3]^3 y[1]^3 y[3]^5 + 6 x[1] x[2] x[3]^3 y[1]^2 y[2] y[3]^5 + 6 x[1] x[2] x[3]^3 y[1] y[2]^2 y[3]^5 + 6 x[2]^2 x[3]^3 y[2]^3 y[3]^5 + x[1]^2 x[3]^3 y[1]^2 y[3]^6 +
 3 x[1] x[3]^4 y[1]^2 y[3]^6 + 2 x[1] x[2] x[3]^3 y[1] y[2] y[3]^6 + x[2]^2 x[3]^3 y[2]^2 y[3]^6 + 3 x[2] x[3]^4 y[2]^2 y[3]^6 + 2 x[1] x[3]^4 y[1] y[3]^7 + 2 x[2] x[3]^4 y[2] y[3]^7 + x[3]^5 y[3]^8
```

```
p[{n1_}] := y[1]^(n1 (n1 + 1) / 2);
p[nlist_] := p[nlist] =
   Module[{n, k, id, nlistnew, pos, sub},
     n = Plus @@ nlist;
       k = Length[nlist];
      id = IdentityMatrix[k];
      nlistnew = Delete[nlist, Position[nlist, 0]];
     pos = Flatten[Position[nlist, j_ /; j > 0]];
      sub = Table[y[i] → y[pos[[i]]], {i, 1, Length[pos]}];
       If[FreeQ[nlist, 0],
        Expand[Sum[y[i]^n p[nlist - id[[i]]], {i, 1, k}]],
        p[nlistnew] /. sub]]
q[{d_}, {m_}, nlist_] := Apply[Multinomial, nlist] *
                           Apply[Times, ylist[Length[nlist]]^(m nlist)];
 q[dlist_, mlist_, nlist_] := q[dlist, mlist, nlist] =
  Module[{d, n, k, id, dlistnew, nlistnew, pos, sub},
   d = Plus @@ dlist;
   n = Plus @@ nlist;
   k = Length[nlist];
   id = IdentityMatrix[k];
   dlistnew = Append[Drop[dlist, -1], Last[dlist] - 1];
    nlistnew = Delete[nlist, Position[nlist, 0]];
    pos = Flatten[Position[nlist, j_ /; j > 0]];
   sub = Table[y[i] → y[pos[[i]]], {i, 1, Length[pos]}];
   Which[d ≠ n, Print["n≠ d1+ +dl"]; Abort[],
    !(FreeQ[nlist, 0]), q[dlist, mlist, nlistnew] /. sub,
    !(FreeQ[dlist, 0]), q[Drop[dlist, -1], Drop[mlist, -1], nlist],
    FreeQ[nlist, 0] && FreeQ[dlist, 0],
    Expand[Sum[y[i]^Last[mlist]
       q[dlistnew, mlist, nlist - id[[i]]], {i, 1, k}]]]]
  p[{8, 8}]
```

$$y[1]^{100} y[2]^{36} + y[1]^{99} y[2]^{37} + 2 y[1]^{98} y[2]^{38} + 3 y[1]^{97} y[2]^{39} + 5 y[1]^{96} y[2]^{40} + 7 y[1]^{95} y[2]^{41} + 11 y[1]^{94} y[2]^{42} + 15 y[1]^{93} y[2]^{43} + 22 y[1]^{92} y[2]^{44} + 28 y[1]^{91} y[2]^{45} + 38 y[1]^{90} y[2]^{46} + 48 y[1]^{89} y[2]^{47} +$$
$$63 y[1]^{88} y[2]^{48} + 77 y[1]^{87} y[2]^{49} + 97 y[1]^{86} y[2]^{50} + 116 y[1]^{85} y[2]^{51} + 141 y[1]^{84} y[2]^{52} + 164 y[1]^{83} y[2]^{53} + 194 y[1]^{82} y[2]^{54} + 221 y[1]^{81} y[2]^{55} + 255 y[1]^{80} y[2]^{56} + 284 y[1]^{79} y[2]^{57} + 319 y[1]^{78} y[2]^{58} +$$
$$348 y[1]^{77} y[2]^{59} + 383 y[1]^{76} y[2]^{60} + 409 y[1]^{75} y[2]^{61} + 440 y[1]^{74} y[2]^{62} + 461 y[1]^{73} y[2]^{63} + 486 y[1]^{72} y[2]^{64} + 499 y[1]^{71} y[2]^{65} + 515 y[1]^{70} y[2]^{66} + 519 y[1]^{69} y[2]^{67} + 526 y[1]^{68} y[2]^{68} +$$
$$519 y[1]^{67} y[2]^{69} + 515 y[1]^{66} y[2]^{70} + 499 y[1]^{65} y[2]^{71} + 486 y[1]^{64} y[2]^{72} + 461 y[1]^{63} y[2]^{73} + 440 y[1]^{62} y[2]^{74} + 409 y[1]^{61} y[2]^{75} + 383 y[1]^{60} y[2]^{76} + 348 y[1]^{59} y[2]^{77} + 319 y[1]^{58} y[2]^{78} +$$
$$284 y[1]^{57} y[2]^{79} + 255 y[1]^{56} y[2]^{80} + 221 y[1]^{55} y[2]^{81} + 194 y[1]^{54} y[2]^{82} + 164 y[1]^{53} y[2]^{83} + 141 y[1]^{52} y[2]^{84} + 116 y[1]^{51} y[2]^{85} + 97 y[1]^{50} y[2]^{86} + 77 y[1]^{49} y[2]^{87} + 63 y[1]^{48} y[2]^{88} +$$
$$48 y[1]^{47} y[2]^{89} + 38 y[1]^{46} y[2]^{90} + 28 y[1]^{45} y[2]^{91} + 22 y[1]^{44} y[2]^{92} + 15 y[1]^{43} y[2]^{93} + 11 y[1]^{42} y[2]^{94} + 7 y[1]^{41} y[2]^{95} + 5 y[1]^{40} y[2]^{96} + 3 y[1]^{39} y[2]^{97} + 2 y[1]^{38} y[2]^{98} + y[1]^{37} y[2]^{99} + y[1]^{36} y[2]^{100}$$

```
q[{5, 3, 2}, {1, 2, 3}, {3, 4, 3}]
```

$$30 y[1]^{8} y[2]^{6} y[3]^{3} + 80 y[1]^{7} y[2]^{7} y[3]^{3} + 130 y[1]^{6} y[2]^{8} y[3]^{3} + 80 y[1]^{5} y[2]^{9} y[3]^{3} + 30 y[1]^{4} y[2]^{10} y[3]^{3} + 60 y[1]^{8} y[2]^{5} y[3]^{4} + 150 y[1]^{7} y[2]^{6} y[3]^{4} +$$
$$360 y[1]^{6} y[2]^{7} y[3]^{4} + 270 y[1]^{5} y[2]^{8} y[3]^{4} + 180 y[1]^{4} y[2]^{9} y[3]^{4} + 30 y[1]^{3} y[2]^{10} y[3]^{4} + 15 y[1]^{8} y[2]^{4} y[3]^{5} + 120 y[1]^{7} y[2]^{5} y[3]^{5} + 320 y[1]^{6} y[2]^{6} y[3]^{5} +$$
$$420 y[1]^{5} y[2]^{7} y[3]^{5} + 270 y[1]^{4} y[2]^{8} y[3]^{5} + 80 y[1]^{3} y[2]^{9} y[3]^{5} + 35 y[1]^{7} y[2]^{4} y[3]^{6} + 240 y[1]^{6} y[2]^{5} y[3]^{6} + 320 y[1]^{5} y[2]^{6} y[3]^{6} + 360 y[1]^{4} y[2]^{7} y[3]^{6} +$$
$$130 y[1]^{3} y[2]^{8} y[3]^{6} + 35 y[1]^{6} y[2]^{4} y[3]^{7} + 120 y[1]^{5} y[2]^{5} y[3]^{7} + 150 y[1]^{4} y[2]^{6} y[3]^{7} + 80 y[1]^{3} y[2]^{7} y[3]^{7} + 15 y[1]^{5} y[2]^{4} y[3]^{8} + 60 y[1]^{4} y[2]^{5} y[3]^{8} + 30 y[1]^{3} y[2]^{6} y[3]^{8}$$

5.0 Computational efficiency

In this section, the computing efficiency of $p[n,k]$, $p[nlist]$ and $q[dlist,mlist,k]$, $q[dlist,mlist,nlist]$ for calculating the numbers $P[nlist,rlist]$ and $Q[dlist,mlist,nlist,rlist]$ respectively.

Table 5.1: Computation time in Mathematica in seconds

dlist,mlist,nlist,rlist	*nlist,rlist*	*P[nlist, rlist]*	*Q[dlist, mlist,nlist,rlist*	*p[n,k]*	*p[nlist]*	*q[dlist, mlist,k]*	*q[dlist,mlist,nlist]*
{5,4,6},{1,2,3},{8,7},{17,14}	{12,8},{120,90}	3436	1400	0.047	4.333×10^{-16}	0.016	4.719×10^{-17}
{2,3,1,4},{1,2,3,4},{5,3,2},{13,9,5}	{8,7,5},{90,70,50}	102404	132	34.33	9.35	0.032	1.377×10^{-17}

From the table above, it is clear that $p[nlist]$ and $q[dlist,mlist,nlist]$ are superior to $p[n,k]$ and $q[dlist,mlist,k]$ respectively since their computations are faster. Hence, calculating $P[nlist,rlist]$ and $Q[dlist,mlist,nlist,rlist]$ for fairly large values of *n* and *k* can be achieved within reasonable time.

6.0 Conclusion

Presented in this paper is an algorithm for generating the permutation distribution of ranks in a k-sample experiment. The method described is fast as revealed by the numerical examples given in this article. Other calculations for the numbers $P[nlist,rlist]$ and $Q[dlist,mlist,nlist,rlist]$ can be obtained in a similar fashion. The proposed algorithm gives insight into finding the exact permutation distribution of a rank statistic in a combinatorial sense.

References

[1] Andrews, G. E. (1976). The theory of Partitions, Addison-Wesley, Reading, Ma.

[2] Bayarri, M. J. and Berger, J. O. (2004). The interplay of Bayesian and frequentist analysis. Statistical Science, 19, 58-80.

[3] Comtet, L. (1974). Advanced Combinatorics: the art of finite and infinite expansions. Dordrecht, Netherland: Reidel, p. 307

[4] Efron, B. (1979). Bootstrap methods: another look at the jackknife. The Annals of Statistics, 7, 1-26.

[5] Efron, B. and Tibshirani, R.J. (1993). An introduction to the bootstrap. NY: Chapman and Hall.

[6] Hall, P. and Tajvidi, N. (2002). Permutation tests for equality of distributions in high dimensional settings. Biometrika, 89, 359-374.

[7] Hirschhorn, M. D. (1999).Another short proof of Ramanujan's Mod 5 Partition Congruences and more. American Mathematical Monthly, 106, 580-583.

[8] Lehmer, D.H. (1964). The machine tools of combinatorics in Applied Combinatorial Mathematics, Beckenbach (Ed.), Wiley, NY.

[9] Liu, C. L. (1968). Introduction to Combinatorial Mathematics. McGraw Hill, Inc.

[10] Nijenhius, A. and Wilf, H.S. (1975). Combinatorial Algorithms, Academic Press, NY.

[11] Odiase, J. I. and Ogbonmwan, S.M. (2005). An algorithm for generating unconditional exact permutation distribution for a two-sample experiment. Journal of Modern Applied Statistical Methods, 4, 319-332.

[12] Opdyke, J. D. (2003). Fast permutation tests that maximize power under conventional Monte Carlo sampling for pairwise and multiple comparisons. Journal of Modern Applied Statistical Methods, 2, 27-49.

[13] Pesarin, F. (2001). Multivariate Permutation tests. New York: John Wiley and Sons.

[14] Rademacher, H. A. (1937). On the partition function *P(n).* Proc. London Math. Soc(2), 43, 241-254.

[15] Scheffe, H. (1943). Statistical Inference in the nonparametric case. The Annals of Mathematical Statistics, 14, 305-332.

[16] Skiena, S. (1990). Implementing Discrete Mathematics, Combinatorics and Graph theory with Mathematica. Reading, Ma: Addison-Wesley.

[17] Spiegelhalter, D. J. (2004). Incorporating Bayesian ideas into health care evaluation. Statistical Science, 19, 156-174.

[18] White, J. S. (1970). Restricted partition generator, Alg. 374, Comm. ACM, 13, 120.

[19] Wilcoxon, F (1945). Individual comparisons by ranking methods, Biometrics 1, 80-83

[20] Wolfram Research, Inc. (2007). Mathematica 6.0, Champaign Illinois.