Sudoku, Parallel Computing and Stochastic Programming

Peter Z. G. Qian

Department of Statistics
University of Wisconsin-Madison
E-mail: peterq@stat.wisc.edu

Collaborators

- Ben Haaland, Assistant Professor, National University of Singapore.
- Jiajie Chen and Xu Xu at the University of Wisconsin-Madison.

Sudoku

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

- Rule: Fill in the missing entries subject to each row, column and three-by-three subsquare containing no duplicated numbers.
- Invented by Howard Garns in the 1970's.
- Became widespread in Japan during the 1980's.
- Now is played by millions of people across the globe.

The Completed Sudoku

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

- For each row, each number of 1, . . ., 9 appears precisely once.
- For each row, each number of 1, . . ., 9 appears precisely once.
- For each three-by-three subsquare, each number of 1, . . ., 9 appears precisely once.

An Ordinary Latin Square vs. a Sudoku Latin Square

1	2	3	4	5	6	7	8	9
2	3	4	5	6	7	8	9	1
3	4	5	6	7	8	9	1	2
4	5	6	7	8	9	1	2	3
5	6	7	8	9	1	2	3	4
6	7	8	9	1	2	3	4	5
7	8	9	1	2	3	4	5	6
8	9	1	2	3	4	5	6	7
9	1	2	3	4	5	6	7	8

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

- Sudoku Latin squares are available for general $s_1 \times s_1$, not only 9×9.
- Three types of uniformity of a Sudoku Latin square give what?

Three Doubly Orthogonal Sudoku Latin Squares

$GF(9) = \{0, 1, 2, x, x+1, x+2, 2x, 2x+1, 2x+2\}$ with $p(x) = x^2 + x + 2$.

$$L_1 = \left(\begin{array}{ccc|ccc|ccc}
0 & 1 & 2 & x & x+1 & x+2 & 2x & 2x+1 & 2x+2 \\
x+1 & x+2 & x & 2x+1 & 2x+2 & 2x & 1 & 2 & 0 \\
2x+2 & 2x & 2x+1 & 2 & 0 & 1 & x+2 & x & x+1 \\
\hline
1 & 2 & 0 & x+1 & x+2 & x & 2x+1 & 2x+2 & 2x \\
x+2 & x & x+1 & 2x+2 & 2x & 2x+1 & 2 & 0 & 1 \\
2x & 2x+1 & 2x+2 & 0 & 1 & 2 & x & x+1 & x+2 \\
\hline
2 & 0 & 1 & x+2 & x & x+1 & 2x+2 & 2x & 2x+1 \\
x & x+1 & x+2 & 2x & 2x+1 & 2x+2 & 0 & 1 & 2 \\
2x+1 & 2x+2 & 2x & 1 & 2 & 0 & x+1 & x+2 & x
\end{array}\right),$$

$L_1^{(1,1)}$: the subsquare in the top-left corner.

$$L_2 = \left(\begin{array}{ccc|ccc|ccc}
0 & 1 & 2 & x & x+1 & x+2 & 2x & 2x+1 & 2x+2 \\
x & x+1 & x+2 & 2x & 2x+1 & 2x+2 & 0 & 1 & 2 \\
2x & 2x+1 & 2x+2 & 0 & 1 & 2 & x & x+1 & x+2 \\
\hline
2x+1 & 2x+2 & 2x & 1 & 2 & 0 & x+1 & x+2 & x \\
1 & 2 & 0 & x+1 & x+2 & x & 2x+1 & 2x+2 & 2x \\
x+1 & x+2 & x & 2x+1 & 2x+2 & 2x & 1 & 2 & 0 \\
\hline
x+2 & x & x+1 & 2x+2 & 2x & 2x+1 & 2 & 0 & 1 \\
2x+2 & 2x & 2x+1 & 2 & 0 & 1 & x+2 & x & x+1 \\
2 & 0 & 1 & x+2 & x & x+1 & 2x+2 & 2x & 2x+1
\end{array}\right),$$

$L_2^{(1,1)}$: the subsquare in the top-left corner.

$$L_3 = \left(\begin{array}{ccc|ccc|ccc}
0 & 1 & 2 & x & x+1 & x+2 & 2x & 2x+1 & 2x+2 \\
x+2 & x & x+1 & 2x+2 & 2x & 2x+1 & 2 & 0 & 1 \\
2x+1 & 2x+2 & 2x & 1 & 2 & 0 & x+1 & x+2 & x \\
\hline
x+1 & x+2 & x & 2x+1 & 2x+2 & 2x & 1 & 2 & 0 \\
2x & 2x+1 & 2x+2 & 0 & 1 & 2 & x & x+1 & x+2 \\
2 & 0 & 1 & x+2 & x & x+1 & 2x+2 & 2x & 2x+1 \\
\hline
2x+2 & 2x & 2x+1 & 2 & 0 & 1 & x+2 & x & x+1 \\
1 & 2 & 0 & x+1 & x+2 & x & 2x+1 & 2x+2 & 2x \\
x & x+1 & x+2 & 2x & 2x+1 & 2x+2 & 0 & 1 & 2
\end{array}\right).$$

$L_3^{(1,1)}$: the subsquare in the top-left corner.

Structures of L_1, L_2 and L_3

L_1, L_2 and L_3 are orthogonal: for any two of them, all 81 level combinations appear precisely once.

$L_1^{(1,1)}$, $L_2^{(1,1)}$ and $L_3^{(1,1)}$ are orthogonal after $\{0, x+1, 2x+2\} \to 0$, $\{1, x+2, 2x\} \to 1$, $\{2, x, 2x+1\} \to 2$.

0	0	0
$x+1$	x	$x+2$
$2x+2$	$2x$	$2x+1$
1	1	1
$x+2$	$x+1$	x
$2x$	$2x+1$	$2x+2$
2	2	2
x	$x+2$	$x+1$
$2x+1$	$2x+2$	$2x$

$\to$

0	0	0
0	2	1
0	1	2
1	1	1
1	0	2
1	2	0
2	2	2
2	1	0
2	0	1

The same story for $L_1^{(i,j)}$, $L_2^{(i,j)}$ and $L_3^{(i,j)}$, $i, j = 1, 2, 3$.

Math: Subfield Projection

- Let $s_1 = p^{u_1}$ and $s_2 = p^{u_2}$ be powers of the same prime p, where $u_1 > u_2 \geq 1$ and $u_2 | u_1$.
- Let F denote $GF(s_1)$ with a primitive element α, where $\alpha = x$. Obtain $G = \{0, \beta, \ldots, \beta^{s_2-1}\}$ as a subfield of F, where $\beta = \alpha^{(s_1-1)/(s_2-1)}$.
- Let $\lambda = u_1/u_2$. Represent any $f(x) \in F$ as

$$f(x) = b_0 + b_1\alpha + \cdots + b_{\lambda-1}\alpha^{\lambda-1},\ b_i \in G\ \ (i = 0, \ldots, \lambda-1).$$

- For $f(x) \in F$, define

$$\phi\{f(x)\} = b_0 + b_1\beta + \cdots + b_{\lambda-1}\beta^{\lambda-1},\ b_i \in G\ \ (i = 0, \ldots, \lambda-1).$$

- For $GF(9)$ to $GF(3) = \{0, 1, 2\}$ mod 3, the subfield projection ϕ is: $\{0, x+1, 2x+2\} \to 0$, $\{1, x+2, 2x\} \to 1$, $\{2, x, 2x+1\} \to 2$.

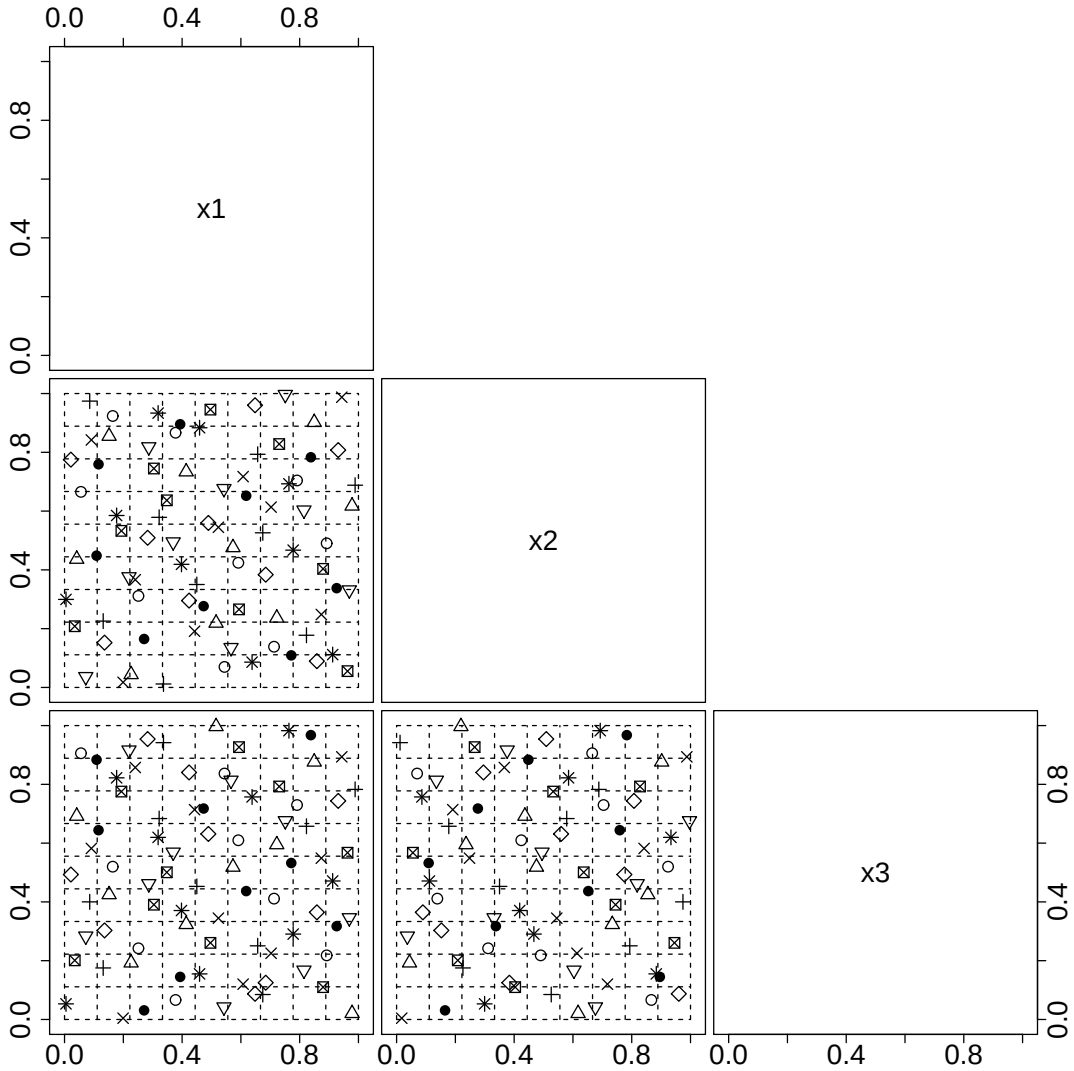
Qian, P. Z. G. and Wu, C. F. J. (2009), "Sliced Space-Filling Designs," *Biometrika*, **96**, 945-956.

Building Block: Doubly Orthogonal Sudoku Latin Squares

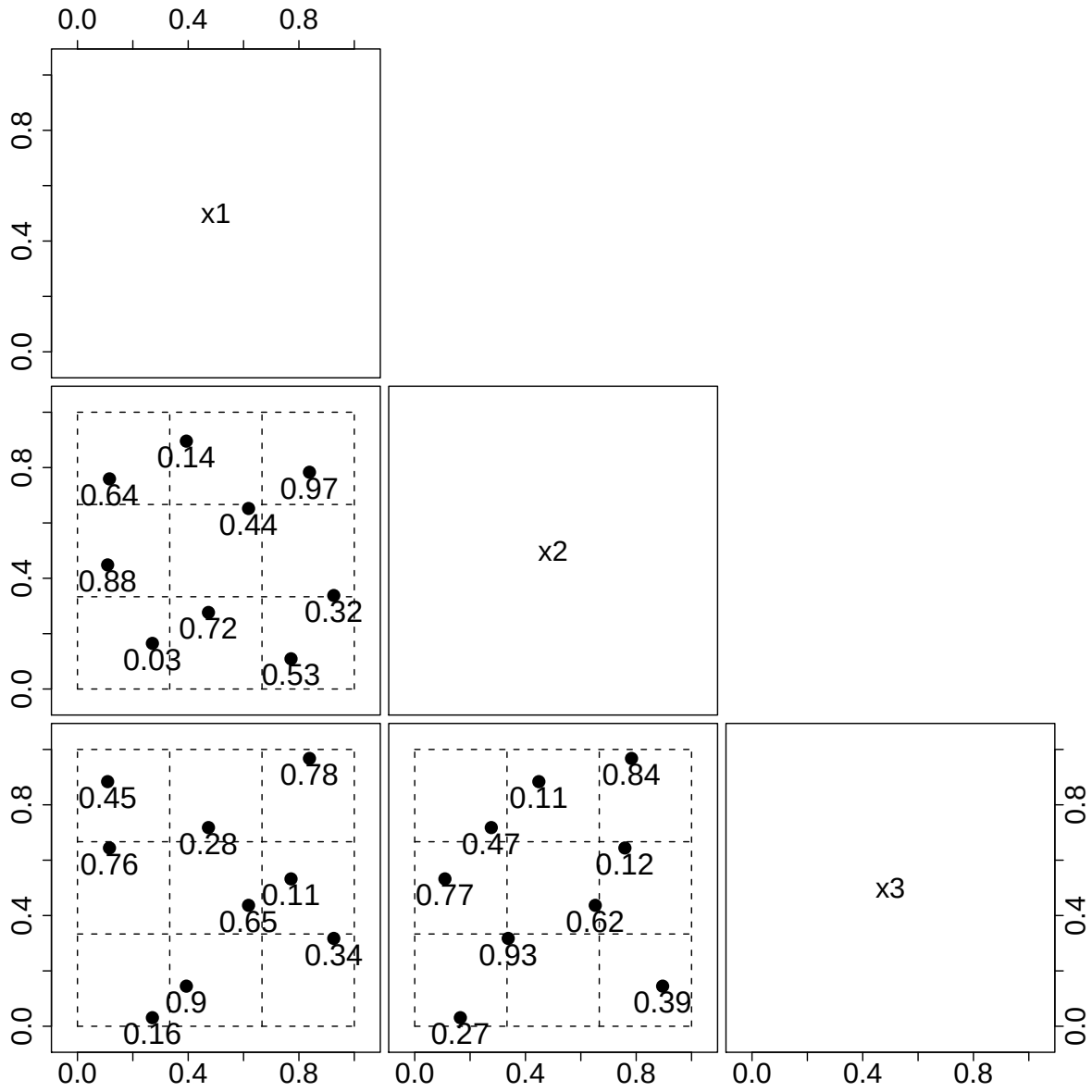
- Let $s_1 = p^{2u}$ and $s_2 = p^u$ be powers of the same prime p for an integer $u \geq 1$. A set of Sudoku Latin squares of order s_1 are *doubly orthogonal* if (i) the complete squares are *orthogonal* and (ii) for any given corner, all subsquares of order s_2 are *orthogonal* after some suitable level-collapsing.
- A *Sudoku based space-filling design* is a space-filling design that can be divided into slices, such that the complete design and each slice achieve maximum stratification in both one- and two- dimensional projections.

Xu, X., Haaland, B. and Qian, P. Z. G. (2011), "Sudoku-Based Space-Filling Designs," *Biometrika*, **98**, 711-720.

A Sudoku Based Space-Filling Design from L_1, L_2, L_3



A Slice of Nine Points



Application: Data Pooling

- Data pooling: combining data from multiple sources (e.g., labs, groups, ...) for studying the same complex phenomenon.
- Use a Sudoku based space-filling design with t slices for pooling data from t different sources, with one slice $\rightarrow$ each source.
- The subdesign for each source is space-filling.
- The collapsed design across all the sources is space-filling, thus enabling efficient data pooling when the source-to-source difference is small.

Variations of Sudoku

- Samurai Sudoku: five 9×9 grids that overlap at the corner region.
- Sequential grids.
- Kakuro and Futoshiki.

The Completed Samurai Sudoku

2	3	4	5	6	7	8	9	1				5	6	7	8	9	1	2	3	4
5	6	7	8	9	1	2	3	4				8	9	1	2	3	4	5	6	7
8	9	1	2	3	4	5	6	7				2	3	4	5	6	7	8	9	1
3	4	5	6	7	8	9	1	2				6	7	8	9	1	2	3	4	5
6	7	8	9	1	2	3	4	5				9	1	2	3	4	5	6	7	8
9	1	2	3	4	5	6	7	8				3	4	5	6	7	8	9	1	2
4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6
7	8	9	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	1	2	3
						2	3	4	5	6	7	8	9	1						
						5	6	7	8	9	1	2	3	4						
						8	9	1	2	3	4	5	6	7						
6	7	8	9	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8
9	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	1	2
3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	1	2	3	4	5
7	8	9	1	2	3	4	5	6				1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9				4	5	6	7	8	9	1	2	3
4	5	6	7	8	9	1	2	3				7	8	9	1	2	3	4	5	6
8	9	1	2	3	4	5	6	7				2	3	4	5	6	7	8	9	1
2	3	4	5	6	7	8	9	1				5	6	7	8	9	1	2	3	4
5	6	7	8	9	1	2	3	4				8	9	1	2	3	4	5	6	7

Construction of Samurai Sudoku Squares: Step 1

Generate a Sudoku Latin square

$$L = \left(\begin{array}{ccc|ccc|ccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
4 & 5 & 6 & 7 & 8 & 9 & 1 & 2 & 3 \\
7 & 8 & 9 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 1 \\
5 & 6 & 7 & 8 & 9 & 1 & 2 & 3 & 4 \\
8 & 9 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
3 & 4 & 5 & 6 & 7 & 8 & 9 & 1 & 2 \\
6 & 7 & 8 & 9 & 1 & 2 & 3 & 4 & 5 \\
9 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8
\end{array}\right)$$

Construction of Samurai Sudoku Squares: Step 2

Using $\{1,2,3,4,5,6,7,8,9\} \to \{2,3,4,5,6,7,8,9,1\}$, obtain $L^{(1)}$ from L.

$$L = \left(\begin{array}{ccc|ccc|ccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
4 & 5 & 6 & 7 & 8 & 9 & 1 & 2 & 3 \\
7 & 8 & 9 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 1 \\
5 & 6 & 7 & 8 & 9 & 1 & 2 & 3 & 4 \\
8 & 9 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
3 & 4 & 5 & 6 & 7 & 8 & 9 & 1 & 2 \\
6 & 7 & 8 & 9 & 1 & 2 & 3 & 4 & 5 \\
9 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8
\end{array}\right) \to L^{(1)} = \left(\begin{array}{ccc|ccc|ccc}
2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 1 \\
5 & 6 & 7 & 8 & 9 & 1 & 2 & 3 & 4 \\
8 & 9 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
3 & 4 & 5 & 6 & 7 & 8 & 9 & 1 & 2 \\
6 & 7 & 8 & 9 & 1 & 2 & 3 & 4 & 5 \\
9 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\hline
4 & 5 & 6 & 7 & 8 & 9 & 1 & 2 & 3 \\
7 & 8 & 9 & 1 & 2 & 3 & 4 & 5 & 6 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9
\end{array}\right)$$

The upper left corner of L is the same as the bottom right corner of $L^{(1)}$.

Construction of Samurai Sudoku Squares: Step 2

Using $\{1,2,3,4,5,6,7,8,9\} \to \{5,6,7,8,9,1,2,3,4\}$, obtain $L^{(2)}$ from L.

$$L = \left(\begin{array}{ccc|ccc|ccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
4 & 5 & 6 & 7 & 8 & 9 & 1 & 2 & 3 \\
7 & 8 & 9 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 1 \\
5 & 6 & 7 & 8 & 9 & 1 & 2 & 3 & 4 \\
8 & 9 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
3 & 4 & 5 & 6 & 7 & 8 & 9 & 1 & 2 \\
6 & 7 & 8 & 9 & 1 & 2 & 3 & 4 & 5 \\
9 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8
\end{array}\right) \to L^{(2)} = \left(\begin{array}{ccc|ccc|ccc}
5 & 6 & 7 & 8 & 9 & 1 & 2 & 3 & 4 \\
8 & 9 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 1 \\
\hline
6 & 7 & 8 & 9 & 1 & 2 & 3 & 4 & 5 \\
9 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
3 & 4 & 5 & 6 & 7 & 8 & 9 & 1 & 2 \\
\hline
7 & 8 & 9 & 1 & 2 & 3 & 4 & 5 & 6 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
4 & 5 & 6 & 7 & 8 & 9 & 1 & 2 & 3
\end{array}\right)$$

The upper right corner of L is the same as the bottom left corner of $L^{(2)}$.

Construction of Samurai Sudoku Squares: Step 2

Using $\{1,2,3,4,5,6,7,8,9\} \to \{6,7,8,9,1,2,3,4,5\}$, obtain $L^{(3)}$ from L.

$$L = \left(\begin{array}{ccc|ccc|ccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
4 & 5 & 6 & 7 & 8 & 9 & 1 & 2 & 3 \\
7 & 8 & 9 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 1 \\
5 & 6 & 7 & 8 & 9 & 1 & 2 & 3 & 4 \\
8 & 9 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
3 & 4 & 5 & 6 & 7 & 8 & 9 & 1 & 2 \\
6 & 7 & 8 & 9 & 1 & 2 & 3 & 4 & 5 \\
9 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8
\end{array}\right) \to L^{(3)} = \left(\begin{array}{ccc|ccc|ccc}
6 & 7 & 8 & 9 & 1 & 2 & 3 & 4 & 5 \\
9 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
3 & 4 & 5 & 6 & 7 & 8 & 9 & 1 & 2 \\
\hline
7 & 8 & 9 & 1 & 2 & 3 & 4 & 5 & 6 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
4 & 5 & 6 & 7 & 8 & 9 & 1 & 2 & 3 \\
\hline
8 & 9 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 1 \\
5 & 6 & 7 & 8 & 9 & 1 & 2 & 3 & 4
\end{array}\right)$$

The bottom left corner of L is the same as the top right corner of $L^{(3)}$.

Construction of Samurai Sudoku Squares: Step 2

Using $\{1,2,3,4,5,6,7,8,9\} \to \{9,1,2,3,4,5,6,7,8\}$, obtain $L^{(4)}$ from L.

$$L = \left(\begin{array}{ccc|ccc|ccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
4 & 5 & 6 & 7 & 8 & 9 & 1 & 2 & 3 \\
7 & 8 & 9 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 1 \\
5 & 6 & 7 & 8 & 9 & 1 & 2 & 3 & 4 \\
8 & 9 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
3 & 4 & 5 & 6 & 7 & 8 & 9 & 1 & 2 \\
6 & 7 & 8 & 9 & 1 & 2 & 3 & 4 & 5 \\
9 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8
\end{array}\right) \to L^{(4)} = \left(\begin{array}{ccc|ccc|ccc}
9 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
3 & 4 & 5 & 6 & 7 & 8 & 9 & 1 & 2 \\
6 & 7 & 8 & 9 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
4 & 5 & 6 & 7 & 8 & 9 & 1 & 2 & 3 \\
7 & 8 & 9 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 1 \\
5 & 6 & 7 & 8 & 9 & 1 & 2 & 3 & 4 \\
8 & 9 & 1 & 2 & 3 & 4 & 5 & 6 & 7
\end{array}\right)$$

The bottom right corner of L is the same as the top left corner of $L^{(4)}$.

Construction of Samurai Sudoku Squares: Step 3

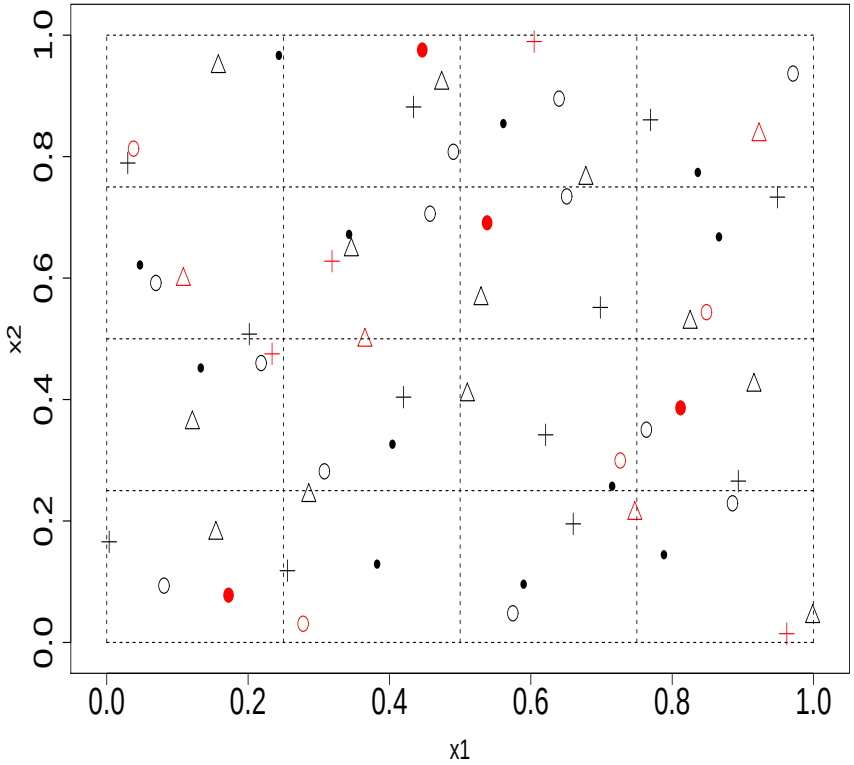
Put L, $L^{(1)}$, $L^{(2)}$, $L^{(3)}$ and $L^{(4)}$ together with L at the center.

2	3	4	5	6	7	8	9	1	$\leftarrow$	$L^{(1)}$		5	6	7	8	9	1	2	3	4
5	6	7	8	9	1	2	3	4				8	9	1	2	3	4	5	6	7
8	9	1	2	3	4	5	6	7		$L^{(2)}$	$\rightarrow$	2	3	4	5	6	7	8	9	1
3	4	5	6	7	8	9	1	2				6	7	8	9	1	2	3	4	5
6	7	8	9	1	2	3	4	5				9	1	2	3	4	5	6	7	8
9	1	2	3	4	5	6	7	8				3	4	5	6	7	8	9	1	2
4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6
7	8	9	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	1	2	3
						2	3	4	5	6	7	8	9	1						
						5	6	7	8	9	1	2	3	4	$\leftarrow$	L				
						8	9	1	2	3	4	5	6	7						
6	7	8	9	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8
9	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	1	2
3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	1	2	3	4	5
7	8	9	1	2	3	4	5	6				1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9				4	5	6	7	8	9	1	2	3
4	5	6	7	8	9	1	2	3	$\leftarrow$	$L^{(3)}$		7	8	9	1	2	3	4	5	6
8	9	1	2	3	4	5	6	7				2	3	4	5	6	7	8	9	1
2	3	4	5	6	7	8	9	1		$L^{(4)}$	$\rightarrow$	5	6	7	8	9	1	2	3	4
5	6	7	8	9	1	2	3	4				8	9	1	2	3	4	5	6	7

Two Doubly Orthogonal Samurai Sudoku Sqaures

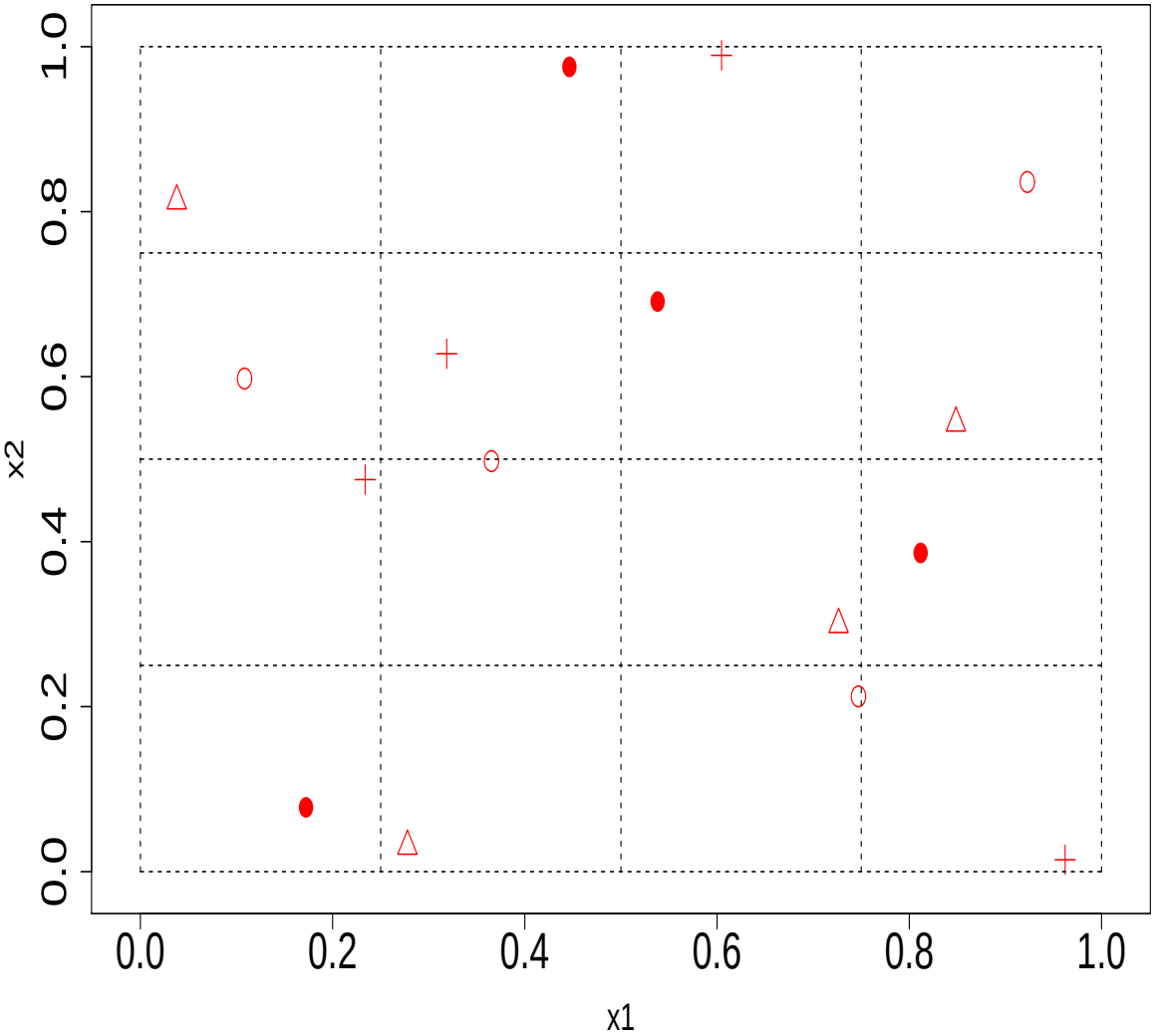
$$\left(\begin{array}{cccc|cccc}
4 & 2 & 3 & 1 & 4 & 2 & 3 & 1 \\
3 & 1 & 4 & 2 & 3 & 1 & 4 & 2 \\
2 & 4 & 1 & 3 & 2 & 4 & 1 & 3 \\
1 & 3 & 2 & 4 & 1 & 3 & 2 & 4 \\
\hline
4 & 2 & 3 & 1 & 4 & 2 & 3 & 1 \\
3 & 1 & 4 & 2 & 3 & 1 & 4 & 2 \\
2 & 4 & 1 & 3 & 2 & 4 & 1 & 3 \\
1 & 3 & 2 & 4 & 1 & 3 & 2 & 4
\end{array}\right)
\quad
\left(\begin{array}{cccc|cccc}
1 & 3 & 2 & 4 & 1 & 3 & 2 & 4 \\
4 & 2 & 3 & 1 & 4 & 2 & 3 & 1 \\
2 & 4 & 1 & 3 & 2 & 4 & 1 & 3 \\
3 & 1 & 4 & 2 & 3 & 1 & 4 & 2 \\
\hline
1 & 3 & 2 & 4 & 1 & 3 & 2 & 4 \\
4 & 2 & 3 & 1 & 4 & 2 & 3 & 1 \\
2 & 4 & 1 & 3 & 2 & 4 & 1 & 3 \\
3 & 1 & 4 & 2 & 3 & 1 & 4 & 2
\end{array}\right)$$

A Samurai Sudoku Based Space-Filling Design

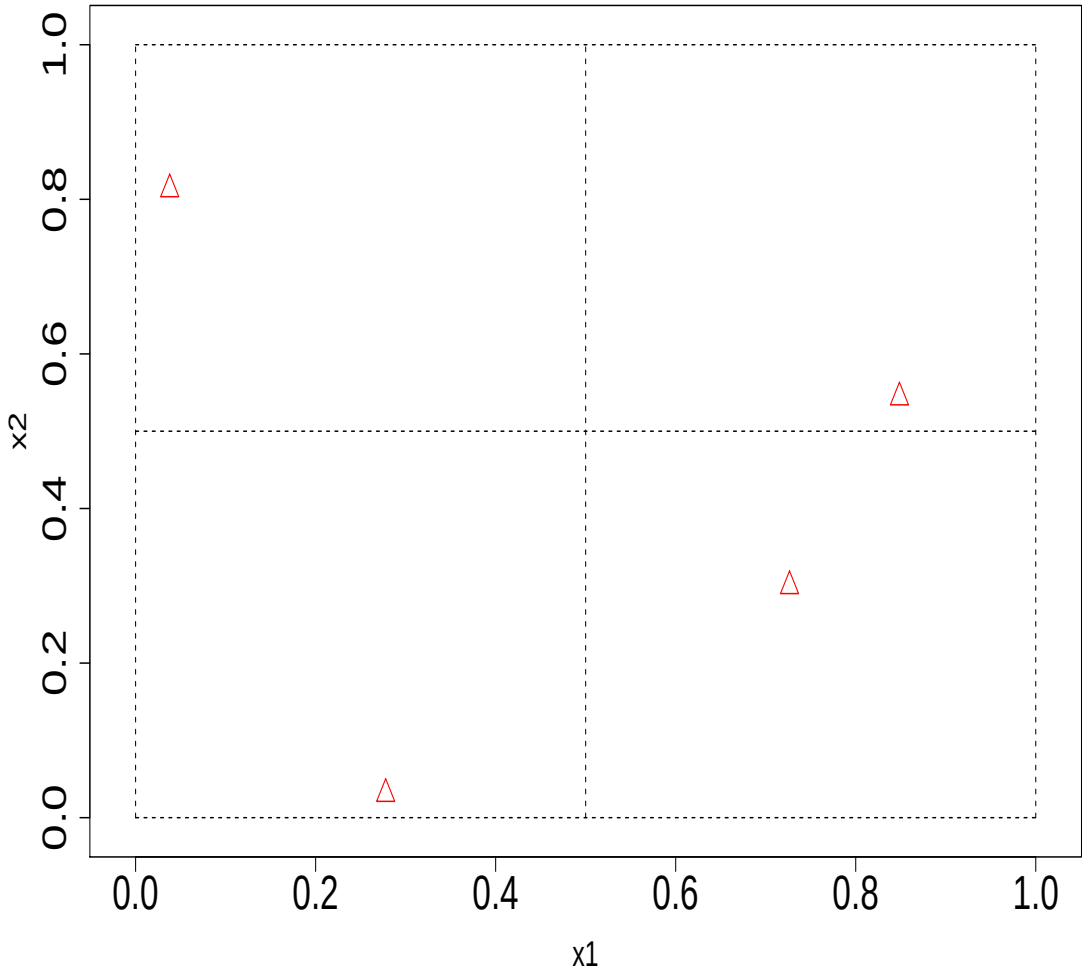


- Five overlapping slices (●, ○, +, △ and red).
- Each slice is a Sudoku-based space-filling design with 16 points.
- The connecting slice can be divided into four subslices (●, ○, +, △).

The Connecting Slice of 16 Points



One Subslice of the Connecting Slice



Complex System Simulations

- **Computer experiments**, i.e., experiments via simulation codes, are becoming increasingly important for studying complex phenomena.
- Applications: defense, nano, energy, biology, security, ...
- Examples:
  - *Design Space Exploration* for running finite element analysis and computational fluid dynamics codes in engine design at Boeing.
  - Agent-based simulations in IBM's *A Smarter Planet Project*.
  - Multi-physics models for security applications at Los Alamos National Laboratory.

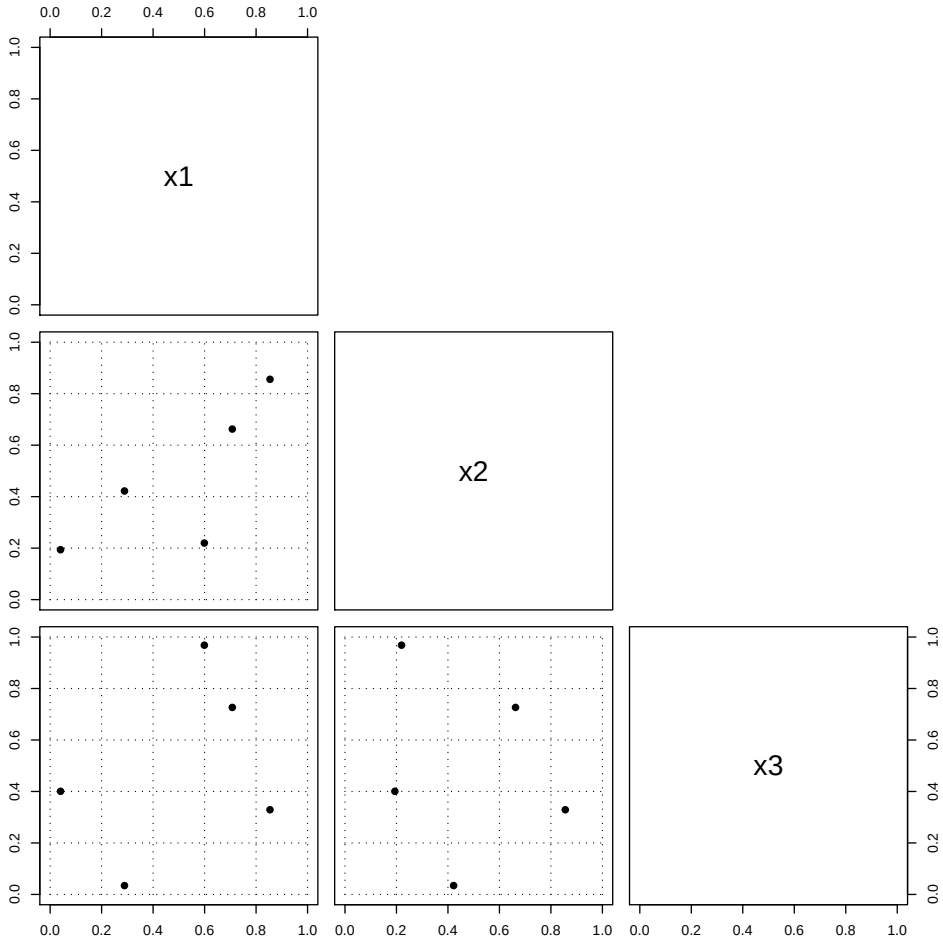
Statistical Designs for Parallel Computing

- Consider a blackbox function $f(\mathbf{x})$, where $\mathbf{x} = (x_1, \ldots, x_q)'$ has the uniform distribution F on $(0,1]^q$.
- Let $\mu = \int f dF$ and $\sigma^2 = \mathrm{var}[f(\mathbf{x})]$.
- Aim: Estimate μ using Machines $1, \ldots, t$, where the set of input values D_i for Machine i has m runs.
- For $i = 1, \ldots, t$, define

$$\hat{\mu}_i = \frac{\sum_{j=1}^{m} f(\mathbf{x}_{ij})}{m}, \tag{1}$$

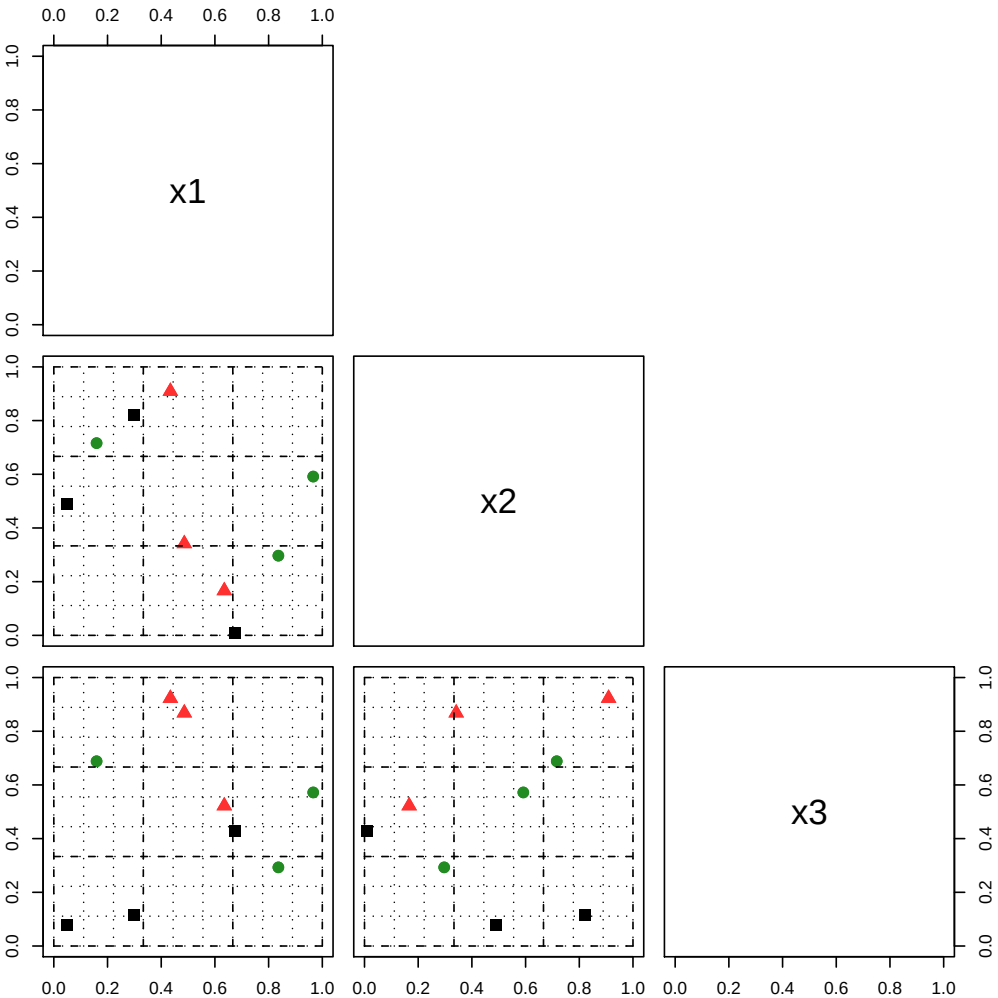
  where $\mathbf{x}_{ij}$ is the jth run of D_i.
- Let $\hat{\mu} = t^{-1} \sum_{i=1}^{t} \hat{\mu}_i$.

An Ordinary Latin Hypercube Design of 5 Runs

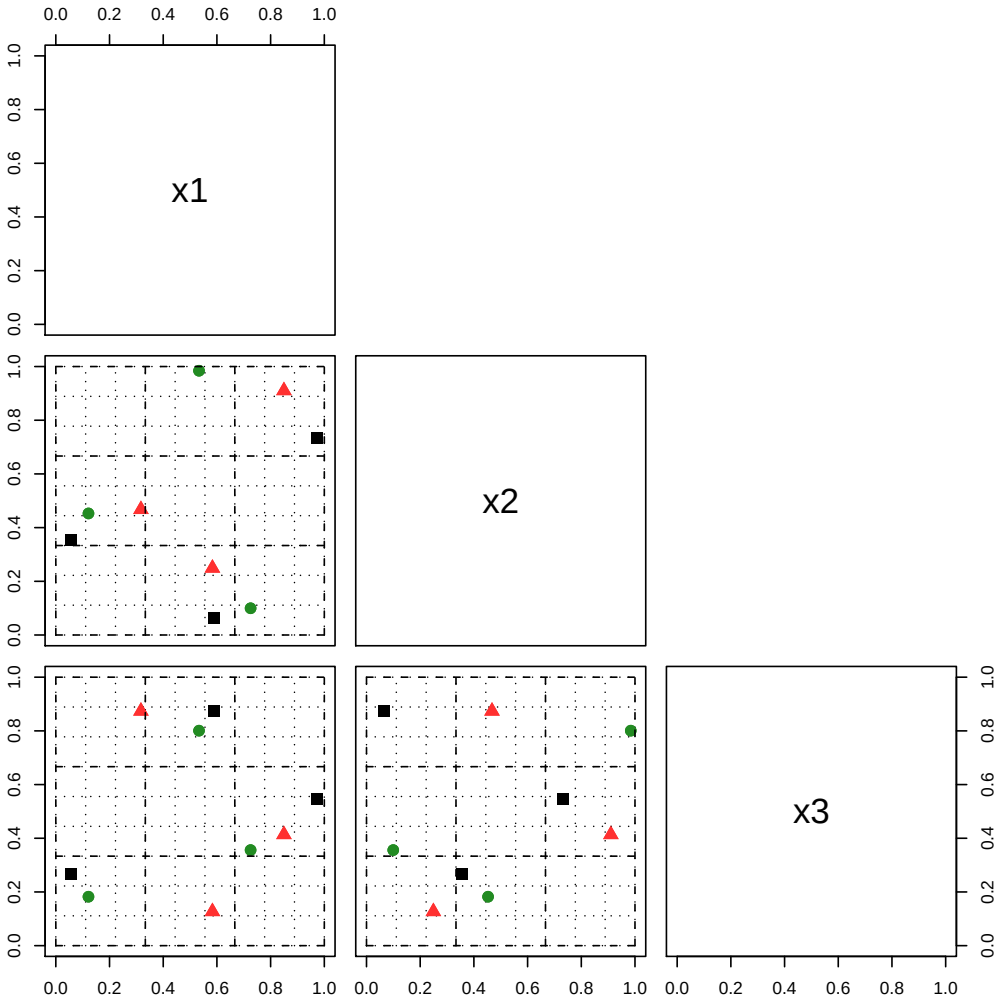


- Achieves maximum marginal uniformity and variance reduction (Mckay et al., 1979; Stein, 1987).

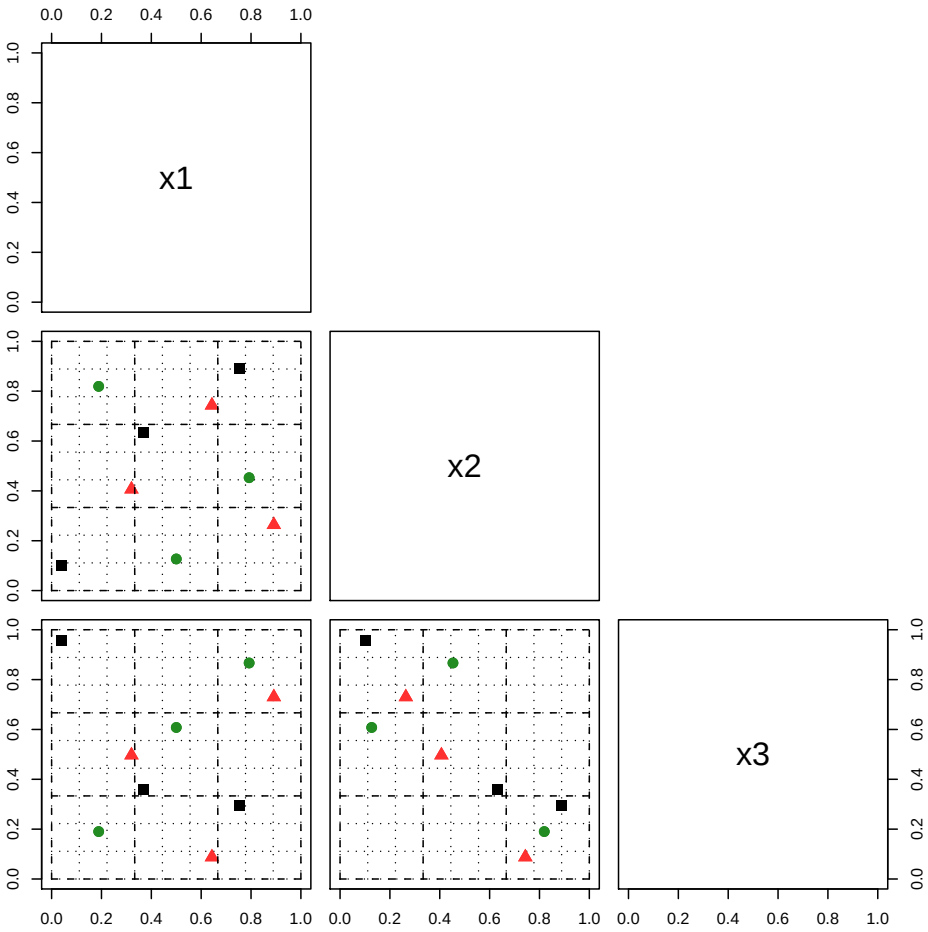
Dividing an Ordinary Latin Hypercube Design into 3 Slices



Combining Three Ordinary Latin Hypercube Designs of 3 Runs



A Sliced Latin Hypercube Design with 3 Slices



Qian, P. Z. G. (2011), "Sliced Latin Hypercube Designs," *Journal of the American Statistcal Association*, in press.

Illustration

- Estimate the mean of

$$f(\mathbf{x}) = \log\left(\frac{1}{\sqrt{x_1}} + \frac{1}{\sqrt{x_2}}\right)$$

  using Machines $1,\ldots,4$, where x_1 and x_2 are $U[0,1)$ random variables.

- For $i = 1,\ldots,4$, the set of input values D_i for Machine i has m runs, and let $\hat{\mu}_i$ be the estimator of μ using D_i. Define $\hat{\mu} = \frac{1}{4}(\hat{\mu}_1 + \hat{\mu}_2 + \hat{\mu}_3 + \hat{\mu}_4)$.

- Compare four methods to generate $D_1,\ldots,D_4$.

  IID: Generate four independent IID samples of m runs each.

  SPLIT: Divide an ordinary Latin hypercube design of $4m$ runs into four slices of m runs each.

  COMBINE: Combine four ordinary Latin hypercube designs of m runs each.

  SLHD: Use an SLHD with four slices of m runs each.

Results

Each method is replicated 2000 times. The accompanying table presents the RMSE's of $\hat{\mu}_1$ and $\hat{\mu}$ over the 2000 replicates.

	m	IID	SPLIT	COMBINE	SLHD
$\hat{\mu}_1$	5	0.1941	0.1800	**0.1136**	**0.1093**
	10	0.1378	0.1234	**0.0600**	**0.0600**
	20	0.0750	0.0865	**0.0326**	**0.0332**
	40	0.0699	0.0614	**0.0183**	**0.0180**
$\hat{\mu}$	5	0.0952	**0.0321**	0.0559	**0.0329**
	10	0.0692	**0.0183**	0.0302	**0.0180**
	20	0.0499	**0.0101**	0.0174	**0.0104**
	40	0.0344	**0.0061**	0.0090	**0.0060**

Sampling Properties of SLHD's

- Estimate $\mu = E[f(\mathbf{x})]$, where the distribution of $\mathbf{x}$ is the uniform measure on $[0,1)^q$.
- Let $\hat{\mu}$ be based on an SLHD with slices $D_1, \ldots, D_t$ of m runs, where $\hat{\mu}_j$ is associated with D_j.
- As $m \to \infty$ with t fixed,

$$\mathrm{var}(\hat{\mu}_j) = m^{-1}\sigma^2 - m^{-1}\sum_{k=1}^{q}\int_0^1 [f_{-k}(x_k)]^2 dx_k + o(m^{-1})$$

and

$$\mathrm{var}(\hat{\mu}) = n^{-1}\sigma^2 - n^{-1}\sum_{k=1}^{q}\int_0^1 [f_{-k}(x_k)]^2 dx_k + o(n^{-1}),$$

where $f_{-k}(x_k) = \int\{f(\mathbf{x}) - \mu\}dF_{-k}$, $dF_{-k} = \prod_{j\neq k} dF_j$, and F_k is the uniform distribution $U[0,1)$ in the kth dimension.

Stochastic Programming

A stochastic program:

$$\min_{x \in X} \{ f(x) := \mathbb{E}_P[F(x, \xi)] \} \tag{2}$$

where $F(x, \xi)$ is a real valued function of two vector variables $x \in \mathbb{R}^n$ and $\xi \in \mathbb{R}^d$, X is a subset of $\mathbb{R}^n$, and ξ is a random vector with probability distribution P.

- The solution to the problem in (2) is denoted by x^*.
- The optimal value of the problem in (2) is denoted by v^*.

Books: Kall and Wallace (1995); Birge and Louveaux (1997); Ruszczynski and Shapiro (2004); and Shapiro et al. (2009).

Sample Average Approximations

Step 1. Obtain a sample $\xi^1, \ldots, \xi^N$ from the distribution P and approximate $f(x)$ by $\hat{f}_N(x) = N^{-1} \sum_{i=1}^{N} F(x, \xi^i)$.

Step 2. Solve the approximated problem:

$$\min_{x \in X} \hat{f}_N(x). \tag{3}$$

- The solution to the problem in (3) is denoted by $\hat{x}_N$.
- The optimal value of the problem in (3) is denoted by $\hat{v}_N$.
- IID sampling is the prevailing choice for Step 1. Under this sampling scheme, (1) $\hat{x}_N$ converges to x^* as $N \to \infty$ and (2) large-sample normality of $\hat{f}_N(x)$ and $\hat{x}_N$ (Birge and Louveaux, 1997; Ruszczynski and Shapiro 2004).

Lower Bound Estimates for Optimal Value (Mak et al., 1999)

Note that $\mathbb{E}[\hat{v}_N] \leq v^*$. Estimate $\mathbb{E}[\hat{v}_N]$ as follows.

Step 1. Generate M independent batches of samples, $\xi^{1,j}, \ldots, \xi^{N,j}$, $j = 1, \ldots, M$, each of size N, and solve the corresponding SAA problem

$$\hat{v}_N^j = \min_{x \in X} \{ \hat{f}_N^j(x) := N^{-1} \sum_{i=1}^{N} F(x, \xi^{i,j}) \} \tag{4}$$

for $j = 1, \ldots, M$.

Step 2. Obtain an approximate $(1-\alpha)100\%$ confidence interval for $\mathbb{E}(\hat{v}_N)$ given by

$$[L_{N,M} - \frac{z_{\alpha/2} s_L(M)}{\sqrt{M}}, L_{N,M} + \frac{z_{\alpha/2} s_L(M)}{\sqrt{M}}], \tag{5}$$

where $L_{N,M} = \frac{1}{M} \sum_{j=1}^{M} \hat{v}_N^j$ and $s_L^2(M) = \frac{1}{M-1} \sum_{j=1}^{M} (\hat{v}_N^j - L_{N,M})^2$.

Comparison of Three Designs

IID: M i.i.d. samples of size N each.

LHD: M ordinary Latin hypercube designs (McKay et al., 1979). Used in Drew and Homem-de-Mello (2005), Linderoth et al. (2006) and Homem-de-Mello (2008).

SLHD: A sliced Latin hypercube design with M slices, each of size N.

Example

Consider a stochastic program:

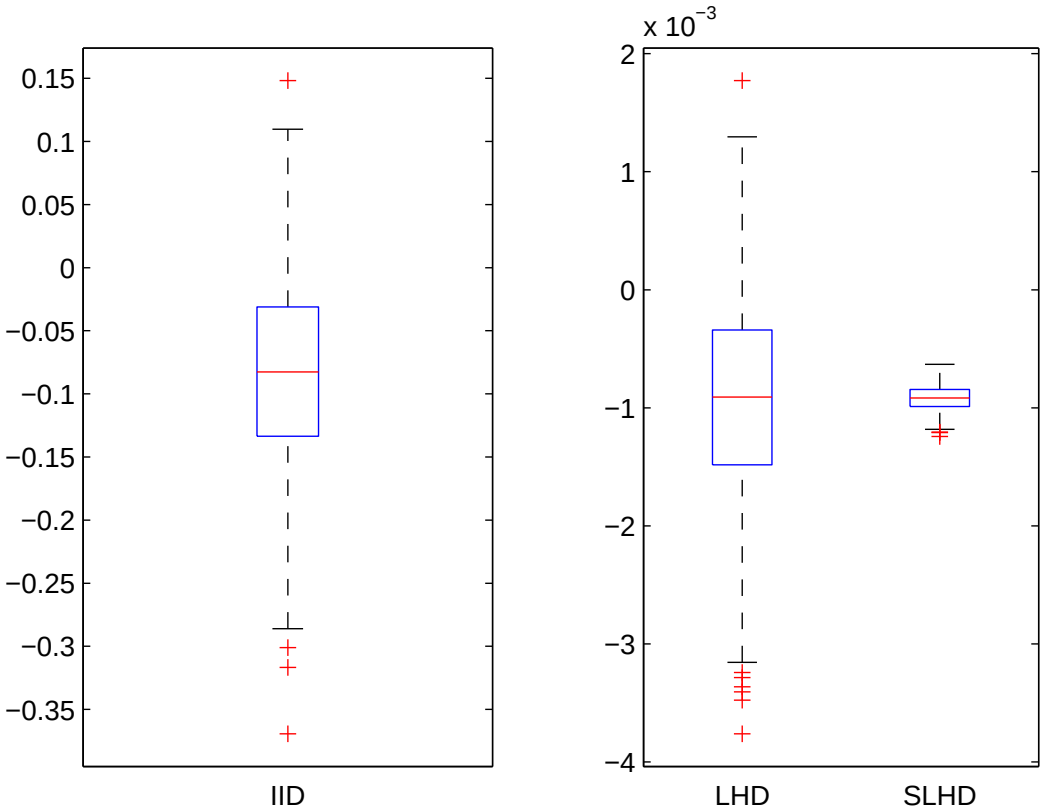
$$\min_x \mathbb{E}[p(\xi)^T x] \quad \text{subject to} \quad Ax \geq b, x \geq 0,$$

where $x = (x_1, x_2)^T \in \mathbb{R}^2$, $\xi = (\xi_1, \xi_2, \xi_3)^T$ is uniformly distributed on $U(0,1]^3$,

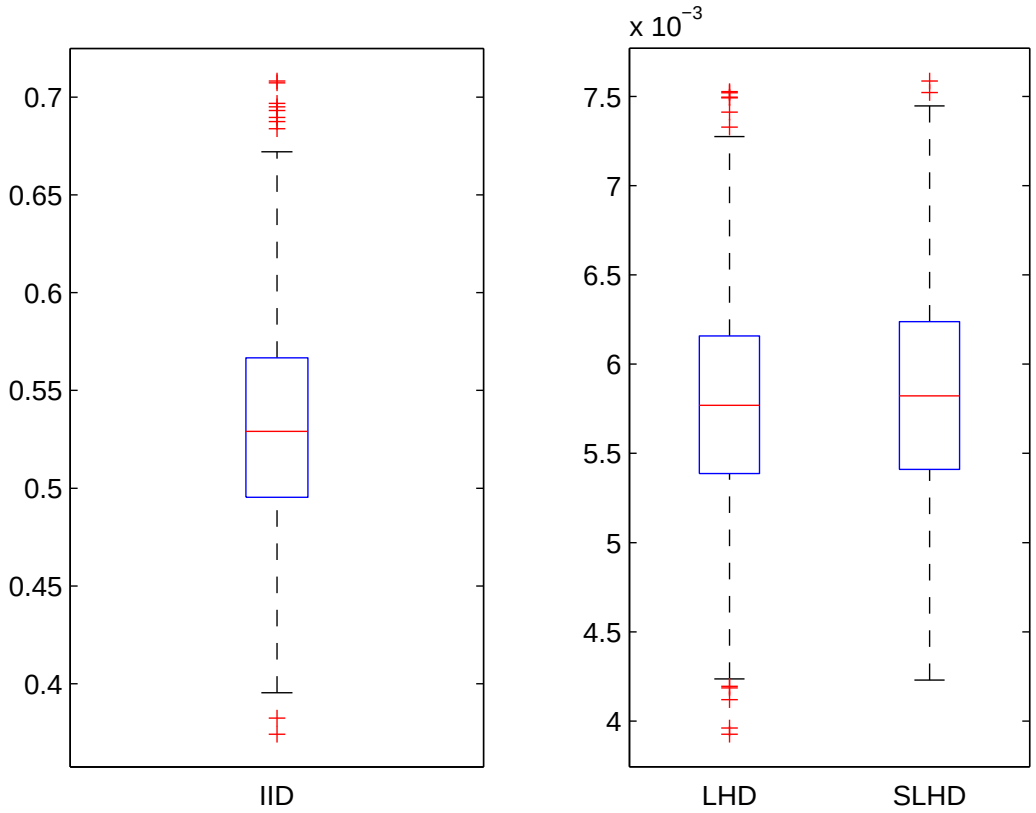
$$p(\xi) = \begin{pmatrix} 6(\xi_1)^2 - 4\xi_2 \\ 4(\xi_3 + 1) \end{pmatrix}, b = \begin{pmatrix} 1 \\ 8 \\ -3 \end{pmatrix}, A = \begin{pmatrix} 1 & 2 \\ 4 & 2 \\ -1 & -1 \end{pmatrix}.$$

The optimal value $v^* = 0$. The set of optimal solutions
$S^* = \{(x_1, x_2) | 2 \leq x_1 \leq 3, x_2 = 0\}$.

Boxplots of $L_{N,M}$ for $N = 100$ and $M = 50$ over 1000 Replicates



Boxplots of $s_N(M)$ for $N = 100$ and $M = 50$ over 1000 Replicates



Conclusions

- Experimental design is *active learning*.
- Other applications: cross-validation, chance-constrained problems, integration and massive kernel models (Haaland and Qian, 2011).
- A puzzle based learning approach: Sudoku, Samurai Sudoku, Sequential grids, Kakuro and Futoshiki.

References

[1] Haaland, B. and Qian, P. Z. G. (2011), "Accurate Emulators for Large-Scale Computer Experiments," *Annals of Statistics*, in press.

[2] Qian, P. Z. G. (2011), "Sliced Latin Hypercube Designs," *Journal of the American Statistcal Association*, in press.

[3] Xu, X., Haaland, B. and Qian, P. Z. G. (2011), "Sudoku-Based Space-Filling Designs," *Biometrika*, **98**, 711-720.

Thank you!