7th International Conference On Research And Educations In Mathematics (ICREM7)

A New Method for Starter Sets Generation by Fixing Any Element

Sharmila Karim
School of Quantitative Sciences
College of Art and Sciences
Universiti Utara Malaysia
06010 Sintok, Kedah, Malaysia
mila@uum.edu.my

Haslinda Ibrahim
Department of Mathematics
College of Science
P.O. Box 32038 Kingdom of Bahrain
linda.combinatorics@gmail.com

Hasimah Sapiri
School of Quantitative Sciences
College of Art and Sciences,
Universiti Utara Malaysia,
06010 Sintok, Kedah, Malaysia
hasimah@uum.edu.my

***Abstract*— a new permutation technique based on distinct starter sets was introduced by employing circular and reversing operations. The crucial task is to generate the distinct starter sets by eliminating the equivalence starter sets. Meanwhile new strategies for starter sets generation without generating its equivalence starter sets were developed and more efficient in terms of computation time compared to old method. However all these algorithms have limitations in terms of fixing element to construct the first set (starter set) to begin with. It would be interesting to derive new strategy by fixing an element in any position. A new method is developed for starter sets generation namely STARSET1 based on circular where any element can be selected randomly to be fixed. The result showed that no redundancy of starter sets is occurring and no equivalence starter sets are obtained.**

Keywords—starter sets, permutation, circular

I. INTRODUCTION

This permutation methods can be classified into two categories: (i) exchange-based techniques; (ii) non-exchange based techniques [1]. The exchange-based techniques generate the new permutation by making possible changes among two consecutive elements such as transposition of non-adjacent elements [2],[3], and transposition with adjacent element [4],[5],[6],[7],[8],[9]. Whilst non-exchange based techniques generate the new permutation with certain restriction such as lexicographic order [10], nested cyclic [11], and partial reversion [12],[13],[14]. Among all these methods, generating permutation under cycling restriction was the simplest method compared to other technique for non-exchange based. Furthermore, this technique also is powerful due to its simplicity [1]. Several work on generating permutation has been done using the cycling restriction idea [1],[14],[16]. The recent construction of generating permutation by cycling restriction method as in [14] and [15],[16] used the idea of fixing element in the first position. Although algorithm in [16] is more efficient in terms of computation time compared to [15], but all these algorithms have limitations in terms of fixing element to construct the first set (starter set) to begin with. Thus this work aims to model a strategy to construct starter set by fixing element in any position by employing circular technique.

II. SOME PRELIMINARIES

A. Definitions

The following definitions will be used throughout this paper.

Definition 2.1. A linear ordering of the elements of the set $[n]=[1,2,3,\cdots,n-1,n]$ is called a permutation.

Definition 2.2. The identity permutation $[1,2,3,\cdots,n-1,n]$, denoted by ε is the permutation that leaves all integers fixed.

Definition 2.3. A *starter set* is a set that is used as a basis to enumerate other permutations.

Definition 2.4. The *reverse set* is a set that is produced by reversing the order of permutation set.

Definition 2.5. A *latin square* of order n is an $n\times n$ array in which n distinct symbols are arranged where each symbol occurs once in each row and column.

Definition 2.6. *The circular permutation* (CP) of order n is a latin square of order n.

Definition 2.7. The *reverse of circular permutation* (RoCP) is also a latin square of order n which is obtained by reversing arrangement element in each row of circular permutation.

Definition 2.8. An *equivalence starter set* is a set that can produce the same permutation from other starter set.

B. Starter Set Enumeration

In this section, we demonstrate the enumeration of starter sets by fixing any element under circular strategy for case n = 4 and 5.

Let S be the set of n elements i.e $(1, 2, 3, ..., n-1, n)$

Our previous method is as follows:

Let n = 5, and S = (1, 2, 3, 4, 5) and their starter as follows:

1 2 4 5 3	**1 2 5 3 4**	**1 2 3 4 5**
1 4 5 3 2	1 5 3 4 2	1 3 4 5 2
1 5 3 2 4	1 3 4 2 5	1 4 5 2 3
1 3 2 4 5	1 4 2 5 3	1 5 2 3 4

where the first element is fix in first position. The process of generating starters based on 3 circular and 4 circular of the element recursively.

Next we attempt to fix any element for circular method on linear ordering arrangement of permutation but it is complicated. Alternatively we rewrite the permutation as m-cycles where the $3 \leq m \leq n$-1. The selection of the m value is as recursive way where we have to start with 3, 4, 5, until n-1 which cannot be random selection.

Let denote the position of the element to be fixed by selection is i. Let consider the presentation of permutation in n- cycle form, (1 2 3 4 5 ... n-1 n). For case i = 1 and n, we already discussed in our previous work in [17,18]

Example: let n = 4, i = 2. S = (1 2 3 4). Thus we decompose it to 3-cycle form as

(2)(1 3 4).

For case n =4, the number of starter sets is 3.

Case 1: i = 2 ,

(2)(1 3 4) → (1 2 3 4)

(2)(3 4 1) → (3 2 4 1)

(2)(4 1 3) → (4 2 1 3)

Their correspond equivalent starter is

(3 2 1 4)

(4 2 3 1)

(1 2 4 3)

Case 2: i = 3, 3 cycle form is (3) (1 2 4)

(3) (1 2 4) → (1 2 3 4)

(3)(2 4 1) → (2 4 3 1)

(3)(4 1 2) → (4 1 3 2)

Their correspond equivalent starter is

(1 4 3 2)

(2 4 3 1)

(4 2 3 1)

Consider n = 5 , $1<i<n$, $3 \leq m \leq 4$.

Case 1: i = 2, value of m are 3 and 4. Thus we will generate 3-cycle and then 4-cycle.

S = (1 2 3 4 5). Initially we need to generate 3-cycle. When i = 2, S = (1)(2)(3 4 5)

The first three initial starter set are

(1)(2)(3 4 5) → (1 2 3 4 5)

(1)(2)(4 5 3) → (1 2 4 5 3)

(1)(2) (5 3 4) → (1 2 5 3 4)

We are done with m = 3. Next is for m = 4. We reuse result the first initial starter set with 3-cycle and construct 4 cycle. The element 2 is stay fix.

(2)(1 3 4 5), (2)(1 4 5 3), (2)(1 5 3 4)

Thus 12 starter sets as follows:

First four starter sets:

(2)(1 3 4 5) → (1 2 3 4 5)

(2)(3 4 5 1) → (3 2 4 5 1)

(2)(4 5 1 3) → (4 2 5 1 3)

(2)(5 1 3 4) → (5 2 1 3 4)

Second four starter sets:

(2)(1 4 5 3) →(1 2 4 5 3)

(2)(4 5 3 1) →(4 2 5 3 1)

(2)(5 3 1 4) →(5 2 3 1 4)

(2)(3 1 4 5) →(3 2 1 4 5)

Third four starter sets:

(2)(1 5 3 4) → (1 2 5 3 4)

(2)(5 3 4 1) → (5 2 3 4 1)

(2)(3 4 1 5) →(3 2 4 1 5)

(2)(4 1 5 3) →(4 2 1 5 3)

Case 2: i = 3. It will repeat a previous step. The different is element to be fixed is 3. When i = 3, S = (1)(3)(2 4 5).

As usual, we develop 3-cycle.

The first 3 initial starter sets:

(1)(3)(2 4 5) →(1 2 3 4 5)

(1)(3)(4 5 2) →(1 4 3 5 2)

(1)(3)(5 2 4) →(1 5 3 2 4)

Next is for m = 4.

Thus 12 starter sets as follows:

First four starter sets:

(3)(1 2 4 5) →(1 2 3 4 5)

(3)(2 4 5 1) →(2 4 3 5 1)

(3)(4 5 1 2) →(4 5 3 1 2)

(3)(5 1 2 4) →(5 1 3 2 4)

Second four starter sets:

(3)(1 4 5 2) →(1 4 3 5 2)

(3)(4 5 2 1) →(4 5 3 2 1)

(3)(5 2 1 4) →(5 2 3 1 4)

(3)(2 1 4 5) →(2 1 3 4 5)

Third four starter sets

(3)(1 5 2 4) →(1 5 3 2 4)

(3)(5 2 4 1) →(5 2 3 4 1)

(3)(2 4 1 5) → (2 4 3 1 5)

(3)(4 1 5 2) →(4 1 3 5 2)

III. General Algorithm

Consider the algorithm presented from the circular strategy as procedure STARSET1 which is a recursion procedure with only one recursive call. The integer *n* be an initial inputs, and temp = 2 is an initial rank of the procedure STARSET1. At each stage of recursion, a rank for procedure STARSET1 is starting at 2 until *n*-1 where each recursion called updates the entries in the storage. The general algorithm for starter sets generation by fixing any elements which based on circular strategy as follows:

Let *S* be the set of n elements. *S* = (1,2,3,4,…,*k*,*k*+1,…,*n*-1,*n*). The element is selected to be fixed denoted as *b*.

Algorithm 1. STARSET1 algorithm

```
Procedure STARSET1(temp,b)
if (temp == n-1)
        print the result;
        return ;
    endif
    for (i = 1; i <= temp+1; i++)
        CP( temp+1);
        call STARSET1 (temp+1, b);
    endfor
```

Algorithm 2. CP(m)

```
for(k = 2; k<= m; ++k)
    old = num[k];
    num[k] = num[k+1];
    num[k+1] = old;
endfor
```

We also provide step by step process of STARSET1 procedure.

Step 1: Set (1,2,3,4,…, *k*,*k*+1,…,*n*-2,*n*-1,n) be an initial starter set.

Step 2: Select an element to be fixed as *b* where $1 \le b \le n$ and rewrite the initial starter set as (b,1,2,3,4,…, *k*,*k*+1,…,*n*-2,*n*-1,n)

Step 3: Identify the last three elements of each starter set in Step 2. By employing CP operation to last three elements on each starter sets in Step 1, the three distinct starter sets are obtained.

Step 4: Identify the last four elements of each starter set in Step 2. By employing CP operation to last four elements on each starter sets in Step 2, the 12 distinct starter sets are obtained.

.

.

.

Step n-2: Identify the last (*n*-1) elements of each starter set in Step (n-3). By employing to the last (*n*-1) elements on each starter set in Step (n-2), the $\frac{(n-1)!}{2}$ distinct starter sets are obtained. Example output from STARSET1 PROGRAM:

Enter the size of the element =5, element 2 is fixed.

42153

12534

52341

32415

12354

32541

52413

42135

32451

42513

52134

12345

Enter the size of the element =5, element 3 is fixed

24351

52314

15342

41325

12354

51342

45321

24315

41352

54321

25314

12345

Enter the size of the element =5, element 4 is fixed.

52143

35241

13542

21345

53241

15342

21543

32145

51342

25143

32541

13245

Conclusion

Enumeration of a strategy to generate starter sets namely circular operations by fixing any elements randomly showed that this strategy guarantee that the generated starter sets are distinct and their equivalence starter sets will not be produced. To list all $n!$ permutations, the circular and revering operation can be employed on those starter sets.

Acknowledgment

This research is financially supported by a research grant from the Universiti Utara Malaysia under RAGS Research Grant Scheme (Code S/O : 12673). The authors would also like to thanks to the reviewers for their time and thoughtful recommendations on the paper.

References

[1] S R. Sedgewick, "Permutation generation methods," Computing Surveys, 1977, pp. 137-164.

[2] Wells, "Generation Of Permutation By Transposition," Math Comp, 1961, pp. 192-195.

[3] B. Heap, "Permutation by Interchanges," Computer Journal, 1963, pp. 293-294.

[4] H. F. Trotter, "Algorithm 115. Perm," Comm. ACM , 1962, pp. 434-435.

[5] S. M. Johnson, "Generation of permutations by Adjacent Transpositions," Math. Comput., 1963, pp. 282-285.

[6] F. Ives, "Permutation enumeration: four new permutation algorithms," Comm. ACM, 1976, pp. 68-72.

[7] J. Gao, and D.Wang, "Permutation Generation: Two New Permutation Algorithms," 2003, available url:http://arxiv.org/abs/cs/0306025.

[8] O. Viktorov, "Permutation generation algorithm," Asian Journal of Information Technology , 2007, pp. 956-957.

[9] A. A. Barisenko, V. V. Kalashnikov, I. A. Kulik, and O. E. Goryachev, "Generation Of Permutations Based Upon Factorial Numbers," IEEE, 2008, pp. 57-61.

[10] J. Ord-Smith, "Generation of permutation sequences Part 1," Computer Journal , 1970, pp. 152- 155.

[11] G. Langdon, "An Algorithm for generating permutations," Communication of ACM , 1967, pp. 298-299.

[12] S. Zaks, "A New Algorithm For Generation Of Permutations," BIT, 24, 1984, pp. 196-204.

[13] D. Shin, "The permutation algorithm for non-sparse matrix determinant in symbolic computation," Proceeding on the 15th CISL winter workshop, Japan, 2002, pp. 42-54.

[14] K. Thongchiew, "A Computerize Algorithm For Generating Permutation And It's Application In Determining A Determinant," Proceeding of World Academy of Science, Engineering and Technology, 21, 2007, pp.178-183.

[15] H. Ibrahim, Z. Omar, and A. Mohd Rohni, "New Algorithm for Listing All Permutations," Modern Sciences, 2010, pp. 89-94.

[16] S. Karim, Z. Omar, and H. Ibrahim, "Integrated Strategy For Generating Permutation," International Journal of Contemporary Mathematical Sciences, Vol. 6, no. 24, 2011, pp. 1167 - 1174.

[17] S. Karim, "New Sequential And Parallel Division Free Methods For Determinant Of Matrices", Ph.D Thesis, Universiti Utara Malaysia, 2013.

[18] S. Karim, H. Ibrahim, and Z. Omar. Circular to the Left Strategy for Starter Sets Generation, Proceeding of *International* Conference on the Analysis Mathematical Application in Engineering and Science ,2014.