Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 7, July 2014, pg.338 – 342

REVIEW ARTICLE

REVIEW ON LATIN SQUARE

Yukti Gupta [(1)], Prof. Aashima Bansal [(2)], Prof. Devdutt Baresary [(3)]

[1]Research Scholar (Department of Computer Science), GVIET, Banur

[2]Professor CSE, GVIET, Banur

[3]Professor CSE, GIET, Khanna

[1] eryuktigupta@gmail.com, [2] aashibansal86@gmail.com, [3] devinception@gmail.com

---

Abstract— Latin square is an n × n array having n different symbols, each one of them appears exactly once in each row and column respectively. It is widely used in steganography, cryptography, digital watermarks, computer games, sudoku, graph analysis, error correcting codes; generate magic squares, statistics and mathematical field. The Sudoku puzzles are a special case of Latin squares. Latin illustrates cayley graphs, hamiltonicity properties and the embedding of squares. It formalizes computation for both symmetric and non-symmetric trivial groups and expounds security aspects by quasigroup cipher. MALS specifies the scheduling channels for generating nodes in a network. Latin defines an optimal solution for 3DPAP. Latin square methodology is based on heuristic cell based technique and generates random Latin square using genetic algorithmic approach both consumes high processing time and decreases the throughput. In this paper there is a discussion on methodologies used for Latin square computation and Sudoku design.

Keywords— Latin Square, Cayley Graphs, Quasigroup, Sudoku, MALS, 3DPAP, Genetic Algorithm

---

I. INTRODUCTION

Latin square is an $n \times n$ array in which each cell contains at most one symbol, chosen form an n-set, such that each symbol occurs exactly once in each row and each column. The latin square name was inspired by Leonhard Euler mathematician, who use latin characters as the symbols. Latin square is said to be in standard form if the letters of left column and top row is in some standard order or in alphabetic order. Latin squares are said to be in orthogonal if one Latin square is superposed on other and each letter of both the squares coincide with each other. If the rows of one Latin square are the columns of the other Latin square then they are said to be conjugate. In other words, if the rows and columns of two Latin square are interchanged then it results into conjugate square. An adjugacy is a generalization of conjugacy that leads to the permutation of the constraints of one another [12]. The triple Latin square of nxn order is represented as (r, c, and s), where r represents the row, c represents the columns, and s depicts the symbol then we obtain a set of n^2 triples called the orthogonal array representation of the square. A cyclic Latin square of order n is derived by cyclic permutation of each row of degree n. A partial Latin square is said to be completed to a Latin square if its empty cells are filled to produce a Latin square. As of know there is no formula for easy computation for the number $L(n)$ of $n \times n$ Latin squares with symbol 1,2,...,n is:

$$\prod_{k=1}^{n} (k!)^{n/k} \geq L(n) \geq \frac{(n!)^{2n}}{n^{n^2}}$$

Here all the known exact values are given. The problem of determining if partially filled square can form a complete Latin square is NP completes [1].

n	Reduced Latin Squares of size n	All latin Squares of size n
1	1	1
2	1	2
3	1	12
4	4	576
5	56	161280
6	9408	812851200
7	16942080	61479419904000
8	535281401856	108776032459082956800
9	377597570964258816	5524751496156892842531225600
10	7580721483160132811489280	9982437658213039871725064756920320000
11	5363937773277371298119673540771840	776966836171770144107444346734230682311065600000

Figure1: The Number of Latin Squares of Various Sizes

Sudoku originally called as number place, is a logic based and combinatorial number placement puzzle. There is an additional restriction in sudoku that there must be nine 3×3 adjacent sub squares and it must contain the digits 1–9 in the standard form. The 9x9 grid consist of digits from 1 to 9 at each column, each row, and each of the nine 3×3 sub-grids exactly once [12]. The 3x3 grid also called blocks, boxes or sub-squares. The puzzle setter provides a partially completed grid, which for a well-posed puzzle has a unique solution. It is mainly used in the field of image authentication, encryption, digital watermark, secret image sharing with reversibility, etc.

II. Background And Related Work

The related work done on latin square is discussed in the discourse addressed. Extensive research can be carried out on analysing the processing time taken to compute the Latin square. In end of eighteen century, it was first discovered by Euler that the enumeration of the different arrangements of n letters in an nxn Latin square, in a square in which each letter occurs once in every row and column respectively. The transformation of any Latin square of degree n is having $(n!)^3$ways. Each reduced square generates a set of different n! (n-1)! squares, by permutation of all columns and rows except the first row. The square is said to be in standard form if the letters of top row and left column of a Latin square is in standard order like alphabetic order. Generally the number of squares in a set of orthogonal squares of side s is not greater than (s-1) are said to be of mutually orthogonal Latin square (MOLS) [10]. Each square letter may be permuted among themselves without destroying the property of orthogonal. If L_n be the number of latin squares of order n and R_n be the number of reduced latin squares of order n then reduced latin square first row and first column in lexicographic order [4]is represented as:-

$$L_n = n!(n-1)!R_n.$$

Here the Latin square L over omega is called in standard or reduced form if both the row a_1 and the column a1 consist the sequence a_1, a2,..., a_n. Hence the binary operation system has an identity element a_1, and is defined as quasigroup. Two Latin squares are called isomorphic over the same set omega if one Latin is obtained from other by combination of permutations of rows, columns, and the entry alphabets [5]. There is an algorithm for generating random Latin square of a given order based on proper set of moves that connect all the squares.

The equivalence classes of latin square can be obtained by performing various operations that change one latin square to another and have the equivalence relation between them Fig.2 [15]. The permutation of rows, columns and names of symbols of Latin square derived new Latin square which divide set of all Latin squares into subset [15]. Latin also includes random Cayley graphs properties which comprises their clique, independence, chromatic numbers and their expansion as well as their connectivity and hamiltonicity properties [4][13]. The number of reduced Latin squares of order n is divisible by f! here f is an integer close to 1/2n. The Latin square which belongs to non-trivial symmetry group tends to zero quickly when the order is increased [10]. The number of symmetric Latin squares is related with the security of the post commutative quasigroup cipher [17].

n	Main classes	Isotopy classes
1	1	1
2	1	1
3	1	2
4	2	2
5	2	2
6	12	22
7	147	564
8	283657	1676267
9	19270853541	115618721533
10	34817397894749939	208904371354363006
11	2036029552582883134196099	12216177315369229261482540

Figure2: Equivalence Classes of Latin square

In a Latin square there is a set of entries that have facility to embed in only one Latin square, those are critical sets. If embedding the Latin square is easy to find and the remaining Latin squares are forced once at a time then the critical set is strong. The semi-strong critical set is a foundation of a strong critical set. It has been proved that $[n^2/4]$ is the size of the smallest strong or semi-strong critical set of order n. The smallest critical set of a Latin square of order 6 is 9[6].

Despite that Latin square has an orthogonal mate if and only if it contains n disjoint transversals. Transversal defines the notions of complete mapping and orthomorphisms in quasigroup, and is fundamental for the mutually orthogonal Latin squares [7] [5]. The Latin square of even order has even number of transversals and latin square of odd order has at least one transversal. If the number of traversal in cyclic latin square of order n is t_n then there exist two real constants c_1 and c_2 such that

$$c_1^n n! \leq t_n \leq c_2^n n!$$

Where $0<c_1<c_2<1$ and $n >= 3$ is odd [7].
The pair of orthogonal Latin squares is called Graeco-latin square [6]. An Euler or Graeco latin square of order n over two sets S and T, each consist of n symbols in nxn arrangement of cells, containing an ordered pair (s,t) for each cell, where s is in the set S and t is in the set T, such that every row and column contains each element of S and each element of T exactly once respectively [10]. Moreover no two cells are having the same ordered pair.
Using the deterministic time division channel access scheduling, Latin square generates nodes in ad hoc network. A medium access based on Latin squares (MALS) is applied for both macro-time division channel access scheduling and micro-time division channel access scheduling [14]. Latin square also provides feasible solution for three dimensional planar assignment problems (3DPAP). For 3DPAP Latin square used the technique of genetic algorithm, for generating random Latin square [15].
A new approach is being studied in 2013 based on the equivalence between Latin square and maximum cliques of a graph and is also valid for Sudoku design. This algorithm could run upto order equal to 7 on a standard pc [7].The specialization of Latin square is called gerechte. Sudoku design is an example of gerechte design in which nxn grid is portioned into n regions, each containing n cells of the grid. The most common size of square is 9×9 and solving an instance of Sudoku problem is NP-complete [13]. So it is difficult to define a deterministic polynomial time algorithm for solving a given Sudoku puzzle of size nxn, where n is any integer. A framework to computing Ln (π) for a general pattern π of length equal to three , for π ∈ Son, in terms of the total number of Latin Squares. For any π ∈ Sn,

$$L_n(\pi) = \left(\frac{n!-n}{n!}\right)^2 L_n.$$

i) It is based on a proper set of moves that connect all the squares and make the distribution of visited square uniform.
ii) Every row and column is in a cyclic increasing or decreasing structure where adjacent elements differ by one (mod n) as shown in figure3 (a)(b) [16].

i	$i-1$			1	n			$i+2$	$i+1$
$i-1$	$\ddots$		1	n			$i+2$	$i+1$	i
		1	n			$i+2$	$i+1$	i	
	1	n	$\ddots$		$i+2$	$i+1$	i		
1	n			$i+2$	$i+1$	i			
n			$i+2$	$i+1$	i				1
		$i+2$	$i+1$	i		$\ddots$		1	n
	$i+2$	$i+1$	i				1	n	
$i+2$	$i+1$	i				1	n	$\ddots$	
$i+1$	i				1	n			$i+2$

Fig.3 (a) the general form of a 123, 231 or 312 avoiding Latin Square

i	$i+1$			n	1			$i-2$	$i-1$
$i+1$	$\ddots$		n	1			$i-2$	$i-1$	i
		n	1			$i-2$	$i-1$	i	
	n	1	$\ddots$		$i-2$	$i-1$	i		
n	1			$i-2$	$i-1$	i			
1			$i-2$	$i-1$	i				n
		$i-2$	$i-1$	i		$\ddots$		n	1
	$i-2$	$i-1$	i				n	1	
$i-2$	$i-1$	i				n	1	$\ddots$	
$i-1$	i				n	1			$i-2$

Fig.3 (b) the general form of a 132, 213, or 321 avoiding Latin Square

iii) With the immense processing power of GPU, its order of magnitude can be faster than CPU for numerically intensive algorithms that are designed to fully exploit the parallelism available.
iv) All patterns of length three are also equivalent for Latin squares and that the growth rate of the number of these Latin Squares is polynomial as opposed to exponential.
v) A pre-processing is there for computing only all valid permutations for each of the minigrids based on the clues in a given Sudoku puzzle [11].

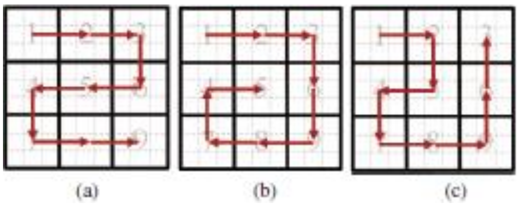

Fig.2 Latin Approaches (a) Zig-Zag way (b) spiral way (c) semi-spiral

Apart from this in order to generate random latin square Ln of order n, these steps are to be followed that is build a undirected graph $G_n=(V_n, E_n)$ .Then generate the largest cliques of G_n, it randomly extract one of the largest clique and its vertices is in lexicographical order, [18]represented as:

$$L_n^{(1)} = I_n + 2\phi(\delta^{(2)}) + \ldots + n\phi(\delta^{(n)})$$

The algorthim is defined for the uniform random sampling of latin square based on largest cliques of proper graphs[18].

III. Conclusion And Proposed Work

In this paper we have listed the methodologies to develop a Latin square computed on CPU. These methodologies have high processing time. Different fields are enclosed in this paper which helps us to use Latin square very frequently in different domains. Thus a framework to develop an efficient algorithmic approach to find a solution to the different Latin Square problems to enhance the existing approaches. Our proposed work is to implementing the Latin square problem definition in GPGPU (GP^2U), which providing heterogeneous environment to execute and capability to reduce time complexity which improves the performance of Latin square computation over intensive domains.

References

[1] C.Colbourn(1984) The Complexity of completing partial Latin squares. Discrete Applied Mathematics8: 25- 30. Doi:10.1016/0166-218X(84)90075-1
[2] Denes, J. and A. Keedwell. 1991. Latin Squares: New Developments in the Theory and applications. North-Holland.

[3] Jacobson, M.T and P. Matthews (1996) Generating uniformly distributed random Latin squares. Journals of Combinatorial Designs 4(6), 405-406
[4] Demetres Christofides, Klas Markstrom (2003) Random Latin square graphs
[5] Brendan D. McKay and Ian M. Wanless(2000) On the number of Latin squares Australian National University, Canberra, ACT 0200, Australia
[6] J.A. Bate, G.H.J. van Rees, The Size of the Smallest Strong Critical Set in a Latin Square University of Manitoba, Winnipeg, Manitoba.
[7] Brendan D. McKay · Jeanette C. McLeod , Ian M. Wanless (2006) The number of transversals in a Latin square, Des Codes Crypt (2006) 40:269–284, DOI 10.1007/s10623-006-0012-8
[8] Yates, F. (1933) the formation of Latin squares for use in field experiments. Empire Journal of Experimental agriculture1(3), 235-244
[9] Dénes, J.; Keedwell, A. D. (1974). Latin squares and their applications. New York-London: Academic Press. p. 547. ISBN.0-12-209350MR 351850
[10] Korani E, Bashiri M., "A model to create orthogonal Graeco Latin square experimental des" , ICMS2009, Istanbul, 2009.
[11] Arnab Kumar Maji and Rajat Kumar Pal,Sudoku Solver using Minigrid based backtracking, 2014
[12] S. E. Bammel and J. Rothstein, The number of 9 × 9 Latin squares, Discrete Math., 11 (1975) 93–95.
[13] Cayley, On Latin squares, Oxford Camb. Dublin Messenger of Math., 19 (1890) 135–137.
[14] Lichun Bao, MALS: Multiple Access Scheduling based on Latin Squares, Donald Bren School of Information and Computer Sciences, University of California, Imine, CA 92697.
[15] D.Selvi G.Velammal and Thevasahayam Arockiadoss, Modified Method of Generating Randomized Latin Squares, IOSR Journal of Computer Engineering (IOSR-JCE)
[16] Michael J. Earnestand Samuel C. Gutekunst, Permutation Patterns in Latin Squares, 2014
[17] Brendan D. McKay and Ian M. Wanless(2000) On the number of Latin squares Australian National University, Canberra, ACT 0200, Australia
[18] Roberto Fontana, Random Latin squares and Sudoku designs generation,2013
[19] Dahl, G. (2009). Permutation matrices related to sudoku. Linear Algebra and its Applications 430(8-9), 2457 – 2463.