

Starter Set Generation Based on Factorial Numbers for Half Wing of Butterfly Representation

Sharmila Karim*, Haslinda Ibrahim

Department of Mathematics & Statistics, School of Quantitative Sciences, Universiti Utara Malaysia, 06010 UUM Sintok, Kedah, Malaysia

Received December 1, 2020; Revised June 21, 2021; Accepted July 13, 2021

Cite this paper in the following citation styles

(a): [1] Sharmila Karim, Haslinda Ibrahim, "Starter Set Generation Based on Factorial Numbers for Half Wing of Butterfly Representation," *Mathematics and Statistics*, Vol. 9, No. 5, pp. 664 - 668, 2021. DOI: 10.13189/ms.2021.090506.

(b): Sharmila Karim, Haslinda (2021). *Starter Set Generation Based on Factorial Numbers for Half Wing of Butterfly Representation. Mathematics and Statistics*, 9(5), 664 - 668. DOI: 10.13189/ms.2021.090506.

Copyright©2021 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

Abstract Permutation is an interesting subject to explore until today where it is widely applied in many areas. This paper presents the use of factorial numbers for generating starter sets where starter sets are used for listing permutation. Previously starter sets are generated by using their permutation under exchange-based and cycling based. However, in the new algorithm, this process is replaced by factorial numbers. The base theory is there are $\frac{(n-1)!}{2}$ number of distinct starter sets. Every permutation has its decimal number from zero until $n!$ for Lexicographic order permutation only. From a decimal number, it will be converted to a factorial number. Then the factorial number will be mapped to its corresponding starter sets. After that, the Half Wing of Butterfly will be presented. The advantage of the use of factorial numbers is the avoidance of the recursive call function for starter set generation. In other words, any starter set can be generated by calling any decimal number. This new algorithm is still in the early stage and under development for the generation of the half wing of butterfly representation. Case $n=5$ is demonstrated for a new algorithm for lexicographic order permutation. In conclusion, this new development is only applicable for generating starter sets as a lexicographic order permutation due to factorial numbers is applicable for lexicographic order permutation.

Keywords Starter Sets, Factorial Numbers, Permutation, Half Wing of Butterfly Representation

1. Introduction

Permutation was applied in the designation of sciences and technologies such as system security [1], and scheduling problems [2]. Recently noise-resistant code-based permutation was developed for reliable information transfer and storage [3]. On the other hand, research in a permutation is still reliable. There are many ways to generate a permutation such as; by making probable modifies among two consecutive elements [4,5]. The other techniques that create the new permutation in a certain way are the lexicographic order [6,7], partial reversion [8], cycle minor [9], and the set of reductions [10]. Meanwhile, our work is on generating permutation using its starter sets based on cycling [11,12,13] and exchange [14]. Our algorithms were successfully implemented to the parallel algorithm [15, 16] and recently we applied our algorithm in the graph theory area where specifically determining Half Wing of Butterfly representation [17,18]. Despite this, we still need to generate all starter sets to determine the Half Wing of Butterfly representation. Then we found research by Borysenko et. al,[19] mentioning that one effective way to generate permutation is by using factorial numbers.

In this paper, we attempt to generate Half Wing of Butterfly representation from a decimal number that represents a starter set. The difference of this paper from our previous work is that we employ factorial numbers to generate start sets instead of using them in a generated permutation. The challenge here is how to avoid the repetition of Half Wing of Butterfly representation by

selecting any decimal number from 0 until $(n-1)!$. Our base theory is there are $\frac{(n-1)!}{2}$ number of distinct starter sets [11]. Thus, we investigate and explore this theory along with its factorial numbers for generating distinct Half Wing of Butterfly representation under lexicographic order permutation.

The organization of the remaining details of the paper is as follows: In section 2, we recall some preliminary definitions. Then the factorial number system is explained in section 3. Section 4 and 5 are devoted to old, and new algorithms in generating starter sets and their application for determining half wing of butterfly representation. Some developing directions are proposed in conclusion.

2. Preliminary Definitions

Definition 1 Let $K_n = (V, E)$. A function $g = (x_1 \ x_2 \ x_3 \ \dots \ x_n)$ maps the vertices $x_i \rightarrow x_j$ where $x_i, x_j \in V \ 1 \leq i, j \leq n$:

Example 1. Let $g = \begin{pmatrix} 1234 \\ 2341 \end{pmatrix}$.

Then, $1 \mapsto 2, 2 \mapsto 3, 3 \mapsto 4, \text{ and } 4 \mapsto 1$. Thus, we write the mapping for g as $(1,2)(2,3)(3,4)(4,1)$.

Definition 2 Let A be a circuit with direction $(1,2,3, \dots, n-1, n, 1)$. Then, a circuit B is the mirror image of A if the direction of B is $(n, n-1, \dots, 3, 2, 1, n)$.

Definition 3 Let A and B be two circuits with n vertices. If B is the mirror image of A , then B is isomorphic to A .

Example 2:

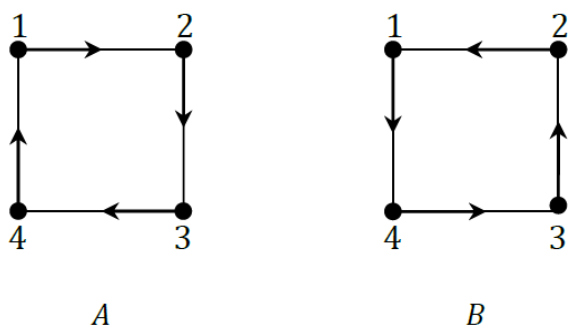


Figure 1. Circuit B is a mirror image of circuit A

Definition 4 A starter set is a set used as a base for listing other permutations.

Definition 5 The i -exchange is a restriction of an element in the i -th position, $2 \leq i \leq n-2$ exchange to the right starting with the element in $n-2$ position until 2-th position recursively for the permutation.

The next section is about the factorial number system in generating permutation.

3. Factorial Number System

A mixed-radix numeral system is this numbering system, sometimes called a factoradic number system. The term factorial number system is used by Knuth [20]. The factorial number has a form:

$$F_{\langle \varphi \rangle} = X_n n! + \dots + X_l l! + \dots + X_0 0! \quad (1)$$

Equation (1) is a mapping from the factorial number into a corresponding decimal number. A natural mapping occurs between the integers $[0, n! - 1]$ and permutations of n elements in lexicographic order, where the integers are represented in factoradic form, (or identical numbers of n digits in factorial representation). The Lehmer code (or inversion table) was known for this mapping [22] for the lexicographic order.

For example, with $n = 3$, the mapping is as follows:

Table 1. Factorial Number and its permutation

Decimal	Factorial Number	Permutation (Lexicographic order)
0	000	123
1	010	132
2	100	213
3	110	231
4	200	312
5	210	321

As can be seen from Table 1, 2 steps are required to obtain permutation from its corresponding decimal number [21]. The steps are as follows:

Step 1: Transform decimal number to factorial number.

Step 2: Transform factorial number to permutation.

Meanwhile, Kuo[7] enhanced these steps in his work and introduced a new number system called an ordinal number. His step is as follows:

Step 1: Transform decimal number to factorial number.

Step 2: Transform factorial number to ordinal number

Step 3: Transform ordinal number to permutation.

The example for his work. Let $n = 3$

Table 2. Factorial Number, ordinal number and its permutation for the case $n = 3$

Decimal	Factorial Number	Ordinal Number	Permutation (Lexicographic order)
0	000	111	123
1	010	121	132
2	100	211	213
3	110	221	231
4	200	311	312
5	210	321	321

Despite these two works, Thanh [23] stated that generation permutation based on factorial number system has weaknesses where the process is long and the algorithm is difficult to parallelize. He suggested an

inversion vector notion that is fast, simple, and highly feasible to parallelize the algorithm. But the algorithm did not involve a decimal number. The example of an inversion vector can be seen in Table 3.

Table 3. Inversion vector and its permutation for the case $n = 3$

No	Inversion vector	Permutation (Lexicographic order)
1	000	123
2	001	132
3	002	231
4	010	213
5	011	312
6	012	321

These three works are quite interesting and we combine their example for the case $n = 3$ in Table 4.

Table 4. Permutation and its number for the case $n = 3$

Decimal	Factorial Number	Ordinal Number	Permutation (Lexicographic order)	Inversion vector
0	000	111	123	000
1	010	121	132	001
2	100	211	213	010
3	110	221	231	002
4	200	311	312	011
5	210	321	321	012

We will refer to their works as a guideline in our algorithm development. In our method, for case, $n = 3$, only a single Half Wing of Butterfly representation. Thus, $n = 3$ is a trivial example for our study. Next is a discussion about our previous algorithm in designing Half Wing of Butterfly representation.

4. Previous Algorithm

In generating a distinct starter set and drawing the half wing of butterfly representation, it contains 3 steps and starts with identity permutation $(1,2,3,4,5,\dots,n)$ with n distinct elements. The steps are as follow:

Step 1: Fixing the 1st element under permutation.

Step 2: Generating the $\frac{(n-1)!}{2}$ distinct circuits using i -th

exchanged restriction.

Step 3: Finding the mapping and drawing half wing of butterfly.




We start enumeration with $n = 4$, and there are 3 starters sets under exchange restriction such as follows:

Table 5. Starter sets, its equivalence starter set, and Lexicographic order

Starter sets	Equivalence starter set	Lexicographic order, and its decimal number
1234	1432	1234 (0)
1324	1423	1243 (1)
1342	1243	1324(2)

Base on the decimal number for the Lexicographic order, supposedly our starter set numbers are 0, 1, and 2. Thus by calling decimal numbers 0, 1, and 2 are enough to develop distinct Half Wing of Butterfly representation.

Table 6. The mapping of starter sets to the distinct Half Wing of Butterfly representation for the case $n = 4$

Decimal Number	Starter set	Mapping	Half Wing
0	(1 2 3 4)	1→2→3→4→1: $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}$	
1	(1 2 4 3)	1→2→4→3→1: $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \end{pmatrix}$	
2	(1 3 2 4)	1→3→2→4→1: $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \end{pmatrix}$	

5. New Algorithm

We modified our previous algorithm where our inputs are the size of permutation and decimal number from 0 to $\frac{(n-1)!}{2} - 1$. We induced the Borisenko et.al[3] algorithm in our algorithm such as follows:

Step 1: Transform decimal number to factorial number.

Step 2: Transform factorial numbers to starter sets.

Step 3: Find the mapping and draw Half Wing of Butterfly.

Let's demonstrate $n = 5$, we choose number between $[0, 11]$, which is number 10.

Step 1: Transform decimal number to factorial number

$$10 \div 1 = 10, \text{ remainder } 0$$

$$10 \div 2 = 5, \text{ remainder } 0$$

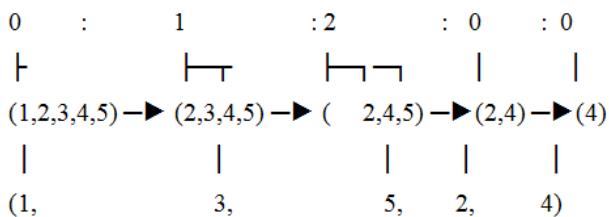
$$5 \div 3 = 1, \text{ remainder } 2$$

$$1 \div 4 = 0, \text{ remainder } 1$$

Reading the remainder backwards gives 1200. Since $n = 5$, we put '0' to the first position of the remainder and became 01200 to make sure it has 5 digits.

Step 2: Transform factorial numbers to starter sets.

0:1:2:0:0 → (13524)
Factorial



Thus, the corresponding starter sets are (13524).

Step 3: Find the mapping and drawing Half Wing of Butterfly.

Table 7. The mapping of starter sets (1 35 2 4) to the distinct Half Wing of Butterfly representation

Decimal Number	Starter set	Mapping	Half Wing
10	(1 35 2 4)	1→3→5→2→4→1 (1 2 3 4 5) (3 4 5 1 2)	

6. Conclusions

In this present paper, we proposed a new algorithm to generate starter sets via decimal numbers under the Lexicographic order and applied it for the representation of Half Wing of Butterfly. This new algorithm is under exploration for the application in cybersecurity specifically under error-correcting codes.

Acknowledgements

This research was supported by the Ministry of Education (MOE) through Fundamental Research Grant Scheme (FRGS/1/2018/STG06/UUM/02/15).

REFERENCES

[1] McCaffrey, J. (2003). Using permutations in NET for improved systems security. Microsoft Developer Network.

[2] Liao, C. J., Liao, L. M., & Tseng, C. T. (2006). A performance evaluation of permutation vs. non-permutation schedules in a flowshop. *International Journal of Production Research*, 44(20), 4297-4309

[3] Borysenko, A. A., Horiachev, O. Y., Matsenko, S. M., & Kobiakov, O. M. (2018, May). Noise-immune codes based on permutations. In 2018 IEEE 9th International Conference on Dependable Systems, Services and

Technologies (DESSERT) (pp. 609-612). IEEE.

[4] Gao, J., & Wang, D. (2003). Permutation generation: Two new permutation algorithms. arXiv preprint cs/0306025.

[5] Viktorov, O. (2007). Permutation generation algorithm. *Asian Journal of Information Technology*, 6(9), 956-957.

[6] Ord-Smith, R. J. (1968). Algorithms: Algorithm 323: Generation of permutations in lexicographic order. *Communications of the ACM*, 11(2), 117.

[7] Kuo, T. I. N. G. (2009). A new method for generating permutations in lexicographic order. *Journal of Science and Engineering Technology*, 5(4), 21-20.

[8] Thongchiew, K. (2007). A computerised algorithm for generating permutation and its application in determining a determinant. In *Proc. of World Academy of Science, Engineering and Technology* (Vol. 21, pp. 178-183)

[9] Monks, M. (2009). Reconstructing permutations from cycle minors. *the electronic journal of combinatorics*, R19-R19.

[10] Ginsburg, J. (2007). Determining a permutation from its set of reductions. *ArsCombinatoria*, 82, 55-68.

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

[12] Karim, S, Ibrahim, H. & Omar, Z. (2014). A circular to the left for starter set generation. *Proceedings of the International Conference Analytical and Mathematical Application in Engineering and Sciences*, 1-6.

[13] Karim, S, Ibrahim, H. & Omar, Z. (2015). A New Method for Starter Sets Generation by Fixing Any Element, *Proceeding of 7th International Conference on Research and Education in Mathematics*, 120-123.

[14] Karim, S, Ibrahim, H., Omar, Z. & Sapiri, H. (2016). New Method for Starter Sets Generation by Fixing Any Element Under Exchange Restriction. *International Journal of Applied Mathematics and Statistics*, Vol. 54(2), 132-141.

[15] Karim, S., Omar, Z., and Ibrahim, I. (2015). Efficient Parallel Algorithm for Listing Permutation with Message Passing Interface (MPI). *International Symposium on Mathematical Sciences and Computing Research*, 343-348.

[16] Karim, S., Omar, Z., and Ibrahim, I. (2019). A New Parallel Model For Starter Sets Generation Via Exchanging Two Elements, *AIP Conference Proceedings* 2138 (1), 030022.

[17] Karim, S., Ibrahim, H., and Darus, M.M. (2017). Determining Distinct Circuit In Complete Graphs Using Permutation, *AIP Conference Proceedings* 1905 (1), 030019

[18] Karim, S., Ibrahim, H., And Darus, M.M. (2019). Representation Of Half Wing Of Butterfly And Hamiltonian Circuit For Complete Graph Using Starter Set Method. *Journal Of Engineering And Applied Sciences*, 14: 7084-7088.

[19] Borysenko, O., Horiachev, O., Kulyk, I., & Yakovlev, M. (2019). Factorial Permutation Generation. In 2019 IEEE International Scientific-Practical Conference Problems of Info-communications, Science and Technology (PIC S&T)

- (pp. 375-378). IEEE.
- [20] Knuth, D. E. (2014). *Art of computer programming, volume 2: Seminumerical Algorithms*. Addison-Wesley Professional.
- [21] Borisenko, A.A, Kalashnikov, V. VKulik, ., I. A., and Goryachev, O. E. (2008). *Generation Of Permutations Based Upon Factorial Numbers*, IEEE, 57-61.
- [22] Ri, W., & Song, O.. (2021). A novel representation for permutations. *IEEE Transactions on Information Theory*, 67(3), 1920-1927. doi:10.1109/TIT.2020.3048905
- [23] Thanh, H. C. (2012). Parallel generation of permutations by inversion vectors. In *2012 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future* (pp. 1-4). IEEE.