Construction of Symmetric Balanced Squares

T.K. Dutta and B.K. Roy
Indian Statistical Institute
203 Barrackpore Trunk Road
Calcutta 700035
India

ABSTRACT. Symmetric balanced squares for different sizes of array and for different numbers of treatments have been constructed. An algorithm, easily implementable on computers, has been developed for construction of such squares whenever the parameters satisfy the necessary conditions for existence of the square. The method of construction employs 1-factorizations of a complete graph or near 1-factorizations of a complete graph, depending on whether the size of the array is even or odd, respectively. For odd sized squares the method provides a solution directly based on the near 1-factorization. In the case of the squares being of even size, we use Hall's matching theorem along with a 1-factorization if $[\frac{n^2}{v}]$ is even, otherwise, Hall's matching theorem together with Fulkerson's [4] theorem, on the existence of a feasible flow in a network with bounds on flow leaving the sources and entering the sinks, lead to the required solution.

1 Introduction

Latin squares and symmetric latin squares are used in designing experiments, requiring two way elimination of heterogeneity in the absence of interaction in the model. The book by Denes and Keedwell [2] contains an excellent presentation of the subject. The combinatorial and algebraic features of the subject are covered, and also the applications to statistics and information theory are emphasized in it. For examples on the use of latin squares in agricultural research one may refer to Gomez and Gomez [5]. Here they have pointed out the limitations of such designs. The requirement that all treatments appear in each row-block and in each column-block is too stringent. As a result, when the number of treatments is large, the design becomes impractical. A more general class of squares, where such

ARS COMBINATORIA 47(1997), pp. 49-64

restriction is removed, is the class of balanced squares (BS's). A BS of size n in v elements is an $n \times n$ array, $D = ((d_{ij}))$, where d_{ij} denotes the element assigned to the j^{th} column in i^{th} row, $i, j, = 1, 2, \cdots, n$, satisfying the following conditions:

i) each element occurs r or $r+1$ times in every row and column (viz., rows and columns are balanced),

ii) each element occurs f or $f+1$ times in the array (viz., array is balanced).

A balanced square of size n in v elements will be abbreviated as BS (n, v). Construction of such a square is quite simple. The elements are to be written sequentially over a row. A new row will start with the element that follows the last element of the previous row. If the element already appears in the first position of one of the previous rows, and if not all the elements have appeared in the first column, then take an element which does not appear and repeat the process. Here is an example of BS (6,4). Note $r = 1$ and $f = 9$.

1	2	3	4	1	2
3	4	1	2	3	4
2	3	4	1	2	3
4	1	2	3	4	1
2	3	4	1	2	3
4	1	2	3	4	1

Table 1.1. BS(6,4)

A BS (n, v) is said to be symmetric if $d_{ij} = d_{ji}$, $i, j = 1, 2, \cdots, n$. A symmetric BS (n, v) is abbreviated as SBS (n, v).

If the number of elements (v) is the same as that of the size of the array (n), then an SBS (n, n) is a symmetric latin square of order n. The addition table of Z_n modulo n is such an example.

A method of construction for SBS (n, v) is discussed by P.J. Schellenberg (private communication). In our paper we propose an alternate method of construction. It uses 1-factorization/near 1-factorization of a complete graph, Hall's matching theorem and Fulkerson's feasibility theorem on network flow with bounds on flow leaving each source and entering each sink.

2 Definitions and some observations

Let $r = [\frac{n}{v}]$, the integral part of n/v, and let $f = [\frac{n^2}{v}]$, the integral part of n^2/v. It can be easily seen that

$$n = (n - rv)(r + 1) + (\overline{r+1}v - n)r \tag{2.1}$$

$$n^2 = (n^2 - fv)(f+1) + (\overline{f+1}v - n^2)f \tag{2.2}$$

Thus an SBS (n, v) has $(\overline{r+1}v - n)$ elements in r cells each and $(n - rv)$ elements in $r+1$ cells each in any row or column. Also, there are $(\overline{f+1}v - n^2)$ elements in f cells each and $(n^2 - fv)$ elements in $f+1$ cells each in the square.

An SBS being symmetric, any element with odd frequency must occupy a position on the main diagonal. So a necessary condition for the array to exist is that the number of elements with odd frequency must not exceed the size of the square, n. It is also easy to note that the maximum number of distinct elements that a symmetric square can accommodate is $n(n+1)/2$. The existence of SBS $(1, v)$ is a triviality. These observations imply the following lemma.

Lemma 2.1. *The necessary conditions for existence of an SBS* $(n, v), n > 1$ *are (i)* $v \leq n(n+1)/2$ *and (ii) the number of elements of odd frequency in the square should not exceed the size of the array,* n.

It can be seen that if SBS (n, v) exist for $v \leq n < 2v$ then SBS (n, v) exist for all $n \geq 2v$. Let $n = rv + s, 0 \leq s < v - 1,\ r \geq 2$. An SBS $(n, v), n \geq 2v$, can then be generated as

A	A	$\cdots$	A	A	B
A	A	$\cdots$	A	A	B
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
A	A	$\cdots$	A	A	B
A	A	$\cdots$	A	C	
B^T	B^T	$\cdots$	B^T		

where A is any symmetric latin square of order v, B is the first s columns of A and C is an SBS $(v + s, v)$.

Lemma 2.2. *If SBS* (n, v) *exist for* $v \leq n < 2v$ *then SBS* (n, v) *exist for all* $n \geq 2v$.

In view of the preceeding lemma, we will restrict our construction algorithm to n and v such that $n < 2v$. The proposed construction algorithm depends on 1-factorization/near 1-factorization of K_n depending on whether the value of n is even or odd, respectively. So we define and discuss it in subsequent sections.

Definition 2.1: Let n be even. A spanning subgraph of K_n consisting of $n/2$ vertex-disjoint edges is called a *1-factor* of K_n. A decomposition of K_n into $(n-1)$ disjoint 1-factors is called a *1-factorization* of K_n.

Definition 2.2: Let n be odd. A spanning subgraph of K_n consisting of an isolated vertex and $(n-1)/2$ vertex-disjoint edges is defined as a *near 1-*

factor of K_n. A decomposition of K_n into n disjoint near 1-factors is called a *near 1-factorization* of K_n.

These factorizations exist for all n. Now we proceed to provide one such construction which will be used in subsequent sections for construction of SBS's.

Let the vertices of K_{2n} be denoted by $0, 1, \cdots, 2n-2, \infty$. Define for $i = 1, 2, \cdots, 2n-1$, the ordered set of edges

$$S_i = \{(\infty, \overline{i-1})\} U \{(\overline{i-1+j},\ \overline{i-1-j}),\ j = 1, 2, \cdots, n-1\}$$

where each of the vertices $i-1+j$ and $i-1-j$ is expressed as one of the numbers $0, 1, 2, \cdots, 2n-2$ modulo $2n-1$. Clearly, the collection $\{S_i, i = 1, 2, \cdots, 2n-1\}$ is a 1-factorization of K_{2n}.

If the number of vertices is odd, say $2n-1$, then we use a near 1-factorization. This is obtained by just replacing the edge $(\infty\ \overline{i-1})$ in the ith factor of K_{2n}, as defined above, by the vertex $\{i-1\}$ to obtain a near 1-factor of K_{2n-1}.

The above edge-decomposition of K_n will be termed as an *array of edges* associated with the corresponding 1-factorization or near 1-factorization. Let there be an $n \times n$ array $D_n = ((d_{ij}))$ associated with this array of edges, whose row and column are labelled with the vertices of K_n. The edge (i, j) represents cells (i, j) and (j, i) of D. We say an element x is placed in the edge (i, j), if and only if, $d_{ij} = d_{ji} = x$; furthermore, we say that x is placed in vertex i of the near 1-factor S_{i+1} if and only if $d_{ii} = x$.

We say m edges are *consecutive* if either they are consecutive edges in the same row of the array of edges, or are divided between rows as consecutive edges at the left most end of a row and the right most end of the subsequent row. To be more precise, the factors are considered in the order

$$S_{2n-1} S_{2n-2} \cdots S_2 S_1$$

and the edges are scanned from right to left. The consecutiveness of edges discussed above implies these edges are consecutive edges in this ordered arrangement of factors. For example let us consider the following 1-factorization of K_8.

$$\begin{array}{cccc}
(\infty\,0) & (1\,6) & (2\,5) & (3\,4) \\
(\infty\,1) & (2\,0) & (3\,6) & (4\,5) \\
(\infty\,2) & (3\,1) & (4\,0) & (5\,6) \\
\vdots & \vdots & \vdots & \vdots \\
(\infty\,6) & (0\,5) & (1\,4) & (2\,3)
\end{array}$$

Here edges (2 5) and (3 4) are consecutive edges, so are the edges (∞ 0) and (4 5).

It is easy to note that any set of $n-1$ consecutive edges, either in the 1-factorization of K_{2n} or in the near 1-factorization of K_{2n+1}, contains $2(n-1)$ distinct vertices of the corresponding complete graph. Any complete row contains all the vertices. This observation is instrumental in providing an algorithm for the construction of SBS's.

3 Case $n < v$ and n odd

Let n and v satisfy the two necessary conditions for existence of an SBS(n, v). Let r and f satisfy the equations 2.1 and 2.2 respectively.

If f is odd then $\overline{f+1}v - n^2$ elements are to be placed in the array of edges using $\frac{1}{2}(f-1)$ edges each and $n^2 - fv$ elements using $\frac{1}{2}(f+1)$ edges each, provided all can be accommodated in the array without using the isolated vertices. Total number of edges required will be $n^2 - v(f+1)/2$ while there are $n(n-1)/2$ edges in the array. Thus, there will be a shortfall of $[n - (\overline{f+1}v - n^2)]/2$ edges, which are to be obtained from the diagonal cells, replacing each edge by two vertices, viz., two diagonal cells. The number of cells left on the diagonal after this allocation will be $\overline{f+1}v - n^2$, meant for allocating elements with odd frequency.

If f is even then all v elements are to be placed in the array of edges allocating $f/2$ edges each for every element. Thus we fall short by $[n - (n^2 - fv)]/2$ edges. These are obtained again from the main diagonal in the same way as before. The remaining $n^2 - fv$ positions left on the diagonal accommodate the elements with odd frequency.

In view of the above observations we propose a simple algorithm to construct an SBS (n, v) when n is odd. Before we proceed with the general algorithm, let us deal with two special cases, namely, $f = n-1$ and $f = n-2$.

Case $f = n-1$. Let $v = n + \Delta$, where $\Delta \geq 1$ as $n < v$. Note, $n^2 = (n-1)(n+\Delta) + [\Delta - (\Delta-1)n]$. Now $f = n-1$ implies $0 \leq \Delta - (\Delta-1)n < v$, viz., $0 < \Delta \leq 1 + 1/(n-1)$. Δ being integer, we have $\Delta = 1$.

So, an SBS $(n, n+1)$ has to be constructed. This is easily achieved by considering a symmetric latin square of order $n+1$, and then dropping any of the $n+1$ rows and the same column from the square, viz., if we drop ith row then ith column has to be dropped to get an SBS $(n, n+1)$.

Case $f = n-2$. Note $n^2 = (n-2)(n+\Delta) + [2\Delta - (\Delta-2)n]$, where $v = n+\Delta, \Delta \geq 1$. Now, $f = n-2$ implies $2 \leq \Delta \leq 2 + \frac{4}{(n-2)}$. So for $n > 6$, we have $\Delta = 2$, and for $n = 5$, $\Delta = 2$ or 3. But, if $\Delta = 3$, viz., $v = 8$, the necessary condition for existence of SBS is violated since the number of odd frequency elements is greater than the array size. So if $n = 5$ then $\Delta = 2$. The only other case to be looked into is that for $n = 3$.

If $n = 3$ then $2 \leq \Delta \leq 6$. For SBS to exist it is required that $v \leq n(n+1)/2 = 6$, this implies $2 \leq \Delta \leq 3$. Let us deal with the case $\Delta = 3$

separately. The following is an SBS (3, 6).

1	4	5
4	2	6
5	6	3

Table 3.1. SBS (3,6)

Now the remaining case, viz., construction of SBS $(n, n+2)$ is discussed. Since n is odd, consider the near 1-factorization of K_n. Allocate $n+2$ elements in the array of edges, each placed in $(n-1)/2$ consecutive edges or vertices, starting with the right most edge in the first row. The isolated vertex, $\{n-1\}$, remains unallocated. During this allocation process, 3 elements are placed in $(n-1)/2$ consecutive edges only; $n-1$ elements are placed in $(n-3)/2$ consecutive edges and a vertex. Three of these latter $n-1$ elements do not contain the vertex $\{n-1\}$. So, attach to any of these three elements the vertex $\{n-1\}$. Thus an SBS $(n, n+2)$ has been constructed.

Example 3.1: $n = 7$ and $v = 9$. Thus $r = 0$ and $f = 5$. This implies 4 elements occur with frequency $f+1 = 6$ and the remaining 5 with frequency $f = 5$. Consider the near 1-factorization of K_7.

0	(1 6)	(2 5)	(3 4)
1	(2 0)	(3 6)	(4 5)
2	(3 1)	(4 0)	(5 6)
3	(4 2)	(5 1)	(6 0)
4	(5 3)	(6 2)	(0 1)
5	(6 4)	(0 3)	(1 2)
6	(0 5)	(1 4)	(2 3)

	0	1	2	3	4	5	6
0	2	6	3	8	4	9	5
1		3	7	4	9	5	1
2			4	9	5	1	6
3				6	1	7	2
4					7	2	8
5						8	3
6							4

Table 3.2. SBS (7,9)

The above is an example of an SBS (7, 9) with frequency of elements 1,4,5 and 9 being 6 and the frequency of the remaining elements being 5.

Algorithm 3.1. (Case $n < v, n$ odd and $f < n-2$)

Step 1. Consider the near 1-factorization of K_n. Calculate the number of positions to be filled on the diagonal to account for the even parts. [An even part of an element with frequency f is $f-1$ if f is odd and is f if f is even.] This is $n-(\overline{f+1}v-n^2)$ or $n-(n^2-fv)$ depending on whether f is odd or even respectively.

Occupy these vertices by placing elements necessarily with even frequency, so that no vertex is repeated for the element placed using

the diagonals and the left most consecutive edges of the last row in the array of edges. A simple algebraic manipulation will give that number of elements of even frequency that occupy vertices of K_n is r, and p vertices of K_n are occupied by an element of even frequency where

$$\begin{array}{rcll} p & = & n-(\overline{f+1}v-n^2) & \text{if } f \text{ odd,} \\ & = & n-(n^2-fv) & \text{if } f \text{ even,} \quad \text{and} \\ r & = & \lceil \frac{p}{f} \rceil, & \text{if } f \text{ even,} \\ & = & \lceil \frac{p}{f+1} \rceil, & \text{if } f \text{ odd.} \end{array}$$

Step 2. Place elements, one by one, starting from right most end of first row, in the required number of consecutive edges and vertices of the near 1-factorization of K_n. During this allocation of elements, whenever an occupied vertex of K_n (that is, a vertex which already has an element placed in it) is encountered, it is simply passed over and the next edge in the sequence is considered. The elements placed in a vertex of the array of edges are the ones having odd frequency.

One can easily note that the construction has proceeded by maintaining the requirement on frequency of the elements, and as $f < n-2$, so the vertices occupied by each element are distinct. Thus the following lemma follows.

Lemma 3.1. *The algorithm 3.1 produces an SBS (n,v), when $n<v$, n is odd and $f<n-2$.*

Example 3.2: $n=7$ and $v=11$. Thus $r=0$ and $f=4$. So, there will be 5 elements with frequency 5 and 6 elements with frequency 4. Now consider the near 1-factorization of K_7. Total number of edges required to accommodate the even parts is $5\times 2+6\times 2=22$. The total number of available edges is $3\times 7=21$. So we fall short by 1 edge, which is accounted for by 2 diagonal positions, viz., 11th element contains the last edge (0, 5) and two diagonal cells from amongst 1,2,3,4 and 6.

0	(1 6)	(2 5)	(3 4)
1	(2 0)	(3 6)	(4 5)
2	(3 1)	(4 0)	(5 6)
3	(4 2)	(5 1)	(6 0)
4	(5 3)	(6 2)	(0 1)
5	(6 4)	(0 3)	(1 2)
6	(0 5)	(1 4)	(2 3)

	0	1	2	3	4	5	6
0	2	7	3	9	4	11	5
1		3	8	5	10	6	2
2			5	10	6	1	7
3				6	1	8	3
4					11	2	9
5						9	4
6							11

Table 3.3. SBS (7,11)

We allocated vertices 6 and 4 to the 11th element along with the edge (0 5). For the rest we start from (3 4), allocate 2 edges for elements not occupying a vertex and two edges plus a vertex for those occupying one. Thus the above is an SBS (7,11) where elements 1,4,7,8,10 and 11 appear 4 times and 2,3,5,6 and 9 appear 5 times in the array. No element appears twice in a row or column.

4 4. Case $n < v$ and n even

Let n and v satisfy the necessary conditions for existence, and let r and f satisfy equations 2.1 and 2.2.

Case f even. Since $f + 1$ is odd, any feasible solution must contain $n^2 - fv$ elements, those with frequency $f+1$ in the array, appearing on the main diagonal an odd number of times, and the remaining elements either do not appear, or appear even number of times, on the main diagonal.

Algorithm 4.1. (Case $n < v$, n even and f even)

Step 1. Consider the 1-factorization of K_n. Starting from the right hand end of the first row of the factorization, place elements one by one, in $f/2$ consecutive edges, until all the edges are exhausted or the left over edges are fewer than $f/2$.

Step 2. Let X be the set of vertices of K_n, Y be the set of elements allocated in step 1, and let E be the edges (x, y) such that the vertex x is not allocated to the element y. Consider the bipartite graph $G(X, Y, E)$. Get a matching of X into Y. Place the matched elements of Y in the corresponding main diagonal cells as represented by the matched vertices.

Step 3. Consider the next element not allocated in step 1, if any. If there are unallocated edges in the last row, allot them to this element. Continue the allocation on diagonal cells, counting two diagonal cells as one edge equivalent, so that no vertex is repeated.

Step 4. If there are unallocated elements, then allocate them on diagonals, at f positions each, replacing the matched elements of step 2. The positions occupied by the element placed in step 3 should not be disturbed. Continue this until all v elements are placed.

Remark 4.1. At least $n+1$ elements will be allocated $f/2$ edges each, in step 1.

Remark 4.2. The matching, discussed in step 2, exists.

Proof: $|X| = n < n + 1 \leq |Y|$, by remark 4.1. $\deg(y) = n - f$ for all $y \in Y$.

Note any vertex x will be contained in either $n-1$ or $n-2$ of the elements depending on whether or not the vertex stands allocated in the last factor S_{n-1}, viz., last row.

$$\text{So, } \deg(x) = u \text{ or } u+1, \text{ where } u = |Y| - (n-1).$$

Therefore, $\min_{x \in X} \deg(x) = u \geq n - f = \max_{y \in Y} \deg(y)$. Thus it follows from Hall's theorem on matching that X can be matched into Y.

Lemma 4.1. *The algorithm 4.1 constructs an SBS(n, v) when $n < v$ and, both n and f are even.*

Proof: Let $t = [\frac{n(n-1)}{f}]$, then in step 1 $y_1, y_2, \cdots, y_t$ elements are allocated to $f/2$ edges. Consequently, they have a frequency of f in the array. At step 2, for n of these t elements, the frequency in the array is increased to $(f+1)$.

At step 3 and 4, remaining elements are entered on the diagonal and/ or left over edges with an individual frequency of f in the array. Some of the n elements placed on the diagonal at step 2 are reduced to frequency f in the array.

Observe that the frequency of the elements in the array is f or $f+1$. The algorithm allocates elements in such a way that no vertex is repeated for any element. Thus the frequency of an element in any row or column is at most 1. Therefore, the algorithm yields an SBS(n, v).

Example 4.1. $n = 8$ and $v = 10$. Thus $r = 0$ and $f = 6$ and there are 4 elements in 7 cells each and 6 elements in 6 cells each in the array. An element appears at most once in any row or column. Consider the 1-factorization of K_8.

(∞0)	(1 6)	(2 5)	(3 4)
(∞1)	(2 0)	(3 6)	(4 5)
(∞2)	(3 1)	(4 0)	(5 6)
(∞3)	(4 2)	(5 1)	(6 0)
(∞4)	(5 3)	(6 2)	(0 1)
(∞5)	(6 4)	(0 3)	(1 2)
(∞6)	(0 5)	(1 4)	(2 3)

	∞	0	1	2	3	4	5	6
∞	9	2	3	4	6	7	8	10
0		1	6	3	8	4	9	5
1			2	7	4	9	5	1
2				$\not{8}_{10}$	9	5	1	6
3					$\not{5}_{10}$	1	7	2
4						$\not{3}_{10}$	2	8
5							$\not{6}_{10}$	3
6								7

Table 4.1: SBS (8,10)

X	:	∞	0	1	2	3	4	5	6
Y	:	1	①	②	2	3	③	4	4
		5	7	8	⑧	⑤	6	⑥	⑦
		⑨							9

Table 4.2: A matching of the diagonals.

The 10th element is placed in the edge $(\infty 6)$ and 4 diagonal positions 2,3,4 and 5. Thus an SBS(8, 10) with 4 elements 1,2,7, and 9 appearing in 7 cells each and the remaining 6 elements in 6 cells each in the array. No element occurs more than once in any row or column.

Case f odd. For any feasible SBS, the $\overline{f+1}v - n^2$ elements that occur with frequency f in the array, must appear an odd number of times in cells of the main diagonal, the remaining $n^2 - fv$ elements either do not appear or appear even number of times on the main diagonal.

Algorithm 4.2. (Case $n < v, n$ even and f odd).

Step 1. Set $t = (f+1)v - n^2$. As n and v satisfies the necessary conditions for existence, so $n - t \geq 0$ and $2|(n-t)$.

Let $l = t + (n-t)/2 = (n+t)/2$. Consider the 1-factorization of K_n.

Step 2. Place l elements in $(f-1)/2$ consecutive edges as before.

Step 3. Place the remaining $v - l$ elements in $(f+1)/2$ consecutive edges, starting from the unoccupied edge consecutive to the last edge of the lth element.

Step 4. Let X be the set of vertices of K_n, Y be the set of first l elements allocated in step 2, and let E be the set of edges (y, x) such that the vertex x is not allocated to the element y. Consider the bipartite graph $G(Y, X, E)$. Get a minimal vertex covering, C, of G, viz., t of the elements of Y have degree 1 in C and the remaining $l - t$ elements

of Y have degree 2 in C, and each vertex of X has degree 1 in C. Place the incident elements of Y in the corresponding diagonals as represented by the matched vertices.

Remark 4.3. All edges of the 1-factorization of K_n are occupied after the allocation of elements in step 3.

Remark 4.4. The minimal vertex covering, discussed in step 4, exists.

Proof: Let X and Y be as defined in step 4 and let E be the set of edges (y,x) such that the vertex x is not allocated to the element y. Consider the network $[Y\ U\ X,\ E]$ with the capacity function c defined as $c(y,x)=1$ for all $(y,x) \in E$, where Y is the set of sources and X is the set of sinks.

Let there be associated with each $y \in Y$, two non-negative numbers $a(y)$ and $a'(y)$, and with each $x \in X$, a non-negative number $b(x)$, defined by $a(y)=1, a'(y)=2$ for all $y \in Y$ and $b(x)=1$ for all $x \in X$.

Finding a minimal vertex covering in step 4 is equivalent to finding a feasible flow f satisfying

$$\begin{array}{rcll} a(y) & \leq & f(y,YUX)-f(YUX,y) \leq a'(y), & y \in Y \\ & & f(YUX,x)-f(x,YUX)=b(x), & x \in X \\ & & 0 \leq f(y,x) \leq c(y,x), & (y,x) \in E. \end{array} \tag{4.1}$$

Fulkerson [4] had shown that the constraint set (4.1) is feasible if and only if each of the constraint sets

$$\begin{array}{rcll} a(y) & \leq & f(y,YUX)-f(YUX,y), & y \in Y \\ & & f(YUX,x)-f(x,YUX) \leq b(x), & x \in X \\ & & 0 \leq f(y,x) \leq c(y,x), & (y,x) \in E. \end{array} \tag{4.2}$$

and

$$\begin{array}{rcll} & & f(y,YUX)-f(YUX,y) \leq a'(y), & y \in Y \\ b(x) & \leq & f(YUX,x)-f(x,YUX), & x \in X \\ & & 0 \leq f(y,x) \leq c(y,x), & (y,x) \in E. \end{array} \tag{4.3}$$

is feasible. So we proceed to show that the constraint sets (4.2) and (4.3) are feasible.

Note $|Y| = l \leq n = |X|.\deg(y) = n-(f-1)$ for all $y \in Y$. Let $r = \lceil \frac{l(f-1)}{n} \rceil$. Note r 1-factors are needed to accomodate l elements, as discussed in step 2. Any vertex x appears r times in these r 1-factors. So, either r of them or $r-1$ of them stand allocated, depending on whether it stands allocated in S_r. This implies, either $l-r$ or $l-r+1$ elements of

Y are not allocated to vertex x. So $\deg(x) = u$ or $u+1$ where $u = l - r$. Therefore, $\max_{x \in X} \deg(x) = u + 1 \leq n - (f-1) = \min_{y \in Y} \deg(y)$.

By Hall's matching theorem a matching of Y into X exists. Take any such matching $M \subseteq E$ and define $f(y,x) = 1$ if $(y,x) \in M$ and zero otherwise. Clearly f satisfies (4.2).

Now, Y', a set of $2l$ elements, is defined from Y as

$$y'_i = y_i, 1 \leq i \leq l,$$

$$y'_i = y_{i-\ell}, (l+1) \leq i \leq 2l.$$

Extend E to E' by adding $|E|$ edges by connecting $y'_{i+\ell}$ to all those x's to which y_i is connected, $i = 1, 2, \cdots, l$.

Note $|Y'| = 2l \geq n = |X|$. $\deg(y') = n - (f-1)$ for all $y' \in Y'$, and $\deg(x) = 2u$ or $2(u+1)$ for all $x \in X$. It can easily be shown that $2u \geq n - (f-1)$. Thus, $\max_{y' \in Y} \deg(y') = n - (f-1) \leq 2u = \min_{x \in X} \deg(x)$. Again, by Hall's theorem a matching of X into Y' exists. Take any such matching $M' \subseteq E'$ and define f' from Y to X by

$$\begin{aligned} f'(y_i, x) &= 2 \text{ if } (y_i, x) \text{ and } (y_{i+\ell}, x) \in M' \\ &= 1 \text{ if } (y_i, x) \text{ or else } (y_{i+\ell}, x) \in M' \\ &= 0 \text{ otherwise.} \end{aligned}$$

Clearly f' satisfies the constraint set (4.3).

Thus, there exists a feasible flow f satisfying the constraint set (4.1) which provides a minimal vertex covering as discussed in step 4.

Lemma 4.2. *The algorithm 4.2 produces an SBS(n, v) when $n < v, n$ is even and f is odd.*

The proof is simple and follows in the same way as that of lemma 4.1, and thus is omitted.

Example 4.2: $n = 8$ and $v = 11$. Thus $f = 5$, $t = 2$ and $l = 5$. So, there will be 9 elements in 6 cells each and 2 elements in 5 cells each in the array SBS(8, 11). Consider the 1-factorization of K_8. As suggested in step 2, l=5 elements, $1, 2, \cdots, 5$, say, are first placed in the array, each in 2 consecutive edges. Following step 3, elements 6, $\cdots$, 11 are placed in the array, each in 3 consecutive edges.

$(\infty 0)$	$(1\ 6)$	$(2\ 5)$	$(3\ 4)$
$(\infty 1)$	$(2\ 0)$	$(3\ 6)$	$(4\ 5)$
$(\infty 2)$	$(3\ 1)$	$(4\ 0)$	$(5\ 6)$
$(\infty 3)$	$(4\ 2)$	$(5\ 1)$	$(6\ 0)$
$(\infty 4)$	$(5\ 3)$	$(6\ 2)$	$(0\ 1)$
$(\infty 5)$	$(6\ 4)$	$(0\ 3)$	$(1\ 2)$
$(\infty 6)$	$(0\ 5)$	$(1\ 4)$	$(2\ 3)$

	∞	0	1	2	3	4	5	6
∞	3	2	4	6	7	9	10	11
0		1	8	4	9	5	11	6
1			5	9	6	11	7	2
2				3	10	7	1	8
3					5	1	8	3
4						4	3	10
5							2	5
6								4

Table 4.3: SBS (8,11)

X	:	∞	0	1	2	3	4	5	6
Y	:	1	①	1	2	2	2	②	1
		③	3	3	③	4	④	4	④
		5		⑤	5	⑤			

Table 4.4: A covering of the diagonals.

So, an SBS(8, 11) is constructed, where 2 elements, 1 and 2, appear in 5 cells each, while the remaining elements appear in 6 cells each. Each element occurs at most once in any row or column.

5 Case $v < n < 2v$

Here, the necessary conditions for existence of an SBS(n, v) are satisfied for all v and n. If an SBS $(n-v, v)$ exists, then SBS(n, v) can be generated as

$$\left[\begin{array}{c|c} A & B \\ \hline B^T & C \end{array}\right]$$

where A is any symmetric latin square of order v, B is the first $n-v$ columns of A, B^T is the transpose of B, and C is an SBS$(n-v, v)$.

If an SBS$(n-v, v)$ does not exist then it implies a violation of either of the necessary conditions for existence. If the condition on the feasible number of elements is violated, viz., if $v > \frac{1}{2}(n-v)(n-v+1)$, then $n-v$ of the v elements occupying the main diagonal occur with frequency 1. Of the remaining elements, $(n-v)(n-v-1)/2$ occur in the off-diagonal positions, each twice and the rest do not appear. So the array will not be balanced. An extended concept is defined in such situations. A symmetric square of size m is said to be near balanced if the rows and columns are balanced, and $|f_i - f_j| \leq 2, i, j = 1, 2, \cdots, v$, where f_i and f_j are frequencies of ith and jth elements respectively in the array. Such an array will be abbreviated as NSBS(m, v). The above array is thus an NSBS $(n-v, v)$.

The other case of non-existence of an SBS$(n-v,v)$ occurs when the number of elements with odd frequency is more than $n-v$. In such situations too, a near balance is attempted.

One can easily check that m^2 can be expressed as $m^2 = x_1(f'-1) + mf' + x_2(f'+1)$ where $x_1 + x_2 = v - m$ and f' is odd. So, to achieve near balance x_1, n and x_2 elements are to be placed in $f'-1, f'$ and $f'+1$ cells each respectively. This can easily be done by placing elements one by one, in the 1-factorization or near 1-factorization of K_m depending on m. The procedure is similar to the ones discussed in algorithms 3.1 and 4.1, and then finding a matching of the elements to the diagonal if m is even.

Example 5.1: $m = 7$ and $v = 10$. Thus $f = 4$ and this implies there has to be 1 element with frequency 4 and 9 elements with frequency 5 in a 7 $\times$ 7 symmetric array, which is infeasible. But $7^2 = 2.4 + 7.5 + 1.6$, viz., $f' = 5, x_1 = 2$ and $x_2 = 1$. So we find an NSBS(7,10).

0	(1 6)	(2 5)	(3 4)
1	(2 0)	(3 6)	(4 5)
2	(3 1)	(4 0)	(5 6)
3	(4 2)	(5 1)	(6 0)
4	(5 3)	(6 2)	(0 1)
5	(6 4)	(0 3)	(1 2)
6	(0 5)	(1 4)	(2 3)

2	6	3	8	4	10	5
	3	8	4	10	5	1
		4	9	6	1	7
			6	1	7	2
				7	2	9
					9	3
						10

Table 5.1: An NSBS (7, 10)

Lemma 5.1. *An SBS(n,v) exists for $v < n < 2v$.*

Proof: Let B be a $v \times \overline{n-v}$ matrix whose first column is b_1 where $b_1^T = (0, 2, 4, \cdots, 2[\frac{v-1}{2}], 1, 3, 5, \cdots, 2[\frac{v}{2}] - 1)$. Generate the $i+1$ th column of B from the ith column by moving the elements of the column up one position and placing the first element in the last position. In B, do not allow the column with 1 in the first position. In such a situation generate $n-v+1$ columns and discard the offending one.

Let C be an NSBS$(n-v,v)$ where $x_1, n-v, x_2$ elements have frequency $f'-1, f'$ and $f'+1$ respectively. Let $x = \min(x_1, x_2)$. Without loss of generality, elements $1, 3, 5, \cdots, 2x-1$ occur with frequency $f'-1$ and elements $0, 2, 4, \cdots, 2x-2$ occur with frequency $f'+1$. Consider the $n \times n$ array D,

$$D = \begin{pmatrix} \underset{v\times v}{A} & \underset{v\times\overline{n-v}}{B} \\ \underset{\overline{n-v}\times v}{B}{}^T & \underset{\overline{n-v}\times\overline{n-v}}{C} \end{pmatrix}$$

where A is the symmetric latin square generated as the addition table of Z_v. Note ith row of D contain $2(i-1)$ twice and $2(i-1)+1$ only once, $i = 1, 2, \cdots, [\frac{v}{2}]$. So by dropping the elements $0, 2, 4, \cdots$, from the diagonal

of A and replacing them by elements $1, 3, 5, \cdots$, the balance in D is restored. So D is an SBS(n, v) for $v < n < 2v$ after the modification.

Example 5.2: $n = 17, v = 10$.

Solution: Consider the NSBS(7, 10) constructed in the example 5.1. Let us renumber the elements in it as $1 \to 0, 2 \to 2, 3 \to 4, 4 \to 5, 5 \to 1, 6 \to 6, 7 \to 7, 8 \to 3, 9 \to 8$ and $10 \to 9$.

$D =$

	$A =$									$B =$						
$\boxed{1} \rightleftharpoons \boxed{0}$	1	2	3	4	5	6	7	8	9	0	2	4	6	8	3	5
1	2	3	4	5	6	7	8	9	0	2	4	6	8	1	5	7
2	3	4	5	6	7	8	9	0	1	4	6	8	1	3	7	9
3	4	5	6	7	8	9	0	1	2	6	8	1	3	5	9	0
4	5	6	7	8	9	0	1	2	3	8	1	3	5	7	0	2
5	6	7	8	9	0	1	2	3	4	1	3	5	7	9	2	4
6	7	8	9	0	1	2	3	4	5	3	5	7	9	0	4	6
7	8	9	0	1	2	3	4	5	6	5	7	9	0	2	6	8
8	9	0	1	2	3	4	5	6	7	7	9	0	2	4	8	1
9	0	1	2	3	4	5	6	7	8	9	0	2	4	6	1	3
										2	6	4	3	5	9	1
										6	4	3	5	9	1	0
										4	3	5	8	6	0	7
										3	5	8	6	0	7	2
										5	9	6	0	7	2	8
										9	1	0	7	2	8	4
										1	0	7	2	8	4	9
	B^T									$= C$						

Table 5.2. An SBS(17,10)

Note, frequency of 0 in the NSBS(7, 10) is 6 while that of 1 and 3 are 4. So, from the diagonal of the first row of the 17 × 17 matrix, 0 is removed and 1 is introduced. The modified array is then an SBS(17, 10).

Thus, based on the previous discussions, the following lemma follows.

Lemma 6. *An SBS(n, v) exists if and only if*

i) $v \leq n(n+1)/2$, *and*

ii) the number of elements with odd frequency is at most n.

References

[1] Claude Berge, *Graphs*, North-Holland, Amsterdam, 1985.

[2] J. Denes and A.D. Keedwell, *Latin Squares and their applications*, Academic Press, New York, 1974.

[3] L.R. Ford, Jr. and D.R. Fulkerson, A simple algorithm for finding maximal network flows and an application to the Hitchcock problem, *Can. J. Math.*, **9** (1957), 210–218.

[4] D.R. Fulkerson, A Network Flow Feasibility Theorem and Combinatorial Applications, *Can. J. Math.*, **11** (1959), 440–451.

[5] K.A. Gomez and A.A. Gomez, *Statistical Procedures for Agricultural Research*, John Wiley and Sons, New York, 1984.

[6] P.J. Schellenberg, Symmetric Balanced Squares, Private communication.