Generating Uniformly Distributed Random Latin Squares

Mark T. Jacobson
National Security Agency, Ft. Meade, MD 20755

Peter Matthews
*Dept. of Mathematics and Statistics, University of Maryland Baltimore County, Baltimore, MD 21228**

ABSTRACT

By simulating an ergodic Markov chain whose stationary distribution is uniform over the space of $n \times n$ Latin squares, we can obtain squares that are (approximately) uniformly distributed; we offer two such chains. The central issue is the construction of "moves" that connect the squares. Our first approach uses the fact that an $n \times n$ Latin square is equivalent to an $n \times n \times n$ contingency table in which each line sum equals 1. We relax the nonnegativity condition on the table's cells, allowing "improper" tables that have a single -1-cell. A simple set of moves connects this expanded space of tables [the diameter of the associated graph is bounded by $2(n-1)^3$], and suggests a Markov chain whose subchain of proper tables has the desired uniform stationary distribution (with an average of approximately n steps between proper tables). By grouping these moves appropriately, we derive a class of moves that stay within the space of proper Latin squares [with graph diameter bounded by $4(n-1)^2$]; these may also be used to form a suitable Markov chain. © **1996 John Wiley & Sons, Inc.**

1. SUMMARY

In this article, we construct Markov chain Monte Carlo algorithms for generating random $n \times n$ Latin squares: by simulating an ergodic Markov chain whose stationary distribution is uniform over this space, we can obtain Latin squares that are (approximately) uniformly distributed.

*This research was completed while the second author was participating in the National Security Agency Sabbatical Program.

CCC 1063-8539/96/050405-33
Journal of Combinatorial Designs, Vol. 4, No. 6 (1996)

The central issue is the construction of "moves" between Latin squares that connect the space, i.e., that allow us to eventually reach any Latin square from any other. Our first approach uses the fact that an $n \times n$ Latin square is equivalent to an $n \times n \times n$ contingency table in which each line sum equals 1; the dimensions of the table are identified with the rows, columns, and symbols of the Latin square. The moves are most easily described using the contingency-table representation. We relax slightly the nonnegativity condition on the table's cells, allowing tables that have a single -1-cell. From a table in this expanded space, a move consists of identifying a suitable $2 \times 2 \times 2$ subtable and incrementing or decrementing each cell by 1 in a way that leaves at most one -1-cell and all line sums equal to 1. From a "proper" contingency table, the subcube is determined by choosing one of the $n^2(n-1)$ 0-cells uniformly at random; the three lines containing the chosen cell each hold one 1-cell; these cells determine the remainder of the subcube and the incrementing/decrementing (the 1-cells get decremented). In a table that has a -1-cell, the three lines containing the -1-cell each hold two 1-cells; in each line, we choose one of these two 1-cells equiprobably to determine the subcube and the incrementing/decrementing (again, the 1-cells get decremented; the -1-cell gets incremented). This gives us an irreducible Markov chain on the expanded space of tables [the diameter of the associated graph is bounded above by $2(n-1)^3$], and the subchain of "proper" tables has the desired uniform stationary distribution (with an average of approximately n steps between proper tables).

By grouping these moves appropriately, we derive a class of new moves that stay within the space of proper Latin squares. Such a move either swaps elements between two rows of a Latin square, along a "cycle," or swaps elements among three rows, using "partial cycles" between two pairs of three rows. These moves also connect the space of Latin squares [with graph diameter bounded above by $4(n-1)^2$], and may be applied randomly to form a suitable Markov chain having a uniform stationary distribution.

Section 2 introduces the problem and our approach. In Section 3, we show that our subcube moves connect the expanded contingency-table space; we use these moves to produce the desired Markov chain in Section 4. In Section 5, we define the alternative moves and use them to produce another Markov chain. Section 6 discusses open issues of mixing and counting, and briefly considers the relevance of this work to the general problem of generating random contingency tables.

2. INTRODUCTION

A *Latin square* (LS) of order n is an $n \times n$ matrix in which each of n distinct symbols appears n times, once in each row and once in each column. Latin squares are of interest to statisticians who design experiments, and to mathematicians who study finite geometries. (References [3] and [4] are general Latin-square references.) Counting Latin squares is a challenging problem: $L(n)$, the number of order-n Latin squares (up to choices of the symbol set), is now known for $n \leq 10$ ([2], [7], [12]), but is still unknown for $n > 10$. There are bounds (see, e.g., [11]):

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

and

$$[L(n)]^{1/n^2} \sim n/e^2.$$

While considering alternatives for selecting Latin squares in an experimental-design context, F. Yates [13] wrote, "... it would seem theoretically preferable to choose a square at random from all the possible squares of given size." In this article, we attack this problem of generating uniformly distributed random Latin squares of a given order.

Why is this a challenge? Counting Latin squares is hard, and the problems of counting and random generation are, in general, closely related (see [10]). The difficulties are illustrated by a couple of unsuitable generation algorithms:

- We could generate random permutations of the symbols to fill a square a row at a time. Each permutation would be restricted to choices that would not cause column conflicts with the already-filled rows. (It is a consequence of P. Hall's marriage theorem that such choices always exist; see, e.g., [11].) However, we have no general way of weighting these choices appropriately in order to achieve the uniform distribution on Latin squares.
- We could generate uniformly distributed random permutations to fill a square a row at a time, restarting from scratch if we produce a column conflict. This algorithm terminates with probability 1, and it produces uniformly distributed random Latin squares. However, the expected number of "starts" we make before successfully completing an order-n LS is $n!^{n-1}/L(n) = e^{n^2(1+o(1))}$, which is unacceptable; the price we pay for uniformity is computational complexity. [A more feasible algorithm, with a similar flavor, was proposed in [8] for generating uniformly random $k \times n$ Latin rectangles; it runs in expected time $O(nk^3)$, provided $k = o(n^{1/3})$.]

In this article, we will generate random Latin squares using *Markov chain Monte Carlo* methods. Roughly speaking, we start with some Latin square of the desired order, and randomly "perturb" it to obtain a new square; repeated random perturbations lead us through a chain of squares. We may regard this chain as a random walk on a graph: the vertices correspond to the squares, and each possible perturbation of a square into another corresponds to an edge between the corresponding vertices. (If each perturbation can be reversed, we may regard the graph as undirected.) It is well known that a random walk on a finite, connected, nonbipartite undirected graph is ergodic, with stationary distribution assigning each vertex probability proportional to its degree.

3. HOW DO YOU PERTURB A LATIN SQUARE?

Our first challenge is to find "moves" between order-n Latin squares that allow us to reach any LS from any other; they must be manageable in the sense that it is easy to select moves (uniformly) at random during Markov-chain simulation. In this section, we will define a class of moves (± 1-*moves*) which are, in Sinclair's words, "simple local perturbations"; the catch is that applying a ± 1-move to a Latin square often produces something that is not quite a Latin square. These *improper squares* will nonetheless be important in helping us connect the *proper* (i.e., Latin) *squares*.

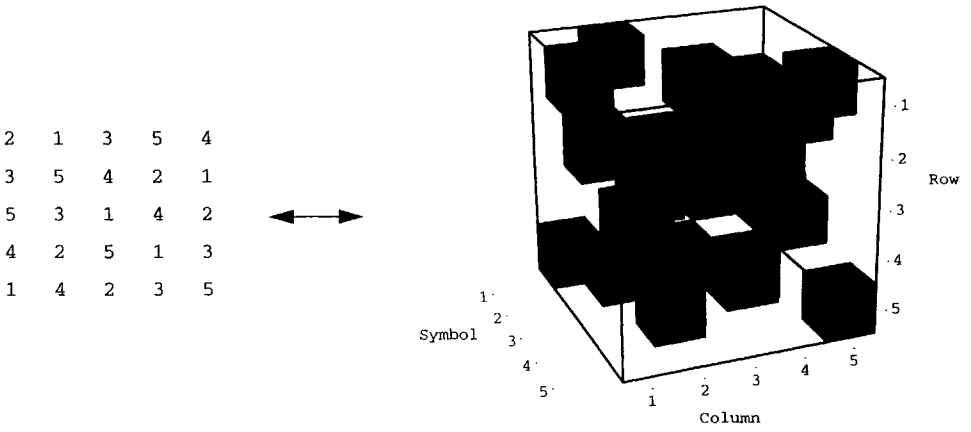


FIG. 1. The incidence cube for an order-5 Latin square.

In an expanded graph whose vertices correspond to the proper and improper order-n squares, the ± 1-moves (which are reversible) provide the edges; our task in this section is to prove the connectivity result:

Theorem 1. *Given two (proper or improper) order-n squares, there exists a sequence of ± 1-moves that transforms one square into the other. An upper bound on the length of the shortest such sequence is* $2(n-1)^3$ $(n \geq 2)$.

An alternative representation of a Latin square will be helpful. An $n \times n$ LS is equivalent to an $n \times n \times n$ 0-1 array whose dimensions are identified with the rows, columns, and symbols of the Latin square: a 1 appears in cell (r, c, s) of the array iff the symbol s appears in row r, column c of the LS. We call this an *incidence cube.* In Figure 1, we plot the 1-cells in the incidence cube that corresponds to an order-5 LS.

It follows immediately from the properties of Latin squares that if we fix any values for two coordinates of an incidence cube and let the remaining coordinate vary over its n values, this "line" of the cube will contain exactly one 1; the set of order-n incidence cubes is precisely the set of $n \times n \times n$ 0-1 cubes each of whose lines contains a single 1. (The n^2 coordinate triples of an incidence cube's 1-cells form the *orthogonal array* corresponding to the LS.) Figure 2 uses a single 1-cell to illustrate: in the three lines that contain it, all remaining entries will be 0. If we fix a value for a single coordinate of an incidence cube and let the other two coordinates vary over their values, we get a *plane* which is parallel to one of the three "coordinate planes"; we refer to such a plane as a row plane, column plane, or symbol plane according to the coordinate that is fixed. [Thus, for example, a symbol plane is a permutation matrix whose 1-entries indicate the (row, column) locations of the chosen symbol in the Latin square.]

The incidence-cube representation of a Latin square highlights a "symmetry of dimensions": any LS concept has analogs obtainable by permuting the notions of "row," "column," and "symbol" arbitrarily. This symmetry is less obvious in the usual matrix representation. For example: Looking at a Latin square in matrix form, we observe that transposing the matrix yields a Latin square as well. Transposition

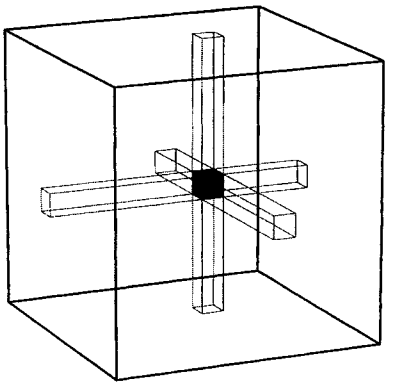


FIG. 2. Each line of an incidence cube contains a single 1-cell.

effectively swaps the roles of rows and columns; from the incidence-cube perspective, this is equivalent to swapping the labels "row" and "column" on the appropriate axes of the cube. But now it's clear that there are four other permutations of the three dimension labels, each of which produces an incidence cube corresponding to an additional *conjugate* of the Latin square.

Terminologically, we will use "Latin square" and "incidence cube" almost synonymously, the suitable representation normally being clear from context. If (r, c, s) is a (row, column, symbol) triple that indexes a 1-cell in the incidence cube, we say that column c and symbol s are incident in row r (with obvious analogs for incidences in columns and symbols); we may also simply say that "we have (r, c, s)." Similarly, "we have $\neg(r, c, s)$" means that the indicated incidence-cube entry is 0.

Conjugation is one way to transform a Latin square. We can also transform a Latin square by applying a *row permutation* (by this we mean that we swap entire rows with each other, not that we permute elements within rows), a *column permutation,* or a *symbol permutation* (a consistent relabeling of the symbols throughout the LS). In incidence-cube terms, such a permutation reorders the row planes, column planes, or symbol planes.

A less-obvious, more "fine-grain" type of transformation is illustrated below: we exchange portions of two rows (indicated by the box) by swapping symbols within columns.

```
b a c e d      b a c e d
c e d b a      c e a d b
e c a d b  →   e c d b a
d b e a c      d b e a c
a d b c e      a d b c e
```

We explain this transformation using a *row-pair graph,* which fully describes a specified pair of rows in a LS (Scheme 1, rows 2 and 3 of the original square). The graph comprises $2n$ vertices and $2n$ edges: there is a (labeled) vertex for each column and for each symbol (to distinguish column vertices from symbol vertices, we use numbers for columns and letters for symbols); an edge connects the vertices of each column-symbol pairing that occurs in either of the two rows, and each edge is labeled with the row of the corresponding occurrence. It is immediate that in a row-

pair graph, each vertex has degree 2 (with its incident edges labeled "oppositely"); the graph is a collection of disjoint cycles having alternating edge-labels; each cycle has even length (column vertices alternate with symbol vertices); no cycle length is less than 4. Now, swapping two symbols within a column corresponds to "toggling" the labels of a pair of adjacent edges in the appropriate row-pair graph; we see that in order to regain a row-pair graph that corresponds to a valid LS, we must toggle the remaining edge labels on the cycle. Performing the corresponding swaps between the two rows of the LS produces a new LS; we call this transformation a *cycle swap.* Swapping along *all* cycles in a row-pair graph has the effect of swapping two entire rows; thus, row permutations can be effected by a sequence of cycle swaps between pairs of rows.

There is nothing special about using rows: there are obvious definitions for column-pair graphs and symbol-pair graphs, and analogous cycle-swap operations. When we "distinguish" a particular dimension, we appear to lose some of the symmetry of the incidence-cube representation. The use of vertices for both remaining dimensions in, e.g., a row-pair graph reminds us that columns and symbols still have symmetric roles; this is easy to forget when we look at the row pair in the usual matrix representation.

Cycle-swapping and conjugation are, in general, insufficient to connect the space of order-n Latin squares: Let n be a prime greater than 3, and consider an order-n *circulant* [e.g., in which row r, column c holds the symbol $s \equiv r + c (r, c, s$ integers mod $n)$]. It is easy to show that every row-, column-, and symbol-pair graph in this LS is a single $2n$-cycle. This means that any cycle swap will swap a pair of entire rows or columns, or all occurrences of a pair of symbols; the omnipresent $2n$-cycle structure gets preserved, and relabeling the dimensions will not affect this. There *are* order-n Latin squares that have a different row-pair structure, but we cannot reach any this way.

Needing better moves, we reconsider the incidence cube. It has nonnegative integers as entries; in statistical parlance, this makes it an $n \times n \times n$ *contingency table* whose line sums all equal 1. The LS-generation problem is thus a special case of the problem of generating a uniformly random contingency table with given marginals; this has been studied lately by Diaconis and Sturmfels [6], but not solved for the general $n \times n \times n$ table of interest to us. Thinking in terms of line *sums,* we consider the following perturbation: select a $2 \times 2 \times 2$ subcube (corresponding to a pair of rows, a pair of columns, and a pair of symbols); to each cell of the subcube add either $+1$ or -1 in such a way that each line sum is unchanged (that is, in each subcube line, one cell is incremented and the other is decremented). Figure 3 illustrates; we call such an operation a ± 1-*move.* Now, a ± 1-move decrements four cells, and if any of these were 0-cells, they will become negative. However, we can limit the number of negative cells by considering only certain subcubes, as follows: Pick any 0-cell in the cube; it lies in three lines, each of which holds a single 1-cell. These cells are sufficient to determine a subcube. Figure 4 shows such a subcube determined by selecting the 0-cell in the bottom front corner. The line-sum constraints determine all cell values in the subcube with the exception of the one in the corner opposite the initially selected 0-cell. We make a ± 1-move by adding 1 to each of the four "known" 0-cells and subtracting 1 from each of the other four subcube cells. This produces either a "proper" incidence cube or a cube containing a *single* -1 value,

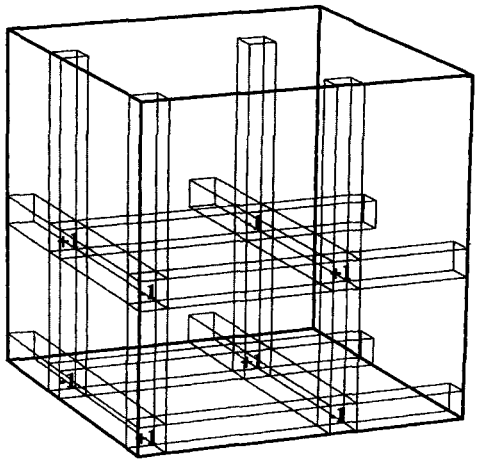


FIG. 3. Cell changes in a ± 1-move.

according as the "unknown opposite corner" held a 1 or a 0. In either case, each line sum still equals 1.

Thus, we are led to expand the space of cubes we consider to include those that have a single -1-entry. (All other entries must be 0's and 1's, and all line sums must equal 1.) We broaden our use of the term "*incidence cube,*" referring to a cube as being *improper* or *proper* according as it does or does not have a -1-entry. [We extend our earlier shorthand: if an improper cube's -1-entry occurs at row r, column c, symbol s, we say that "we have $-(r, c, s)$."]

Our interest, of course, is the set of proper cubes; the improper cubes are merely "stepping stones" to allow us to connect the subspace we are really interested in. If we can find a suitable Markov chain on the larger space, we could consider the (Markov) subchain consisting of the chain's proper cubes.

From each proper cube, there are $n^2(n-1)$ such moves we can make, corresponding to the possible 0-cell selections. The subcubes thus identified are exactly those containing either three or four 1-cells. Moves based on different 0-selections produce different cubes, unless the 0-cells reside on the same "density-4" subcube (such a subcube can be selected in four ways, each leading to the same new (proper) cube; we treat these as distinct moves, and note that the new cube, via the same subcube, indicates four "reversal" moves that return us to the old cube.)

What about moves from an improper incidence cube? We focus on the -1-cell. It lies in three lines of the cube, each of which has exactly two 1-cells (since the line sums equal 1). In each of these three lines, we select one of the 1-cells; these determine a $2 \times 2 \times 2$ subcube, as illustrated in Figure 5. Using the selected subcube, we make the ± 1-move that turns the -1 into a 0. The three 1-cell choices may be made independently, so there are $2^3 = 8$ of these moves, each of which produces a distinct new cube. If, in the selected subcube, the "undetermined" cell opposite the -1-cell holds a 1, the move produces a proper cube; otherwise, the new cube is improper, with the -1 moving to this opposite corner. (It is easy to see that, in either case, there is a "reversal" move that would return us to the old cube.) These 8 moves will suffice for us, but we note that there are other ± 1-move possibilities that

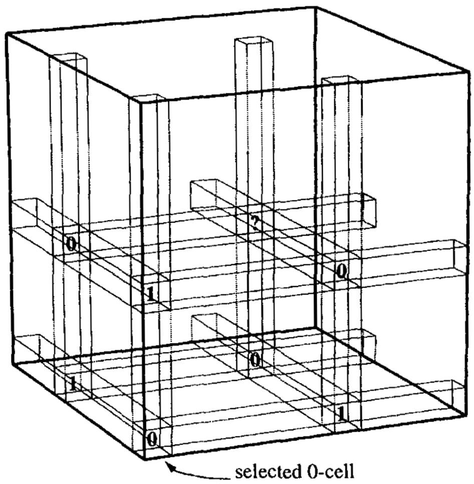


FIG. 4. *In a proper cube, a single 0-cell determines a ±1-move.*

we disregard: other choices of subcubes would result in moves that "slide" the -1 along one of its three lines (such an effect can be achieved by a pair of our moves), or in moves that do not move the -1 (if there are density-4 subcubes that do not include the -1-cell).

Our ±1-moves are easily expressed in matrix form, using symbolic arithmetic; we explain using examples.

a	b	c	d	e		a	b	c	e	d
b	c	d	e	a		b	c	d	d	a − d + e
c	d	e	a	b	→	c	d	e	a	b
d	e	a	b	c		d	e	a	b	c
e	a	b	c	d		e	a	b	c	d

In the *proper* (i.e., Latin) *square* on the left, we pick a cell (row 1, column 4) and a symbol not present there (e $\neq$ d); we locate the occurrences of this symbol in the selected row and column, determining a 2×2 subsquare, and alternately "add" and "subtract" the symbolic difference e − d to the cells in the subsquare so that the symbol in our selected cell gets replaced and row and column "sums" are preserved. [The choices correspond to the $n^2(n-1)$ possible moves.] On the right, we obtain a typical *improper square:* each row and column "sum" equals the sum of all the symbols; each cell has a single symbol, except for one *improper cell* (in the *improper row* and *column*) which has three (the *improper symbol* appears there with a -1 coefficient). Illustrating a ±1-move from this improper square, we select one of the two "positive" occurrences of the improper symbol from the improper row (the d in say, column 3) and another from the improper column (in, say, row 5), determining a 2×2 subsquare; we then select one of the two "positive" symbols in the improper cell (say, e), and again add and subtract the symbolic difference (e − d) around the subsquare, preserving row and column sums, and moving the impropriety. We obtain

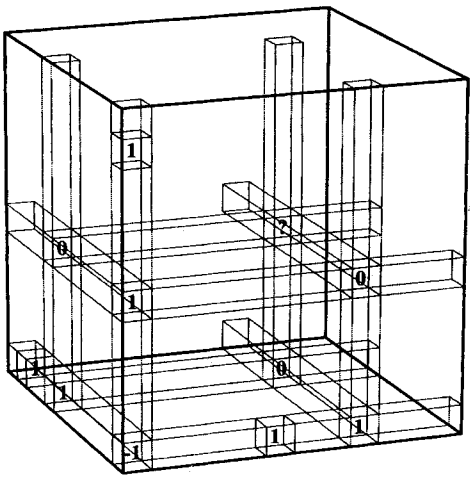


FIG. 5. Selection of a ± 1-move in an improper cube.

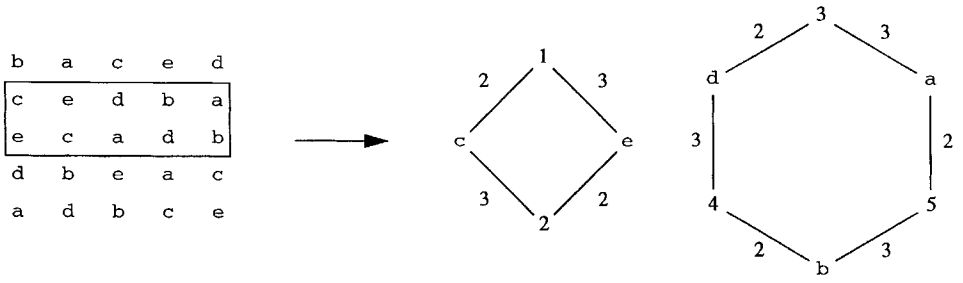
the improper square at the left:

	a	b	c	e	d		a	b	c	e	d
	b	c	**e**	d	**a**		**e**	c	**b**	d	a
→	c	d	e	a	b	→	c	d	e	a	b
	d	e	a	b	c		d	e	a	b	c
	e	a	**b + d − e**	c	**e**		**b**	a	**d**	c	e

(boxes indicate cells that change). One more ± 1-move, as indicated, can bring us to a proper square; note that this Latin square's first two rows form a row-pair graph comprising a 4-cycle and a 6-cycle. This is encouraging: we have started from an order-5 circulant, and have broken its stubborn 10-cycle structure.

En route to a connectivity proof, we extend the *row-pair graph* definition to cover improper squares/cubes. As before, a row-pair graph fully describes a specified pair of rows, using a (labeled) vertex for each column and for each symbol. An edge connects the vertices of each column-symbol pair that corresponds to a 1-entry in either of the pertinent row planes of the incidence cube. Again, we label each edge according to the row that induced it. Impropriety brings more variety to row-pair graphs: first, we observe that (for an order-n cube) a row-pair graph will have either $2n+1$ or $2n$ edges according as an improper row is or is not involved. (In an improper square or incidence cube, the -1-entry appears explicitly, but the -1-entry is implicit in a row-pair graph that involves an improper row.) We classify row-pair graphs into types, which we now enumerate; we provide examples using the improper square

a	f		b	e	c	d
d	e		a	f	b	c
b	c	d − e + f		a	e	e
f	a		e	c	d	b
c	d		e	b	a	f
e	b		c	d	f	a



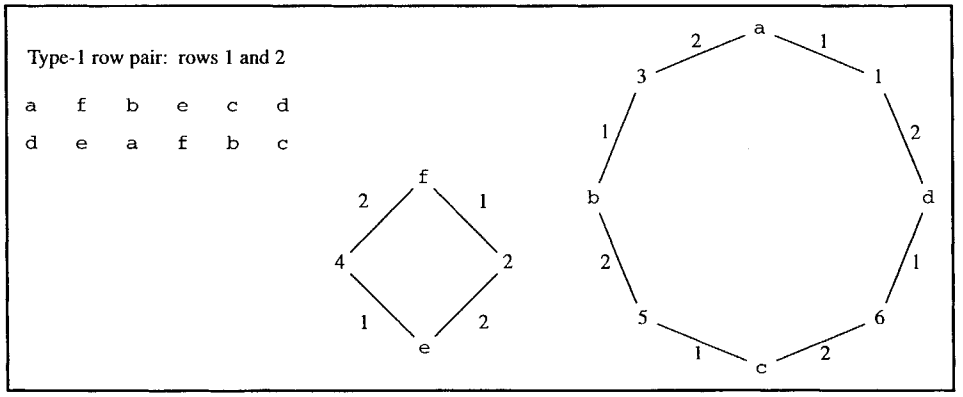
Scheme 1.

In our description, we will call the chosen rows r and r'.

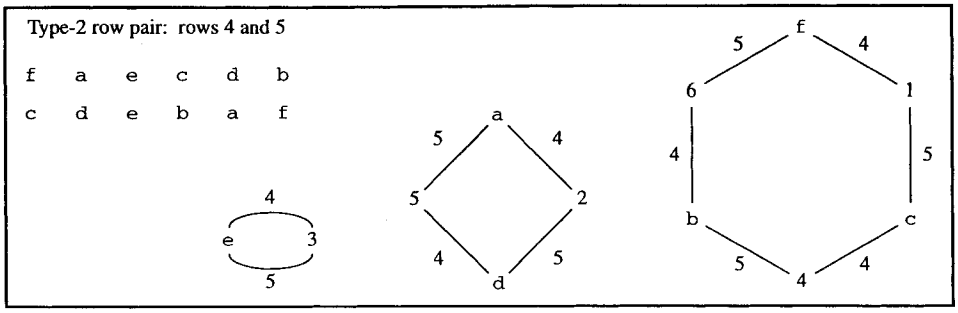
Type 1. Rows r and r' are proper, with no column conflicts. (We have seen this type before.) Each vertex in the graph has degree 2 (with incident edges labeled oppositely), so the graph is a collection of even-length cycles having alternating edge-labels (hereafter, these attributes will be implied when we refer to "cycles" in row-pair graphs); the cycles are disjoint, and none has length less than 4. For a proper square, every row-pair graph is of this type. Rows 1 and 2 of our sample square form a type-1 row-pair graph (Scheme 2).

Type 2. Rows r and r' are proper, but in one column the same symbol appears in both rows. (These are the two "positive" occurrences of the improper symbol in the improper column.) As in the type-1 case, the graph is a collection of disjoint cycles, but now we have a single cycle of length 2: there are two edges connecting the improper column and symbol vertices. All other cycles have lengths of at least 4. Continuing with our sample square, we see that rows 4 and 5 form a type-2 graph (Scheme 3).

The third type of row-pair graph occurs when one of the chosen rows is improper; things are more complicated, so we begin with an example (using rows 3 and 4 of our square, Scheme 4). In general, suppose we have $-(r, c, s)$. In the row-r plane of the



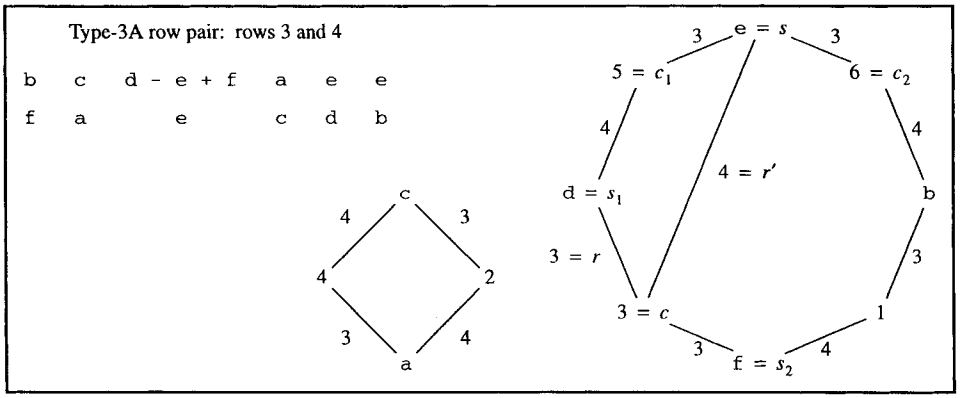
Scheme 2.



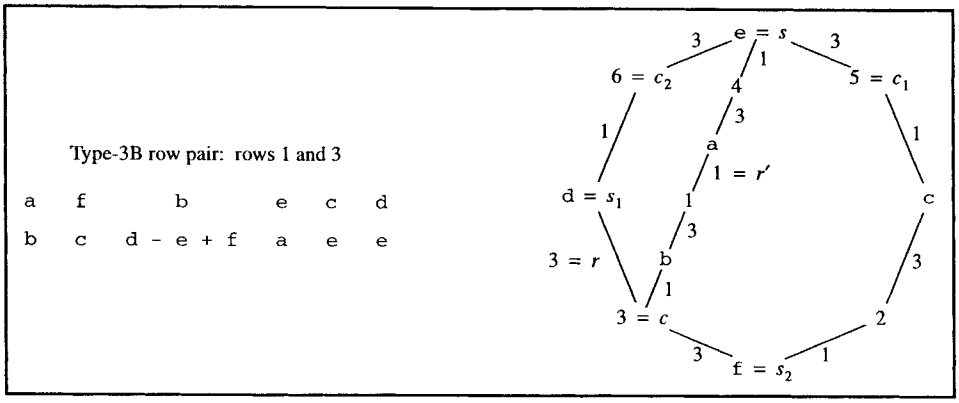
Scheme 3.

incidence cube, the two lines that contain the -1-cell each have two 1-entries; call these cells (r, c, s_1), (r, c, s_2), (r, c_1, s), and (r, c_2, s). (In our sample square, we have $r = 3$, $c = 3$, $s = \mathtt{e}$, $\{c_1, c_2\} = \{5, 6\}$, and $\{s_1, s_2\} = \{\mathtt{d}, \mathtt{f}\}$.) In the row-pair graph, each vertex has r'-degree 1; vertices c and s each have r-degree 2, but each other vertex has r-degree 1. Since c and s are the only vertices having odd (total) degree, they must belong to the same component of the graph; the other components, if any, are disjoint cycles of lengths of at least 4. (Working "backwards" from the row-pair graph, the improper column and symbol are identified by the degree-3 vertices; the improper row is determined by the "majority" edge-label of either degree-3 vertex.) Now, suppose we follow a path from c that starts with the r'-edge. Each non-s vertex we encounter has degree 2, so this path is uniquely determined (and has alternating edge-labels) until we reach s or return to c. We subclassify the row-pair graph according to what happens as we follow this path:

Type 3A. In this case the path reaches s immediately after leaving c; that is, we have (r', c, s). Every vertex in this component other than c and s has degree 2, which means that each of the two r-edges from s must begin a path (having alternating



Scheme 4.

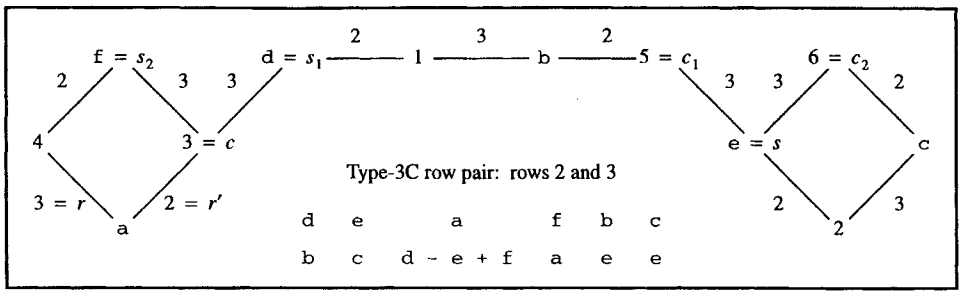


Scheme 5.

edge-labels) that reaches s_1 or s_2. (The example shown in Scheme 4 is a type-3A graph.) That is, the component is a pair of cycles that share the single (r'-)edge $\{c, s\}$; we will refer to this shared edge as the *chord* of a type-3A row-pair graph.

Type 3B. In this case we have $\neg(r', c, s)$, but our path reaches s (along an r'-edge) without having returned to c. (This means that the path reached neither s_1 nor s_2, which are adjacent to c via r-edges.) The situation is similar to the 3A case (but with a longer path between c and s): again, each of the two r-edges from s must begin an alternating-edge-label path that reaches s_1 or s_2. The component is a pair of cycles that share the odd-length path between c and s; since each of the three paths between s and c is at least 3-long, each of the two cycles is at least 6-long. Rows $r = 3$ and $r' = 1$ of our sample square offer us a type-3B graph (Scheme 5).

Type 3C. We have $\neg(r', c, s)$, and our path returns to c without encountering s. The last edge reaches c along an r-edge from either s_1 or s_2; without loss of generality, say it comes from s_2. (The path is a cycle, of length at least 4.) We now follow the (alternating-edge-label) path from c that begins with the untraversed (r-)edge from c, $\{c, s_1\}$; this path is uniquely determined until we reach s (along an r-edge). Now, in



Scheme 6.

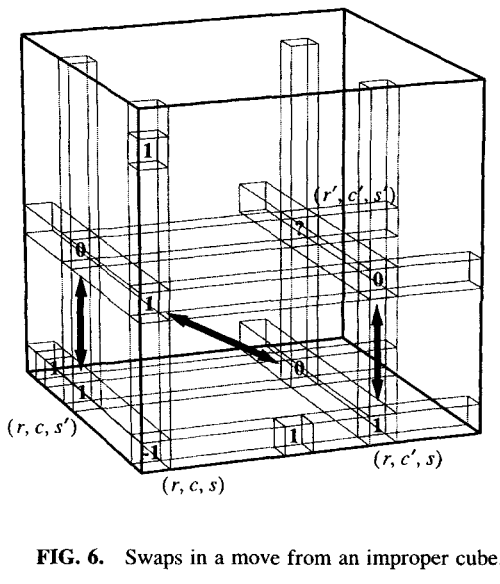


FIG. 6. Swaps in a move from an improper cube.

this component, only s and the yet-unvisited vertices are incident with untraversed edges; each of these vertices has an untraversed r-edge and an untraversed r'-edge, so these vertices and untraversed edges form a cycle. Thus, the row-pair graph comprises two disjoint cycles (each at least 4-long) linked via an odd-length *bridge* (with alternating edge-labels) between c and s. Our sample square provides us with an example of this final type, via rows $r = 3$ and $r' = 2$ (Scheme 6).

Row-pair graphs will help us navigate when we construct paths of ± 1-moves between squares. Each ± 1-move involves two rows; we consider the effect of such a move on the corresponding row-pair graph. First, suppose we have an improper square with $-(r, c, s)$, and that the opposite corner of the "move subcube" is to be (r', c', s'). r' is one of the two legitimate "opposite" row choices if and only if we have (r', c, s); put another way, a ± 1-move may be made between two rows of an improper square iff the rows form a type-3A graph. Similarly, c' and s' are suitable choices iff we have (r, c', s) and (r, c, s'); in terms of the rows r-r' row-pair graph, we can select the column (vertex) c' by traversing either of the two r-edges from s, and we can select the symbol (vertex) s' by traversing either of the two r-edges from c. Now, in terms of the subcube cells, we can interpret the effects of the move as follows (we refer to Figure 6, which pictures premove cell values).

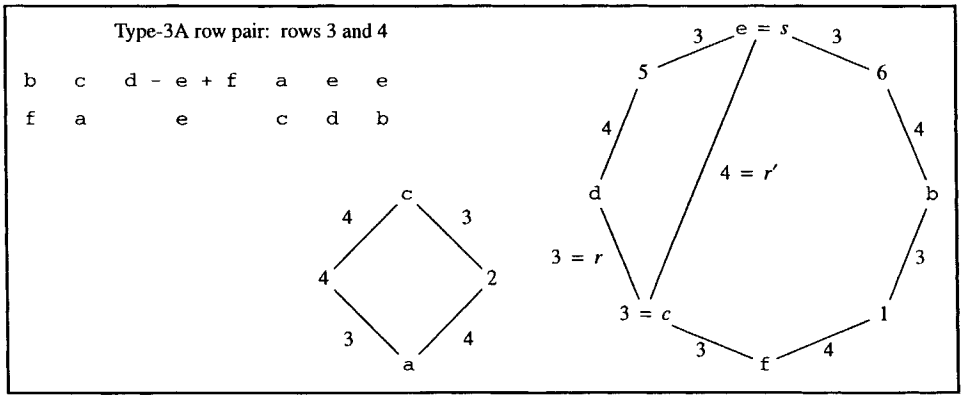
- We swap cell contents in pairs: (r, c', s) with (r', c', s), (r, c, s') with (r', c, s'), (r', c, s) with (r, c', s') (these swaps are indicated in the picture by arrows; in each case, we swap a 0-entry with a 1-entry). In the row-pair graph, the first two of these swaps have the effect of "toggling" the labels of the two edges we traversed in selecting c' and s' : each was labeled r before the move, but will be labeled r' afterward. The third swap deals with the chord: before the move, the chord was an r'-edge between c and s; the move replaces this edge with a new edge connecting c' and s' that has the opposite edge-label (r).
- The -1-entry in (r, c, s) becomes a 0 and the entry in (r', c', s') gets decremented. If (r', c', s') held a 0, these changes do not affect the row-pair graph

directly; we observe that the earlier swaps effectively "slid" each endpoint of the chord along one of the traversed edges, producing a new type-3A graph that reflects the impropriety which is now at (r', c', s'). If, on the other hand, (r', c', s') held a 1, the graph had an r'-edge $\{c', s'\}$ which the move now deletes; however, the earlier "chord change" produced a new $\{c', s'\}$ edge, labeled r. In this case, the cycle structure has changed: the chord has been "absorbed" into a single cycle in what is now a type-1 graph; the impropriety is gone; the move produced a proper square.

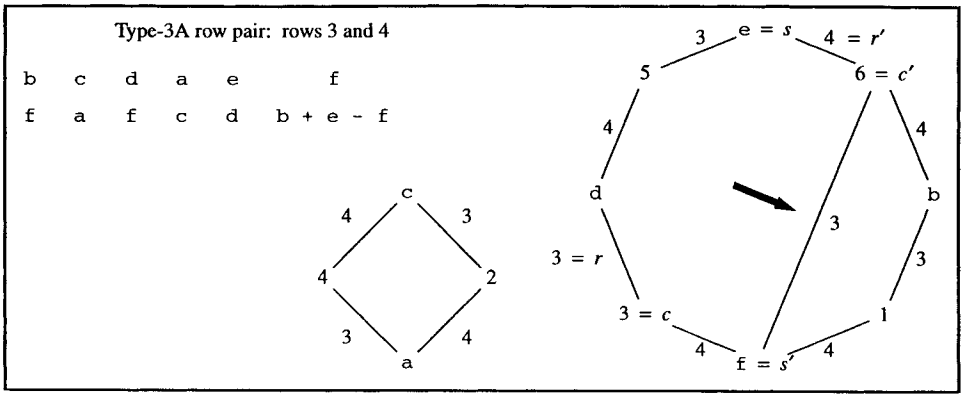
Using row-pair graphs, we have characterized the moves from an improper square: Find a type-3A row pair (there are two); slide each endpoint of the chord along one of its two identically labeled incident edges, toggling the labels of the chord and the two traversed edges. If the chord lands on an existing edge, it replaces it; we get a type-1 graph and a proper square. Otherwise, we get a new type-3A graph, indicating an impropriety (which has shifted to the "opposite" row). (N.B.: A move between rows r and r' affects the other row-pair graphs that involve either of these rows; we have not described the effects on those graphs.)

An an example, we consider rows 3 and 4 of our earlier improper square; they form a type-3A graph (Scheme 7). Sliding the column endpoint of the chord would take us to one of the symbol possibilities `d`, `f`; sliding the symbol endpoint would take us to one of the column possibilities 5, 6. Suppose we choose 6 and `f`; this produces the move that adds/subtracts `f − e` to the subsquare formed with columns 3 and 6. Sliding the chord to its new endpoints and toggling its edge label and the labels of the traversed edges, we get Scheme 8. Had we instead chosen 5 and `d` (adding/subtracting `d − e` to the subsquare formed with columns 3 and 5), the sliding chord would have replaced an existing edge. We would have moved to a proper square (Scheme 9).

Predictably, a move from a proper square introduces a chord into the appropriate row-pair graph. Recall that such a move amounts to picking a 0-cell in the (proper) incidence cube. There is a one-to-one correspondence between the $n^2(n-1)$ 0-cells in the cube and the $(2n)\binom{n}{2}$ edges in the collection of (type-1) row-pair graphs: a



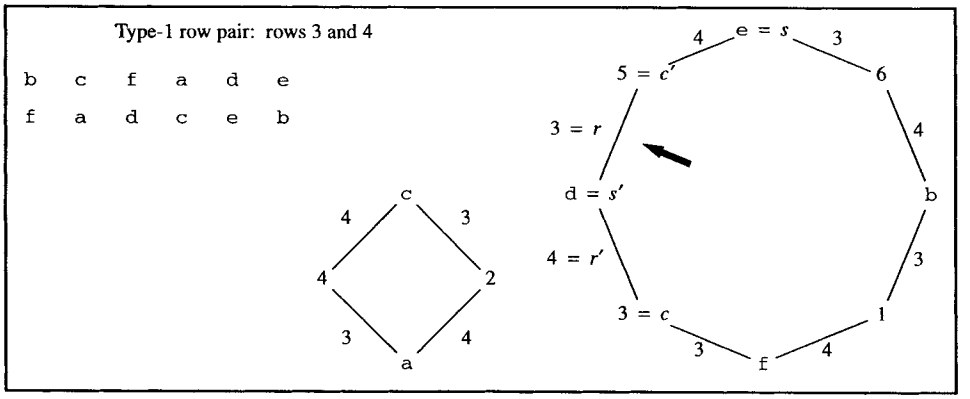
Scheme 7.



Scheme 8.

0-cell (r, c, s) corresponds to the r'-edge $\{c, s\}$ in the r-r' row-pair graph where r' is the unique row for which we have (r', c, s). This r'-edge is adjacent to two r-edges; traversing these edges leads us to a column vertex c' and a symbol vertex s'; these "one-step-away" endpoints give us the remaining coordinates for the move subcube. We again analyze the move using (slightly different) subcube-cell pairings (Fig. 7):

- Swapping cell (r', c, s) with (r, c, s), (r, c', s) with (r', c', s), and (r, c, s') with (r', c, s') corresponds to toggling the labels of the original r'-edge and the two r-edges we traversed to get to c' and s'.
- The 0-entry in (r, c', s') becomes a 1, which means that the r-edge $\{c', s'\}$ is added to the row-pair graph. If (r', c', s') held a 0, the move introduces an impropriety; this new edge is a chord in a type-3A graph. However, if (r', c', s') held a 1, the new edge *replaces* the old r'-edge $\{c', s'\}$; the graph component was a 4-cycle, corresponding to a density-4 subcube. In this case, the move toggles all four edges, effecting a cycle swap (and producing a proper square).



Scheme 9.

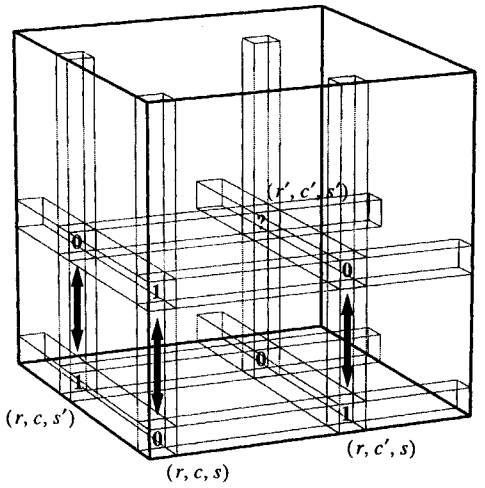


FIG. 7. Swaps in a move from a proper cube.

Again, we use row-pair graphs to characterize the moves from a proper square: Pick any row pair; pick an edge in the row-pair graph and toggle its label; from each of its endpoints, in opposite directions, traverse an edge and toggle its label; add an edge between the two points thus reached, labeling it to make cycles with alternating edge-labels. As before, if the new edge lands on an existing edge, it replaces it; in this case, we chose a 4-cycle, and we get another type-1 graph and a proper square (note that any choice of initial edge on a 4-cycle would produce the same square; this is the density-4 subcube situation we saw earlier). Otherwise, the new edge is a chord in a type-3A graph, and we get an improper square.

To demonstrate row-pair graph changes from a proper square, suppose that we start from where we left off in our last example. Scheme 9 shows rows 3 and 4 of a proper square. Picking the edge $\{5, \mathtt{d}\}$ in the row-pair graph corresponds to picking the 0-cell at $(4, 5, \mathtt{d})$ of the incidence cube; stepping away from this edge takes us to the vertices 3 and e. Toggling the original edge label, the labels on the edges we traversed, and adding the chord (with the label 4) produces the type-3A row pair of Scheme 7; this corresponds to adding/subtracting $\mathtt{d} - \mathtt{e}$ to the subsquare formed with columns 3 and 5. (We have demonstrated the reversal of the move that had given us the proper square.) If instead we pick any edge on the 4-cycle in Scheme 9, this leads us to add/subtract $\mathtt{a} - \mathtt{c}$ around the subsquare formed with columns 2 and 4, effecting the 4-cycle swap to produce another proper square (Scheme 10).

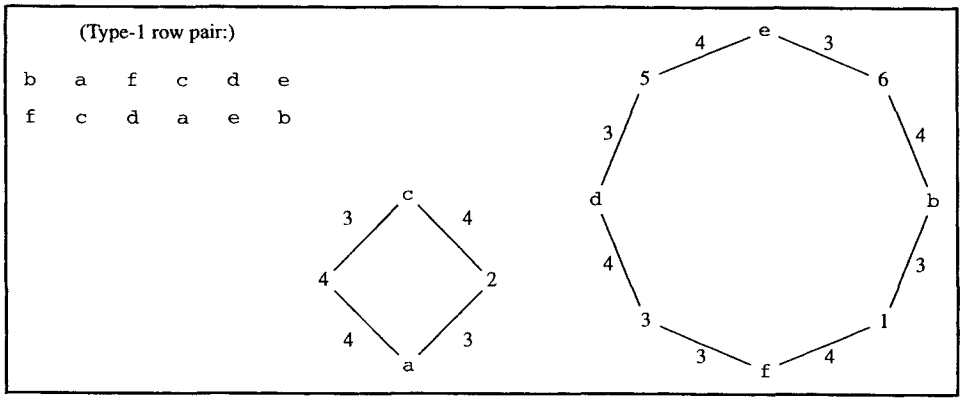
Suppose we have a proper square in which we want to swap along a longer cycle. The row-pair graph suggests a sequence of ± 1-moves that do this: Pick any edge on the cycle (in the type-1 row-pair graph) and "traverse and toggle" its two adjacent edges, introducing a chord (and producing a type-3A graph). This corresponds to the first ± 1-move. On each subsequent move, slide each endpoint of the chord one more step away from the original edge. When the chord replaces an existing edge, we again have a type-1 graph (and a proper square); the cycle swap is complete. We demonstrate by swapping along the 8-cycle in Scheme 9. Selecting the initial edge $\{5, \mathtt{d}\}$ produces the row pair in Scheme 7; another move takes us to the row pair in

Scheme 8, and a final move completes the cycle swap as shown in Scheme 11. In general: For a swap along a cycle of length $2k$ (in a proper square), we can start from any of the $2k$ edges in the cycle and accomplish the swap in $k-1$ moves. Notice also that if we begin with an improper square, we can quickly move to a proper square by making ± 1-moves between a pair of rows: we choose either of the type-3A row pairs and slide the chord until it is absorbed; this will never require more than $(n-1)/2$ moves.

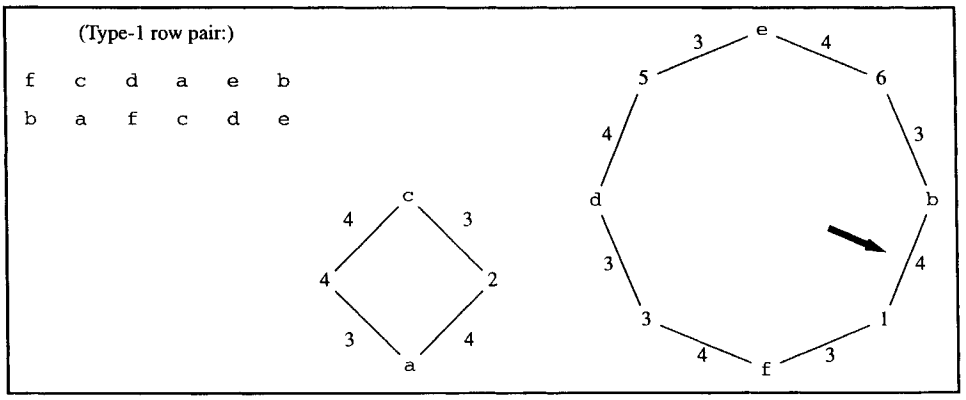
There is still nothing magic about the row perspective; consideration of column-pair graphs and symbol-pair graphs would be equally appropriate. (Note that a density-4 incidence subcube corresponds to 4-cycles in the pertinent row-, column-, *and* symbol-pair graphs.) For specificity's sake and representational clarity, we will generally use the row perspective.

We now outline our proof of connectivity. Given two proper squares, we start from one and construct a sequence of ± 1-moves (leading us through a sequence of *working squares*) that reaches the other (the *target square*). In doing so, we "correct" each row in turn so that it matches its target square counterpart; during construction, we identify the row currently being corrected as the *target row.* (In order to obey the single-impropriety rule, we may momentarily change rows that were already corrected, but we are careful to undo any such disturbances.) We correct a target row by effecting a sequence of two-element swaps within the row, repeatedly using the following lemma:

Lemma 2. *Suppose that we have a working square in which the target row t is proper with (t, c, s) and (t, c', s'), with the additional condition that, if the square is improper, we have $-(r, c, s)$ (for some $r \neq t$). For some (unique) r', we have (r', c', s); if the square is proper, define $r = r'$. Then there is a sequence of ± 1-moves that leaves a working square having (t, c, s') and (t, c', s) but, apart from making this swap in row t, changes incidences in only rows r and r'; additionally, if the new working square is improper, it has either $-(r, c, s')$ or $-(r', c, s')$.*



Scheme 10.



Scheme 11.

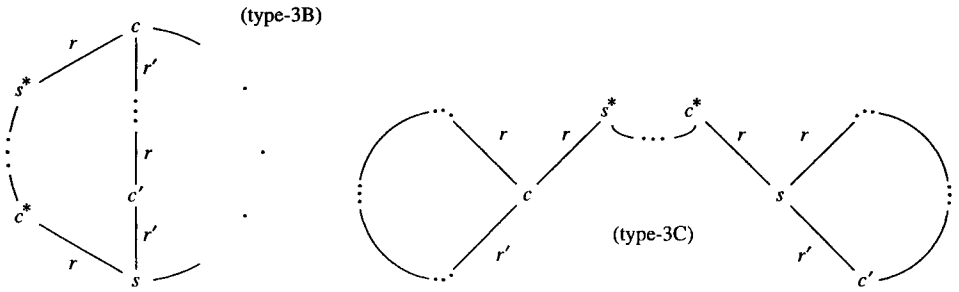
Proof. We may illustrate the pertinent portion of the working square as follows:

	col c	col c'
row t	s	s'
row r	$x+y-s$	
row r'		s

(this representation covers the proper-square case, if we take $y = s$). We consider two cases:

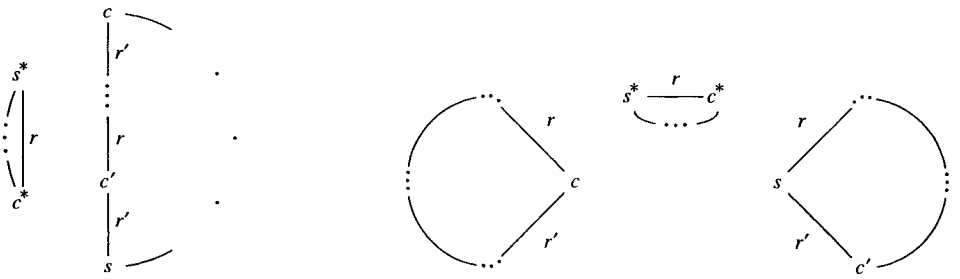
Case 1. $r = r'$; we're actually looking at a 2×2 subsquare. If $\{s, s'\} \cap \{x, y\} \neq \emptyset$, we can effect the swap with a single ± 1-move: we add/subtract $s' - s$ around the subsquare. Otherwise, we make two ± 1-moves on this subsquare: we add/subtract $x - s$ [momentarily giving us $-(t, c', x)$], and then add/subtract $s' - x$; the only *net* change to row t is the swap. If we are left with an improper square ($s' \notin \{x, y\}$), we have $-(r, c, s')$ as well as (t, c, s').

Case 2. $r \neq r'$; it's not as easy to make the swap. The goal is to reduce this to Case 1, by obtaining either (r, c', s) or $-(r', c, s)$; however, we cannot immediately make a move between rows r and r'. Consider the rows r-r' row-pair graph; $(r', c', s) \Rightarrow \neg(r', c, s)$, so the graph is type-3B or type-3C (Scheme 12). Let c^* and s^* be adjacent to s and c, respectively, via r-edges; choose them so that they are on the same c-s path (as shown). We will accomplish our stated short-term goal if we can change this graph by toggling either the edge label of $\{c', s\}$, or (instead) those of $\{c^*, s\}$ and $\{c, s^*\}$; first, we have to get the impropriety out of the way. For some row $u \notin \{r, r'\}$ we have (u, c, s); make the ± 1-move based on the "corners" (u, c, s), (r, c^*, s), and (r, c, s^*). Call this "move U" for future reference; we will be careful to undo its effect on row u later on. In row r, this removes the impropriety (possibly to row u) and the incidences (r, c^*, s) and (r, c, s^*), and produces the new incidence (r, c^*, s^*); this is reflected in the rows r-r' row-pair graph by deleting the two r-edges at the ends of the chosen c-s path and adding a new r-edge between c^* and s^*, as shown for the respective graph types (Scheme 13). Whichever type we had, we now have a collection of disjoint cycles: a type-2 or type-1 graph, according as the c-s path we chose had length 3 or length greater than 3 ...

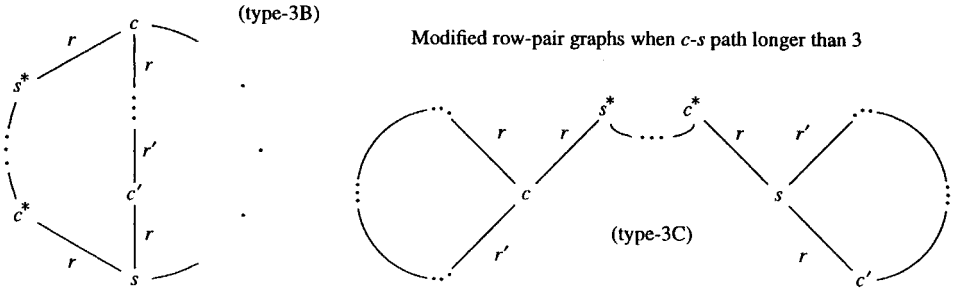


Scheme 12.

- We have $\neg(r', c^*, s^*)$ (and an r-r' graph of type 1) iff the c-s path we chose had length longer than 3. We will swap along the cycle containing c' and s, but we cannot do this right now if the square is improper. If it is [i.e., we have $-(u, c^*, s^*)$], define row $v \neq r$ such that we have (v, c^*, s^*); note that $v \notin \{r, r', u\}$. We make (and remember) a sequence of ± 1-moves between rows u and v to produce a proper square (by examining the rows u-v row-pair graph and sliding the chord). Now we can go back to the type-1 rows r-r' graph and make moves to swap the cycle containing c' and s; this gives us (r, c', s). From this proper square, we reverse any moves we just made between rows u and v. Row u is now as it was following move U; we have (u, c, s^*), (u, c^*, s), and (r, c^*, s^*), so we can reverse move U. The only net change has been to rows r and r' (we have toggled the edge labels on the cycle containing $\{c', s\}$ without changing the rest of the graph; see Scheme 14); since we now have both $-(r, c, s)$ and (r, c', s), we can proceed with the desired row-t swap as in Case 1.
- If the c-s path we chose had length 3, c^*, and s^* now lie on a 2-cycle; we have (r', c^*, s^*) and (r, c^*, s^*), so we must be improper with $-(u, c^*, s^*)$. We make the ± 1-move based on the "corners" (r', c^*, s^*), (u, c, s^*), and (u, c^*, s) (the latter two incidences were produced by move U). In row u, this move exactly reverses the changes that were made by move U. In row r', the move produces incidences (r', c^*, s) and (r', c, s^*) and removes (r', c^*, s^*). This reconnects things in the rows r-r' row-pair graph: the net effect of the two moves is



Scheme 13.



Scheme 14.

to toggle the labels of the three edges on the chosen c-s path; the new rows r-r' graph is of the same type as before (3B or 3C; see Scheme 15). Having $-(r', c, s)$ and (r', c', s), we can proceed as in Case 1.
The proof of the lemma is complete.

□

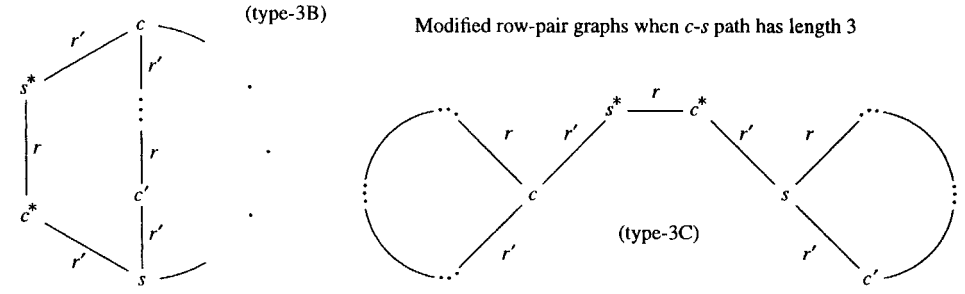
A suitable sequence of two-element swaps will fully correct the target row, but we need to organize them so that we always satisfy the impropriety condition of Lemma 2; for this, we introduce another tool. When the working-square target row is proper, we express its differences from its target-square counterpart as a *discrepancy graph:* a row-pair graph in which each edge is labeled with either a "W" or a "T" (for Working and Target) according to the square in which the incidence occurs. For example, if an order-6 target-square row holds

```
d  c  f  b  a  e
```

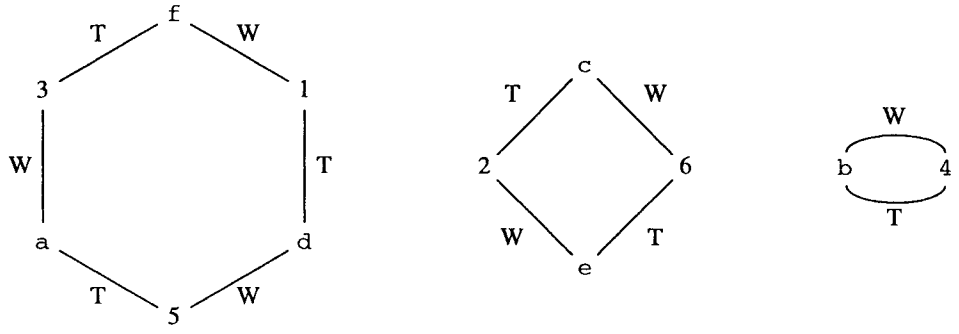
and the corresponding row of the working square holds

```
f  e  a  b  d  c
```

then we have a discrepancy graph as shown in Scheme 16. A discrepancy graph is a collection of disjoint, even-length *discrepancy cycles* with alternating edge-labels;



Scheme 15.



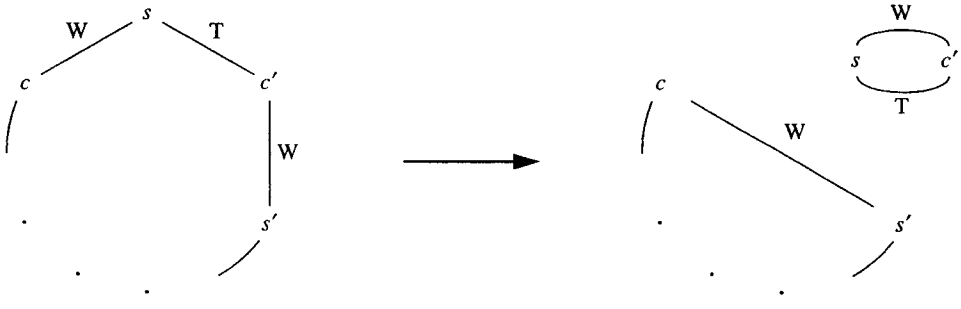
Scheme 16.

a length-2 ("trivial") discrepancy cycle indicates that, for the indicated column and symbol, what we have is what we want. The following lemma allows us to correct discrepancies along a nontrivial cycle with "minimal collateral impact":

Lemma 3. *Given proper working and target squares, select a nontrivial discrepancy cycle in (target) row t. Let R be (the indices of) the minimal set of other working-square rows that must be altered in order to correct this cycle. (That is, each T-edge $\{c, s\}$ in the discrepancy cycle indicates a column-symbol incidence that must move to row t from another (working-square) row r'; R comprises all such r'.) Then there is a sequence of ± 1-moves that corrects the discrepancies along the cycle, producing a proper square, without changing other incidences in row t or incidences in rows that do not belong to R.*

(Here, and in the proof, "change" again refers to *net* change: we allow moves to change a row (or a portion of the target row) that is "off-limits," provided that subsequent moves reverse such changes.)

Proof. Select any W-edge $\{c, s\}$ on the initial discrepancy cycle; let c' be adjacent to s via a T-edge, and let s' be adjacent to c' via a W-edge. We have (r', c', s) for some $r' \in R$; we use Lemma 2 (Case 1) to produce a working square having (t, c, s') and (t, c', s). This shortens the discrepancy cycle by chipping off a 2-cycle, as shown in Scheme 17 (note that if the new working square is improper, we have $-(r', c, s')$). Now redefine s to be s'. Until the shortened cycle becomes trivial, we can repeat this



Scheme 17.

process, using the (new) W-edge $\{c, s\}$ to lead us to new c', s', and r', with which we apply Lemma 2. Note that we change row t only by doing swaps involving columns and symbols on the original discrepancy cycle; in each other row r' that we change, the first (unreversed) change is made in order to move a targeted incidence into row t, so $r' \in R$. When we have reduced the original discrepancy cycle to a collection of 2-cycles, we may still have an impropriety $-(r, c, s)$. If so, we have (t, c, s) and (u, c, s) for some $u \neq t$; we can reach a proper square by making ± 1-moves between rows r and u (by sliding the chord in their type-3A row-pair graph). If these are the first (unreversed) changes to row u, then we had (u, c, s) in the original working square; the T-edge $\{c, s\}$ was in the original discrepancy cycle, so $u \in R$. This proves the lemma. □

We are finally ready to prove Theorem 1.

Theorem 1. *Given two (proper or improper) order-n squares, there exists a sequence of ± 1-moves that transforms one square into the other. An upper bound on the length of the shortest such sequence is $2(n-1)^3$ $(n \geq 2)$.*

Proof. Since we are never more than $(n-1)/2$ ± 1-moves from a proper square, it is sufficient to consider proper "endpoints" here. Pick a target row t, and consider any nontrivial row-t discrepancy cycle. To correct the discrepancies, column-symbol incidences (corresponding to the T-edges) must be moved to row t of the working square from other rows; however, none of these incidences will be found in rows that already match their target-square counterparts, so we may apply Lemma 3 with the assurance that the minimal impacted-row set R contains only uncorrected rows. We apply the lemma to nontrivial row-t discrepancy cycles until none are left; repeating this correction process on the remaining rows eventually brings us to the target square. The bound on the number of moves follows from examining the construction in the lemmas. Suppose we have a proper square and we go to work on a discrepancy cycle of length $2k$. To chip off the first 2-cycle takes 1 move. To chip off a subsequent 2-cycle, we use 1 or 2 moves (Lemma 2, Case 1), possibly preceded by moves to get us from Case 2 to Case 1: move U and its reversal (2 moves), the rows u-v cycle swap and its reversal (up to $n-1$ moves), and the appropriate rows r-r' cycle swap (up to n-3 moves). That is, chipping off the 2-cycle could require as many as $2n$ moves. When we have chipped off the initial 2-cycle and $k-2$ more, we may still need up to $(n-1)/2$ more moves to absorb a remaining impropriety. Summing this bound over all discrepancy cycles in a row gives us a bound on the number of moves required to correct the row; the sum is maximized when the row has a single $2n$-long discrepancy cycle:

$$1 + 2n(n-2) + (n-1)/2 < 2n(n-1) \text{ moves.}$$

When we have corrected $n-2$ rows, the rest of the job can be accomplished by cycle swaps between the two remaining rows (requiring no more than $n-1$ moves). With the allowance for improper "endpoints," our upper bound on the minimal number of moves is

$$2n(n-1)(n-2) + (n-1) + (n-1) = 2(n-1)^3.$$

The proof is complete. □

4. A LATIN-SQUARE MARKOV CHAIN WHOSE STATIONARY DISTRIBUTION IS UNIFORM

By Theorem 1, we can use the ± 1-moves to construct an irreducible Markov chain on the space of proper *and improper* order-n squares. We will consider the Markov subchain that consists of only the proper squares we hit. First, we define $\mu(n)$, the *average size-biased row-pair-cycle length* over order-n Latin squares:

$$\mu(n) = \left(2n \binom{n}{2} L(n)\right)^{-1} \sum_{L} \sum_{\{r,r'\}} \sum_{c} (l(c))^2$$

where $L(n)$ is the number of order-n Latin squares; the sums are over all order-n Latin squares L, all row pairs $\{r, r'\}$, and all cycles c in the r-r' row-pair graph of L; and $l(c)$ is the length of the cycle c. We do not know how to compute $\mu(n)$ in general, but obviously $\mu(n) \leq 2n$; we suspect that $\mu(n) = n + O(1)$. [We have $\mu(2) = 4$, $\mu(3) = \mu(4) = 6$, $\mu(5) = 58/7$, $\mu(6) = 424/49$.]

Theorem 4. *Let X_0^* be an arbitrarily distributed order-n Latin square that starts a Markov chain of (proper and improper) squares: to each square, apply a move chosen uniformly at random from the permissible ± 1-moves [$n^2(n-1)$ from a proper square, 8 from an improper square]. Let $X^* \equiv (X_0^*, X_1^*, X_2^*, \ldots)$ be the subsequence of proper squares we encounter; then X^* is a Markov chain with a (unique) stationary distribution that is uniform over the set of order-n Latin squares. If $n \geq 3$, the chain is ergodic. The expected number of ± 1-moves between X_{k-1}^* and X_k^* converges to $\mu(n) - 3$ as $k \to \infty$.*

Before proving Theorem 4, we digress to characterize what happens when we make sequences of ± 1-moves that stay within a chosen pair of rows. This will help us find a relation between the numbers of proper and improper squares, which is of importance in proving the hitting-time result.

In type-1 and type-3A row-pair graphs, we refer to *arcs*. An arc is an odd-length path of edges having alternating labels (one endpoint is a column vertex and the other is a symbol vertex). In a type-1 graph, an arc is *trivial* if it includes only one edge or all but one edge of the cycle on which it lies. In a type-3A graph, we will consider only the two (nontrivial) arcs *subtended* by the chord: the two edge-disjoint paths (excluding the chord) whose endpoints are those of the chord. We say that a proper square and an improper square are *reachable* from each other if there is a sequence of ± 1-moves, involving only two rows r and r', that transforms one into the other without encountering a proper square en route. In such a case, it is clear from the chord-sliding interpretations of the ± 1-moves (Section 3) that the corresponding r-r' row-pair graphs differ in only one component: one cycle of the type-1 graph comprises the two subtended arcs of the type-3A graph, with edge labels toggled along one of the subtended arcs. In the context of the initial square, we refer to this arc-to-be-toggled as the *toggling arc* that corresponds to the square we reach; this gives a one-to-one correspondence between the reachable squares and the nontrivial arcs in the initial square's (suitable) row-pair graphs: each such arc can be a toggling arc, and two sequences of r-r' moves that start from the same initial square produce the same new square if (and only if) they produce the same

new r-r' row-pair graph. For future reference, we call this collection of observations the *same-rows characterization.*

Proof of Theorem 4. In general: Let $X \equiv (X_0, X_1, \ldots)$ be an irreducible Markov chain with (stationary) transition probabilities and finite state space S. For $A \subseteq S$ and $k \geq 1$, define $T_A^{(k)}$ to be the kth hitting time of the set A :

$$T_A^{(k)} = \inf\{t \geq 1 : |\{n \leq t : X_n \in A\}| = k\}.$$

X is positive recurrent, and has a unique stationary distribution π with

$$\pi(a) = \frac{1}{E[T_{\{a\}}^{(1)} | X_0 = a]} > 0, \quad a \in S.$$

Now let A be a nonempty subset of S to which we wish to restrict the chain, and suppose $X_0 \in A$ almost surely; $P\{T_A^{(k)} < \infty \ \forall k \geq 1\} = 1$, so with probability 1, the sequence

$$X^* \equiv (X_0^*, X_1^*, X_2^*, \ldots) \equiv (X_0, X_{T_A^{(1)}}, X_{T_A^{(2)}}, \ldots)$$

is defined and is an irreducible Markov chain with state space A. The stationary distribution π^* of the subchain X^* is given by

$$\pi^*(a) = \frac{\pi(a)}{\pi(A)}, \quad a \in A.$$

Additionally, as $k \to \infty$,

$$\frac{T_A^{(k)}}{k} \to \frac{1}{\pi(A)} \text{ a.s., } E\left[\frac{T_A^{(k)}}{k}\right] \to \frac{1}{\pi(A)},$$

and if X^* is ergodic (i.e., we have aperiodicity),

$$E[T_A^{(k)} - T_A^{(k-1)}] \to \frac{1}{\pi(A)}$$

(where $T_A^{(0)} \equiv 0$.) That is, we get X^*'s stationary distribution by rescaling that of X (restricted to the set A); the expected number of X-steps between X^*-observations converges to the reciprocal of the probability of A under X's stationary distribution.

Now, let X be our proper-and-improper chain, and let A be the set of proper squares. The proper squares have identical probabilities under the stationary distribution of X; this gives us uniformity of X^*'s stationary distribution. For $n \geq 3$, there are order-n Latin squares with row-pair cycles longer than 4. Such a Latin square has positive probability of transiting to an improper square; if it does, there is then a $1/8$ chance that the move will be immediately reversed. That is, there are Latin squares from which the X^* chain has positive probability of staying put; the X^* chain thus has aperiodic states. (For $n = 2$, we note that there are no improper squares, and that our chain is, in fact, periodic: it alternates between the 2 order-2 Latin squares.) For the stationary distribution, π of X, we have

$$\pi(A) = \frac{n^2(n-1)L(n)}{n^2(n-1)L(n) + 8I(n)}$$

where $L(n)$ and $I(n)$ denote the numbers of proper and improper order-n squares, respectively; it remains for us to show that this equals $1/(\mu(n) - 3)$. For this, we need to know the relative sizes of $I(n)$ and $L(n)$. Consider the bipartite graph whose vertices are the proper and improper squares and in which an edge connects a proper square with an improper square iff they are reachable from each other. We can count the number of edges in this graph by summing either the degrees of the proper vertices L or the degrees of the improper vertices I :

$$\sum_{\text{proper } L} |\{\text{improper } I : I \text{ reachable from } L\}| = \sum_{\text{improper } I} |\{\text{proper } L : L \text{ reachable from } I\}|;$$

we now determine the summands. By the same-rows characterization, a proper square L can reach exactly as many improper squares as there are nontrivial arcs in its (type-1) row-pair graphs. For arcs on a row-pair cycle c of length $l(c)$, there are $l(c)/2 - 2$ possible nontrivial lengths; for each arc length, there are $l(c)$ different arcs along the cycle. Thus, the number of improper squares reachable from L is

$$\sum_{\{r,r'\}} \sum_{c} l(c) \left(\frac{l(c)}{2} - 2 \right)$$

where the sums are over all row pairs $\{r, r'\}$ and all cycles c in the rows r-r' row-pair graph of L. Conversely, an improper square I can reach 4 (distinct) proper squares: there are two possible toggling arcs in each of its two type-3A row-pair graphs. Substituting in the preceding equation, we get

$$\sum_{L} \sum_{\{r,r'\}} \sum_{c} l(c) \left(\frac{l(c)}{2} - 2 \right) = \sum_{I} 4;$$

recalling the definition of $\mu(n)$ (and noting that $\sum_c l(c) = 2n$), we have

$$2n \binom{n}{2} L(n) \left(\frac{\mu(n)}{2} - 2 \right) = 4I(n)$$

[which tells us that there are $O(n^4)$ times as many improper squares as proper ones]. Thus,

$$\pi(A) = \frac{n^2(n-1)L(n)}{n^2(n-1)L(n) + 8I(n)} = \frac{1}{1 + (\mu(n) - 4)} = \frac{1}{\mu(n) - 3},$$

completing the proof. □

It seems appropriate to include an example of a Latin square generated by simulating such a chain. Using Mathematica, we started with the 17×17 circulant whose first row and first column were

a b c d e f g h i j k l m n o p q,

and made ± 1-moves until we reached the 2400th proper square. Between successive proper squares, we took an average of 17.24 steps and made an average of 18.55 cell

changes. The order-17 Latin square we ultimately obtained:

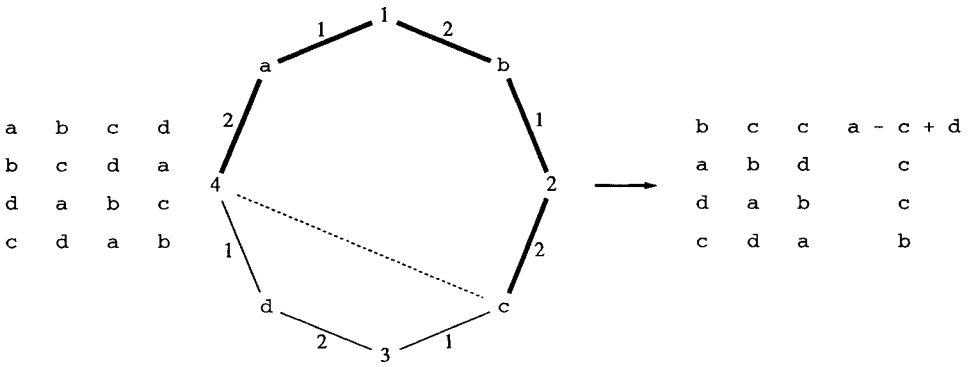
b	i	j	o	d	l	k	c	f	q	a	p	g	m	e	h	n
k	p	b	l	o	f	n	m	q	h	i	c	d	j	g	a	e
j	l	d	a	b	q	o	e	n	g	h	k	m	c	i	f	p
q	c	p	e	m	b	l	g	k	o	f	n	a	h	j	i	d
n	g	c	i	j	p	d	q	a	f	o	l	h	e	k	m	b
o	n	a	d	q	e	c	i	p	j	l	m	k	b	f	g	h
a	h	f	q	n	m	j	l	c	d	e	i	b	k	p	o	g
i	m	n	f	l	j	g	h	e	c	d	a	o	p	b	q	k
h	o	k	n	p	i	b	d	m	l	g	j	e	f	a	c	q
l	e	g	m	f	o	q	k	j	p	c	b	n	a	h	d	i
e	q	l	p	c	d	i	f	h	b	k	o	j	g	m	n	a
f	j	e	b	k	n	h	a	d	i	p	g	c	o	q	l	m
g	a	h	j	i	c	m	p	l	e	b	d	f	q	n	k	o
d	b	m	c	a	h	e	o	g	k	n	f	q	i	l	p	j
p	f	q	g	h	k	a	b	o	n	m	e	i	l	d	j	c
c	k	i	h	g	a	f	n	b	m	j	q	p	d	o	e	l
m	d	o	k	e	g	p	j	i	a	q	h	l	n	c	b	f

5. MAKING MOVES "PROPERLY"; ANOTHER MARKOV CHAIN

Expanding our state space to include improper squares allowed us to achieve connectivity using a simple set of moves (the ± 1-moves). Inspired by the same-rows characterization (Section 4), we now consider moves that stay within the space of (proper) Latin squares. There are two types of these *proper moves.* A *two-rowed* proper move is merely a cycle swap between a pair of rows. In *three-rowed* proper moves, we use improper squares as "stepping stones." Recall that each improper square can reach two proper squares via one row pair, and two other proper squares via another row pair (the improper row is common to both row pairs). A three-rowed move has two steps: the first step is to some reachable improper square; using this *pivot square*'s "other" suitable row pair, the second step moves to either of the reachable proper squares. (Here, the improper square is merely an artifice that allows us to think about proper moves in a simple and symmetric way; we do not actually stop on any. Later, we will describe these moves without using it.)

We demonstrate a three-rowed proper move using a 4×4 example (Scheme 18). We start from the Latin square on the left; using 5-long toggling arc we have highlighted in its rows 1-2 row-pair graph, we can reach the pivot square on the right. At this point, rows 1 and 3 form the other type-3A pair, as shown in Scheme 19; using the highlighted 3-long toggling arc, we reach the Latin square on the right, completing a proper move. (The alternative completion would use the subtended 5-long arc for toggling.)

Our connectivity results for the ± 1-moves will allow us to show that proper moves connect the Latin squares.



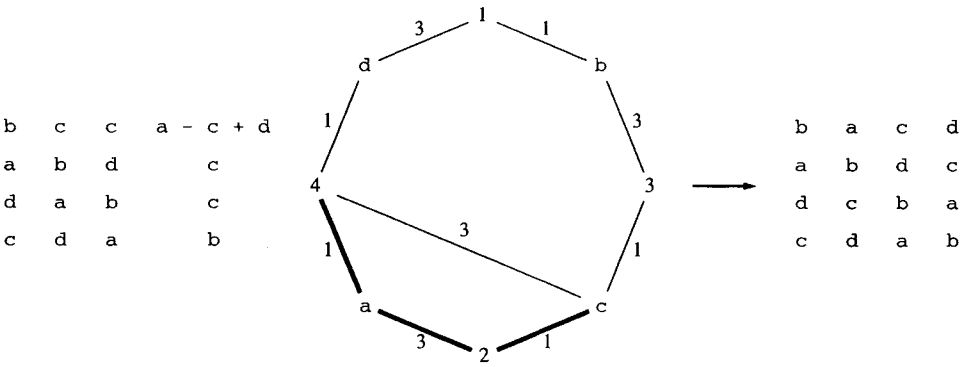
Scheme 18.

Lemma 5. *Suppose we have a Latin square P and a sequence of ± 1-moves that transforms it into another Latin square P' without encountering any other proper squares. The same transformation can be accomplished via a sequence of proper moves.*

Proof. Each ± 1-move involves two rows; partition the sequence into subsequences of consecutive moves such that

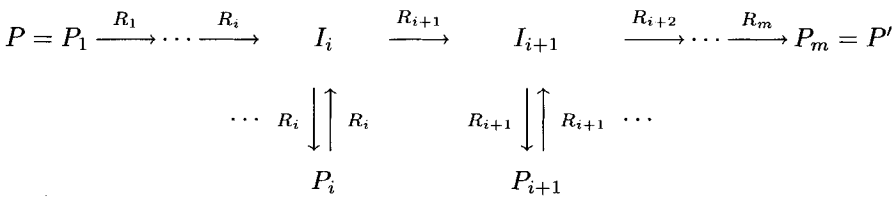
- all moves within a subsequence involve the same row pair;
- moves in two adjacent subsequences involve two different row pairs.

If there is but one subsequence, we get from P to P' via ± 1-moves that involve only two rows; by reasoning as in the same-rows characterization, we see that either $P' = P$ or P' is attainable from P by a cycle swap between the two rows—a single proper move. Otherwise, number the subsequences $1, 2, \ldots m$ $(m > 1)$, and let R_i be the row pair used in subsequence i; for $i < m$, let I_i be the improper square we have after applying the ± 1-moves through subsequence i (and note that R_i and R_{i+1} share exactly one row). Define $P_1 = P$ and $P_m = P'$, and for each $i \in \{2, \ldots, m-1\}$, let P_i be a square reachable from I_i using rows R_i. We may picture this as below: the



Scheme 19.

top line represents the original sequence of ± 1-moves, broken into subsequences; the intermediate P_i's underneath can be reached by inserting new ± 1-moves, which we immediately reverse.



For each $i < m$, P_i can reach I_i using rows R_i, and I_i can reach P_{i+1} using rows R_{i+1}; that is, we have a three-rowed move from P_i to P_{i+1}, and thus a sequence of $(m-1)$ three-rowed moves from P to P'. This proves the lemma. □

Theorem 6. *For any pair of Latin squares, there exists a sequence of proper moves that transforms one into the other. An upper bound on the length of the shortest such sequence is* $4(n-1)^2$ $(n \geq 2)$.

Proof. By Theorem 1, there is a sequence of ± 1-moves that transforms the initial LS into the target, possibly encountering intermediate proper squares. Apply Lemma 5 to the subsequences of consecutive ± 1-moves that "span" successive proper squares; this proves connectivity. From the proof of Lemma 5, it is easy to see that the number of proper moves required is bounded by the number of times, in the course of the ± 1-moves, that the row pair changes or we move from a proper square. We return to the construction in Lemmas 2 and 3, and count conservatively: For a discrepancy cycle of length $2k$, one move from a proper square chips off the first 2-cycle. Chipping off a subsequent 2-cycle could involve 6 row pairs ("Case 2": move U, the rows u-v swap, the rows r-r' swap, the rows u-v unswap, the reversal of move U; "Case 1": a further row pair), but if we choose $u = t$ in "move U" of Lemma 2, Case 2, this collapses to no more than 4 row pairs. After all discrepancies are resolved, absorbing an impropriety may involve another row pair. Summing over all discrepancy cycles in a row gives us a bound on the number of proper moves needed to correct the row; again, the sum is maximized when the row has a single $2n$-long discrepancy cycle:

$$1 + 4(n-2) + 1 = 4n - 6 \text{ moves.}$$

When we have corrected $n-2$ rows, all that is left is to swap some cycles between the two remaining rows; each cycle swap is a proper move (no more than $n/2$ required). The bound on the minimal number of proper moves is thus

$$(4n-6)(n-2) + n/2,$$

which is smaller than $4(n-1)^2$ for $n \geq 2$. The proof is complete. □

We now use proper moves to produce a satisfactory Markov chain.

Theorem 7. *Consider the following Markov chain* $Y \equiv (Y_0, Y_1, \ldots)$ *of order-n Latin squares: Let* Y_0 *be distributed arbitrarily; obtain* Y_t *from* Y_{t-1} $(t > 0)$ *by making a proper move chosen randomly as follows:*

- *Select an ordered pair of rows* (r', r) *uniformly at random from the* $n(n-1)$ *ordered pairs of distinct rows; select* l *uniformly at random from* $\{2, 3, \ldots, n\}$;

select a uniformly random vertex from Y_{t-1}'s r-r' row-pair graph (the choice may be restricted to column vertices or to symbol vertices). Let $2k$ denote the length of the cycle containing the chosen vertex.

- *If $l \geq k$, make a two-rowed move by swapping along the row-pair cycle containing the chosen vertex. Otherwise ($l < k$), make a three-rowed move: Define a toggling arc of length $2l - 1$ by starting from the chosen vertex and proceeding along the path that begins with the incident r'-edge; this nontrivial arc determines the pivot square. The arc-endpoints are incident in some third row t; from the pivot square's (type-3A) r-t row-pair graph, select a toggling arc equiprobably from the two subtended arcs, and move to the corresponding proper square.*

Then Y has a (unique) stationary distribution that is uniform over the set of order-n Latin squares; if $n \geq 3$, the chain is ergodic.

(We will call r' the *initial row,* r the *pivot row,* the arc-endpoints the *pivot column* and *pivot symbol,* and t the *alternate row.*)

Proof. A three-rowed move corresponds to choices of an "initial" toggling arc (which leads to the pivot square) and a "completion" toggling arc (which leads from the pivot square to the new proper square); in the chain here, each three-rowed move from a LS has probability $1/(2n^2(n-1)^2)$ of being selected. Different three-rowed moves *may* produce the same LS, but not if they use the same pivot square; for each three-rowed move, there is exactly one move that reverses it "through" the same pivot square. Hence, for any Latin squares L and L',

$$\begin{aligned} \mathrm{P}\{L \to L' \text{ via a 3-rowed move}\} &= \sum_I \mathrm{P}\{L \to L' \text{ via pivot square } I\} \\ &= \sum_I \mathrm{P}\{L' \to L \text{ via pivot square } I\} \\ &= \mathrm{P}\{L' \to L \text{ via a 3-rowed move}\} \end{aligned}$$

(where $L \to L'$ denotes a transition to L' given that the chain starts from L). Also,

$$\begin{aligned} \mathrm{P}\{L \to L' \text{ via a 2-rowed move}\} &= \mathrm{P}\{L' \to L \text{ via a 2-rowed move}\} \\ &= \frac{2k(n-k+1)}{n^2(n-1)^2} \text{ or } 0, \end{aligned}$$

according as L and L' do or do not differ by a single $2k$-long row-pair cycle. By Theorem 6, Y is irreducible; combining two- and three-rowed-move results, we have $\mathrm{P}\{L \to L'\} = \mathrm{P}\{L' \to L\}$, so Y is time-reversible and has the uniform stationary distribution. (We did not need the fact that it is impossible to step from one LS to another by *both* a two-rowed move and a three-rowed move; we leave this as an exercise.) It remains to show that Y has aperiodic states for $n \geq 3$. Let L be a Latin square from which we can make a three-rowed move (for $n \geq 3$, there are such LSs): suppose that L can move to L' via the pivot square I. Following this move with its reversal gives us a two-move sequence that returns to L. Now, L reaches I via some toggling arc that lies on one of L's row-pair cycles; begin a new sequence from L with a full swap of this cycle (a two-rowed move). This takes us to a square L^* which can also reach I, so there is a three-rowed move from L^* to L'. Appending

the original $L' \to L$ move completes a three-move return to L, so L is an aperiodic state; Y is ergodic. The proof is complete. □

Note that we have defined proper moves from a particular "dimensional perspective": we have distinguished the row dimension. With obvious changes to definitions, we get proper moves that are based on column- or symbol-pair graphs, and corresponding results. When running a Markov chain using proper moves, we might want to randomly select the "perspective" (row, column, symbol) of each move.

A proper move, as chosen by Theorem 7, is easy to describe directly in terms of the usual matrix representation of a Latin square. (We do lose some dimensional symmetry in the wording.) Select an initial row r', a (different) pivot row $r \neq r'$, a length $l \in \{2, \ldots, n\}$, and a pivot symbol s uniformly at random [$n^2(n-1)^2$ possibilities]. Swap elements within columns of rows r' and r, beginning with the column that has s in row r' and then following the row-pair cycle (as dictated by duplicate symbols in row r'). Swap in no more than l columns: If a cycle swap is completed before the lth swap, end the (two-rowed) move; otherwise, in the lth column, swap between row r' and whatever *alternate* row t holds the pivot symbol. We illustrate this "first stage" by repeating our earlier 4×4 example, in which we selected $r' = 2$, $r = 1$, $l = 3$, and $s = \mathrm{c}$:

r	a	b	c	d		a	[c]	c	d		[b]	c	c	d		b	c	c	d
r'	b	c	d	a	→	b	[b]	d	a	→	[a]	b	d	a	→	a	b	d	[c]
t	d	a	b	c		d	a	b	c		d	a	b	c		d	a	b	[a]
	c	d	a	b		c	d	a	b		c	d	a	b		c	d	a	b

If the last swap involved a third row ($t \neq r$), let row t assume the role that row r' had: swap within columns of rows t and r, beginning (equiprobably) with either column holding a duplicate symbol, and stopping when we reach a Latin square. In our example, row $t = 3$ has a's in columns 2 and 4; we pick column 2, and reach a LS immediately:

r	b	c	c	d		b	[a]	c	d
r'	a	b	d	c	→	a	b	d	c
t	d	a	b	a		d	[c]	b	a
	c	d	a	b		c	d	a	b

Our proper moves are similar to moves considered by others. We could regard the first stage of a proper move as a more general sort of cycle swap: one that swaps (within distinct columns) between an initial row and either of *two* other rows, leaving the initial row with a complete set of symbols. Our proper move allows only the *final* swap to involve a third row, but we could instead allow our pivot and alternate rows to participate *freely* in swaps with the initial row; at the end of such a cycle swap, some clean-up involving the pivot and alternate rows would again be necessary to reach a Latin square. Such moves (from the "column perspective") have been proposed by Asratyan and Mirumyan [1], but seem less manageable for Markov-chain purposes: it is not clear how to deal with the issues of weighting and reversibility. [Interestingly, the connectivity proof in [1] implicitly yields an $O(n^4)$ bound on the number of moves required to travel between two Latin squares; since our moves are a subset of theirs, a tighter bound on their "graph diameter" would

be our $4(n-1)^2$.] Pittenger's moves [9], defined from the "symbol" perspective, correspond to our two-rowed and three-rowed "$l = 2$" moves.

6. OBSERVATIONS AND CONJECTURES

Commenting on Yates' aforementioned recommendation regarding random Latin squares, D. A. Preece [4] opined pessimistically that "such a wide choice requires in practice that the latin squares of the size in hand should have been enumerated and representative squares tabulated." Instead, we have offered two Markov chains that may be used to produce random Latin squares; each has the uniform stationary distribution. We solved the connectivity problem by finding a larger space that could be connected with simple moves, and then found compositions of these moves that stay within the original space. Random moves are easy to generate and apply, especially if the simulation program maintains simultaneous "views" of the square from row-column, column-symbol, and symbol-row "perspectives." (Each view can be stored as a two-dimensional array, indexed by the corresponding dimensions; a ± 1-move can then be applied in constant time, and, for proper moves, cycles can be followed with similar efficiency.)

In order to use either of our Markov chains to generate almost-uniformly distributed Latin squares, we must know how rapidly the chain converges to the (uniform) stationary distribution. The best hope in this problem appears to be path-counting methods, as in, e.g., [5], [10]. Consider the graph in which vertices correspond to squares and edges correspond to possible transitions. For each pair of squares we must construct a (possibly randomized) connecting path. To get reasonably rapid mixing we must then show that the edges in the graph have total flows that are bounded by (informally) something like $O(n^k)L(n)$ (for some k); the smaller k is, the more rapidly the chain mixes. Recall that our proofs construct connecting paths that go directly (in some vague sense) from one square to the other; there is no obvious bottleneck, so we may have a reasonable set of paths for rapid-mixing considerations. (Of our two chains, we suspect that the "improper" one mixes more rapidly, in terms of real simulation time: executing a proper move takes time comparable to that needed to execute an equivalent sequence of ± 1-moves; substituting an equal number of random ± 1-moves seems likely to mix things up more.)

Another potential application of our Markov chains is the approximate counting of the number of order-n Latin squares. We illustrate briefly how this could be done given appropriate proof of rapid mixing. Define a *partial Latin square* (PLS) to be an array, comprising a $k \times n$ Latin rectangle ($0 \leq k < n-1$) and a partially specified $(k+1)$st row of j entries ($0 \leq j < n$), that can be extended to an $n \times n$ Latin square. Partial Latin squares are self-reducible in the sense of [10]. If one can generate a random completion of a PLS, then one can probabilistically count the number of order-n Latin squares as discussed below.

Presently, it is not clear whether either of our sets of moves connects each space of PLS completions. In constructing our paths, we occasionally made moves that momentarily disturbed the entries we had already fixed; for generating completions of a PLS, this sort of disturbance would take us outside our state space. This is not always bad (recall improper squares), but in this case dealing with an extended state space might be more difficult. There is an easier remedy, which we now sketch.

A careful reading of the proof of Lemma 2 reveals that our paths disturbed already-fixed cells of the Latin square in only one circumstance ("Case 2"). We had an improper square and wished to perform a cycle swap between two rows. To do this without creating a second -1, we got rid of the impropriety (possibly disturbing already-fixed cells), performed the cycle swap, and restored the impropriety. Here are two ways to modify the Markov chain to avoid this, First, one could allow two -1's in the incidence cube; then the cycle swap could be performed without any disturbance to the fixed elements. It would be straightforward to construct a reversible Markov chain having an appropriate stationary distribution on this enlarged state space, as we did for squares having a single impropriety. Alternatively, we could keep as the state space the set of proper and "singly improper" squares, but expand the set of legal moves: if, whether at a proper or improper square, we allowed ± 1-moves and cycle swaps, then again there would be no need to disturb the cells already fixed. Again, it would be straightforward to construct a reversible Markov chain with this move set and a useful stationary distribution. We state the following without proof, as the proof is almost exactly that of Theorem 1:

Theorem 8. *Given an order-n partial Latin square L in which $kn + j$ entries are specified, let S be the set of its completions to proper or improper squares. Construct a graph that has vertex set S and an edge connecting each pair of vertices that differ by a ± 1-move or a cycle swap. Then this graph is connected, with diameter $O(n^2(n-k))$.*

Again, the proof gives a set of paths between vertices on this graph with no obvious bottleneck. Given a mixing rate, we can use standard ideas to count Latin squares approximately: We extend a PLS cell by cell, each time estimating the proportion of completions that match the extension we choose. We do this in two stages. First, we choose a cell (in the partially specified row, if there is one); there is some symbol that occupies this cell in at least $1/n$ of the possible completions. By simulating the Markov chain and sampling it at sufficiently distant epochs, we can identify, with confidence $1-\varepsilon$, a symbol that occupies this cell in at least $1/(2n)$ of the completions. Simulating and sampling again, we can estimate the true proportion p with error $\pm\delta$ and confidence $1-\varepsilon$; the number of completions of the PLS is $1/p$ times the number of completions of the extended PLS we get by putting this symbol in the chosen cell. We can continue this until a straightforward counting problem remains [e.g., the number of two-row extensions of a (specific) $(n-2)\times n$ Latin rectangle]. Starting from scratch, or from a single-rowed Latin rectangle, we can probabilistically count the number of Latin squares of any given size by multiplying the estimated fractions. Given a polynomial mixing rate, e.g., that the dominant eigenvalue of the transition matrix is at most $1 - cn^{-k}$ for some c and k, this can be done in polynomial time, with a good probabilistic bound on the error. As we do not have a mixing-rate result, we omit the standard details.

The general problem of generating a contingency table at random from those with given marginals has been attacked by Diaconis and Sturmfels [6], using algebraic geometry. Given a particular "shape" of table, they use a computer algebra system to find a set of moves that is rich enough to connect the space (for any given marginals) without leaving it. For $2 \times m \times n$ tables, they find moves that resemble the possible cycle swaps between rows of a two-rowed Latin rectangle; if we relax the nonnega-

tivity condition to allow a single -1-cell, we can replace each of their moves by a sequence of ± 1-moves. For the $3 \times 3 \times 3$ problem, they offer a set of 110 moves; 27 of these are the ± 1-moves, and (an additional) 54 correspond to our proper moves for the order-3 LS problem. Once again, if we allow a single -1-cell, we can replace each of their moves by a sequence of ± 1-moves. Diaconis and Sturmfels [6] have no general answer for the $n \times n \times n$ problem; we have solved one instance of it here. With some slight relaxation of the cell-nonnegativity constraints, it may be that the ± 1-moves are sufficient to solve more general problems involving three-dimensional tables.

ACKNOWLEDGMENTS

We are grateful to Robert Kibler for his translation of [1].

REFERENCES

[1] A. S. Asratyan and A. N. Mirumyan, *Transformations of Latin squares* (Russian), Diskret. Mat. **2** (1990), 21–28. MR91m:05034.

[2] S. E. Bammel and J. Rothstein, *The number of* 9×9 *Latin squares,* Discrete Math. **11** (1975), 93–95.

[3] J. Dénes and A. D. Keedwell, *Latin squares and their applications,* Academic Press, New York, 1974.

[4] ———, eds., *Latin squares: New developments in the theory and applications,* North-Holland, Amsterdam, 1991.

[5] P. Diaconis and D. Stroock, *Geometric bounds for eigenvalues of Markov chains,* Ann. Appl. Probab. **1** (1991), 36–61.

[6] P. Diaconis and B. Sturmfels, *Algebraic algorithms for sampling from conditional distributions,* manuscript, 1993.

[7] B. D. McKay and E. Rogoyski, *Latin squares of order 10,* Electronic J. Combinatorics **2** (1995), N3.

[8] B. D. McKay and N. C. Wormald, *Uniform generation of random Latin rectangles,* J. Combin. Math. Combin. Comput. **9** (1991), 179–186.

[9] A. Pittenger, *Mappings of Latin squares,* manuscript, 1995.

[10] A. Sinclair, *Algorithms for random generation and counting: A Markov chain approach,* Birkhäuser, Boston, 1993.

[11] J. H. van Lint and R. M. Wilson, *A course in combinatorics,* Cambridge University Press, Cambridge, 1992.

[12] M. B. Wells, *Elements of combinatorial computing,* Pergamon, Oxford, 1971.

[13] F. Yates, *The formation of latin squares for use in field experiments,* Empire J. Exper. Agric. **1** (1933), 235–244. Reprinted in *Experimental design: Selected papers of Frank Yates;* Hafner, Darien, CT, 1970.

Received May 18, 1995
Accepted February 27, 1996