Retransmission Permutation Arrays

Douglas R. Stinson

David R. Cheriton School of Computer Science
University of Waterloo

Linear Algebraic Techniques in Combinatorics/Graph Theory
Banff International Research Station
Monday, January 31, 2011

This talk is based on joint work with Jeff Dinitz, Maura Paterson and Ruizhong Wei.

definitions

A type 1 retransmission permutation array of order n (denoted type-1 $RPA(n)$) is an $n \times n$ array, say A, in which each cell contains a symbol from the set $\{1, \ldots, n\}$, such that the following properties are satisfied:

(i) every row of A contains all n symbols, and

(ii) for $1 \leq i \leq n$, the $i \times \lceil \frac{n}{i} \rceil$ rectangle in the upper left hand corner of A contains all n symbols.

- a type 2 array is one in which property **(ii)** instead holds for rectangles in the upper right corner of A.
- a type 3 array is one in which property **(ii)** instead holds for rectangles in the lower left corner of A.
- a type 4 array is one in which property **(ii)** instead holds for rectangles in the lower right corner of A.
- An RPA is latin if every column of A contains all n symbols.

an example

A type-$1, 2, 3, 4$ $LRPA(4)$:

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

- An $r \times \lceil \frac{n}{r} \rceil$ rectangle is called basic if it does not contain an $r' \times \lceil \frac{n}{r'} \rceil$ rectangle where $r' < r$ and $\lceil \frac{n}{r} \rceil = \lceil \frac{n}{r'} \rceil$.
- In verifying property **(ii)**, it suffices to consider only basic rectangles. The basic rectangles that must be verified in the above example have dimensions 1×4, 2×2 and 4×1.

an example

A type-$1, 2, 3, 4$ $LRPA(4)$:

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

- An $r \times \lceil \frac{n}{r} \rceil$ rectangle is called basic if it does not contain an $r' \times \lceil \frac{n}{r'} \rceil$ rectangle where $r' < r$ and $\lceil \frac{n}{r} \rceil = \lceil \frac{n}{r'} \rceil$.
- In verifying property **(ii)**, it suffices to consider only basic rectangles. The basic rectangles that must be verified in the above example have dimensions 1×4, 2×2 and 4×1.

an example

A type-$1, 2, 3, 4$ $LRPA(4)$:

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

- An $r \times \lceil \frac{n}{r} \rceil$ rectangle is called basic if it does not contain an $r' \times \lceil \frac{n}{r'} \rceil$ rectangle where $r' < r$ and $\lceil \frac{n}{r} \rceil = \lceil \frac{n}{r'} \rceil$.
- In verifying property **(ii)**, it suffices to consider only basic rectangles. The basic rectangles that must be verified in the above example have dimensions 1×4, 2×2 and 4×1.

an example

A type-$1, 2, 3, 4$ $LRPA(4)$:

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

- An $r \times \lceil \frac{n}{r} \rceil$ rectangle is called basic if it does not contain an $r' \times \lceil \frac{n}{r'} \rceil$ rectangle where $r' < r$ and $\lceil \frac{n}{r} \rceil = \lceil \frac{n}{r'} \rceil$.
- In verifying property **(ii)**, it suffices to consider only basic rectangles. The basic rectangles that must be verified in the above example have dimensions 1×4, 2×2 and 4×1.

another example

A type-$1,2,3,4$ $LRPA(6)$:

1	2	3	4	5	6
4	5	6	1	2	3
3	6	5	2	1	4
2	1	4	3	6	5
5	4	1	6	3	2
6	3	2	5	4	1

another example

A type-$1, 2, 3, 4$ $LRPA(6)$:

1	2	3	4	5	6
4	5	6	1	2	3
3	6	5	2	1	4
2	1	4	3	6	5
5	4	1	6	3	2
6	3	2	5	4	1

another example

A type-$1, 2, 3, 4$ $LRPA(6)$:

1	2	3	4	5	6
4	5	6	1	2	3
3	6	5	2	1	4
2	1	4	3	6	5
5	4	1	6	3	2
6	3	2	5	4	1

another example

A type-$1,2,3,4$ $LRPA(6)$:

1	2	3	4	5	6
4	5	6	1	2	3
3	6	5	2	1	4
2	1	4	3	6	5
5	4	1	6	3	2
6	3	2	5	4	1

another example

A type-$1, 2, 3, 4$ $LRPA(6)$:

1	2	3	4	5	6
4	5	6	1	2	3
3	6	5	2	1	4
2	1	4	3	6	5
5	4	1	6	3	2
6	3	2	5	4	1

one more example

A type-$1, 2, 3, 4$ $LRPA(8)$:

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

This array satisfies two symmetry properties:

- $a_{i,j} + a_{i,n+1-j} = n + 1$.
- $a_{i,j} = \pi(a_{j,i})$, where $\pi = (1)(2\ 5)(3\ 8)(4\ 7)(6)$.

motivation

- Li, Liu, Tan, Viswanathan, and Yang published a paper entitled Retransmission $\neq$ repeat: simple retransmission permutation can resolve overlapping channel collisions (Eighth ACM Workshop on Hot Topics in Networks, 2009) in which they utilise type-$1,2$ $RPA(n)$ to resolve overlapping channel collisions.
- Suppose a message is divided into n pieces and broadcast using n consecutive groups (i.e., sets of carrier frequencies).
- Two such channels may overlap in an arbitrary number $j \leq n$ groups.
- a type-$1,2$ $RPA(n)$ gives a schedule for rebroadcasting messages in n "rounds" in such a way that all n pieces of a message are received in the minimum number of rounds, regardless of the overlap value, j.

related (?) work

- C.J. Colbourn and K.E. Heinrich. Conflict-free access to parallel memories, *Journal of Parallel and Distributed Computing* **14** (1992), 193–200. In the above paper (and other related papers), fixed sized, arbitrarily positioned rectangles in a latin square are required to contain each symbol at most once.
- R.A. Bailey, P. Cameron and R. Connelly. Sudoku, gerechte designs, resolutions, affine space, spreads, reguli, and Hamming codes, *American Mathematical Monthly*, Volume 115, Number 5, May 2008, pp 383–404. A Sudoku square is a latin square of order n, where $n = m^2$, such that it can be partitioned into n square subarrays of side m such that every one of these subarrays contains all n symbols. They are examples of gerechte designs which are used in agricultural experiments.

commentary

- It doesn't seem possible to construct RPAs by "standard" design-theoretic approaches such as difference methods, finite fields, recursive constructions, etc.
- We instead end up employing a variety of ad hoc techniques, some algorithmically based, some using graph theory, counting arguments, etc.

commentary

- It doesn't seem possible to construct RPAs by "standard" design-theoretic approaches such as difference methods, finite fields, recursive constructions, etc.
- We instead end up employing a variety of ad hoc techniques, some algorithmically based, some using graph theory, counting arguments, etc.
- It has long been observed that the study of many combinatorial problems suffers from a lack of a general theory – this is sometimes cited as evidence that combinatorics is not "deep mathematics".

commentary

- It doesn't seem possible to construct RPAs by "standard" design-theoretic approaches such as difference methods, finite fields, recursive constructions, etc.
- We instead end up employing a variety of ad hoc techniques, some algorithmically based, some using graph theory, counting arguments, etc.
- It has long been observed that the study of many combinatorial problems suffers from a lack of a general theory – this is sometimes cited as evidence that combinatorics is not "deep mathematics".
- However, from the point of someone trying to solve these problems (e.g., me), the end result seems to be that these problems are harder to solve due to the lack of applicable theory. Hmm . . .

main existence results

Table: Existence results for retransmission permutation arrays

type of RPA	existence result
type-1 $RPA(n)$	all integers $n \geq 1$
type-$1,2$ $RPA(n)$	all integers $n \geq 1$
type-$1,3$ $RPA(n)$	all integers $n \geq 1$
type-$1,4$ $RPA(n)$	all integers $n \geq 1$
type-$1,2,3,4$ $RPA(n)$	all even integers $n \geq 2$
type-$1,2,3,4$ $LRPA(n)$	even integers $n \leq 16$, $n = 36$
type-$1,2,3,4$ $LRPA(n)$	odd integers $n \leq 9$

constructing a type-1 $RPA(7)$

Suppose $n = 7$. The basic rectangles have dimensions 1×7, 2×4, 3×3, 4×2, and 7×1.
We begin by filling in the 1×7 basic rectangle:

1	2	3	4	5	6	7

constructing a type-1 $RPA(7)$

Suppose $n = 7$. The basic rectangles have dimensions 1×7, 2×4, 3×3, 4×2, and 7×1.
We begin by filling in the 1×7 basic rectangle:

1	2	3	4	5	6	7

Next, we consider the 2×4 basic rectangle. We place the symbols $5, 6, 7$ in the first three cells of the second row of this rectangle:

1	2	3	4	5	6	7
5	6	7				

constructing a type-1 $RPA(7)$ (cont.)

Now we turn to the 3×3 basic rectangle, filling in the first cell of the third row with the symbol 4:

1	2	3	4	5	6	7
5	6	7				
4						

constructing a type-1 $RPA(7)$ (cont.)

Now we turn to the 3×3 basic rectangle, filling in the first cell of the third row with the symbol 4:

1	2	3	4	5	6	7
5	6	7				
4						

Next, we look at the 4×2 basic rectangle. We have to fill in the symbols 3 and 7:

1	2	3	4	5	6	7
5	6	7				
4	3					
7						

constructing a type-1 $RPA(7)$ (cont.)

The last basic rectangle has dimensions 7×1. It is completed by filling in the symbols $2, 6$ and 3 into the first cells in the last three rows:

1	2	3	4	5	6	7
5	6	7				
4	3					
7						
2						
6						
3						

constructing a type-1 $RPA(7)$ (cont.)

Finally, we fill in all remaining cells in such a way that each row is a permutation, for example,

1	2	3	4	5	6	7
5	6	7	1	2	3	4
4	3	1	2	5	6	7
7	1	2	3	4	5	6
2	1	3	4	5	6	7
6	1	2	3	4	5	7
3	1	2	4	5	6	7

first theorem

The process described in the above example always works. Therefore we have

Theorem

For all integers $n \geq 1$*, there exists a type-*1 $RPA(n)$*.*

type-$1,2$ $RPA(n)$

- Suppose n is even.
- We consider arrays $A = (a_{i,j})$ where, for all $1 \leq i, j \leq n$, it holds that $a_{i,j} + a_{i,n+1-j} = n + 1$.
- Suppose we construct a type-1 $RPA(n)$, ensuring that after the basic rectangles have been filled in, no row contains two symbols that sum to $n + 1$ (except for the first row, which is already a permutation of the n symbols).
- Then we can easily fill in the rest of A to construct a type-$1,2$ $RPA(n)$:
  1. For every filled cell (i, j), we define $a_{i,n+1-j} = n + 1 - a_{i,j}$.
  2. At this point, no row contains any symbol more than once, so it is then a simple matter to complete each row to a permutation of the n symbols.

constructing a type-$1, 2$ $RPA(8)$

Suppose $n = 8$. The basic rectangles have dimensions 1×8, 2×4, 3×3, 4×2, and 8×1.

We begin by filling in the 1×8 basic rectangle:

1	2	3	4	5	6	7	8

constructing a type-$1, 2$ $RPA(8)$

Suppose $n = 8$. The basic rectangles have dimensions 1×8, 2×4, 3×3, 4×2, and 8×1.
We begin by filling in the 1×8 basic rectangle:

1	2	3	4	5	6	7	8

Next, we consider the 2×4 basic rectangle. We place the symbols $5, 6, 7, 8$ in the first four cells of the second row of this rectangle, noting that no two of these symbols sum to 9:

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

constructing a type-$1,2$ $RPA(8)$ **(cont.)**

Now we turn to the 3×3 basic rectangle, filling in the first two cell of the third row with the symbols 4 and 8 (note that $4 + 8 \neq 9$):

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

constructing a type-$1,2$ $RPA(8)$ (cont.)

Now we turn to the 3×3 basic rectangle, filling in the first two cell of the third row with the symbols 4 and 8 (note that $4 + 8 \neq 9$):

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

Next, we look at the 4×2 basic rectangle. We have to fill in the symbols 3 and 7 (note that $3 + 7 \neq 9$):

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

constructing a type-$1,2$ $RPA(8)$ (cont.)

The last basic rectangle has dimensions 8×1. It is completed by filling in the symbols $2, 6, 8$ and 7 into the first cells in the last four rows:

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

constructing a type-$1,2$ $RPA(8)$ (cont.)

Now, we "reflect" each row:

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

constructing a type-$1, 2$ $RPA(8)$ (cont.)

Finally, we fill in all remaining cells in such a way that each row is a permutation.

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

proving that the method works

- When copying numbers from "red" cells to "green" cells, how do we ensure that we never have two numbers in a row that sum to $n+1$?
- It is easy to avoid this in practice, but giving a proof is seemingly much more challenging.
- In the end, we used two different techniques to make the proof rigourous.
- First, we perform interchanges within rows, to ensure that there do not exist any two elements in a red cell that sum to $n+1$.
- That this is always possible can be proven using a certain "alternating path" graph-theoretic argument.

proving that the method works (cont.)

- However, this is not sufficient to complete the proof, because there may already be filled non-green cells in some row(s) containing green cells (for example, when the 4×2 rectangle is filled in in the case $n = 7$.
- We need to ensure that we never place a symbol y from a red cell into a row that already contains the symbol $x = n + 1 - y$.
- It is important to note that there is at least one completely empty row available, so there is some flexibility in where these numbers are placed.
- A complicated counting argument completes the proof (for details of the proof, see the paper!).

the main lemma

Lemma

Suppose $w > b_L \geq \cdots \geq b_1$ *are positive integers and suppose* d *is a positive integer. Denote* $b = \sum_{i=1}^{L} b_i$ *and suppose that*

$$0 \leq t \leq (L+d)w - b.$$

Suppose $B_1, \ldots, B_L$ *are pairwise disjoint sets such that* $|B_i| = b_i$ *for* $1 \leq i \leq L$. *Finally, suppose that* $|B| = t$. *Then there exists a partition*

$$B = \left(\bigcup_{i=1}^{L} C_i\right) \bigcup \left(\bigcup_{i=1}^{d} D_i\right),$$

where the following properties are satisfied:

1. $w \geq b_L + |C_L| \geq b_{L-1} + |C_{L-1}| \geq \cdots \geq b_1 + |C_1| \geq |D_1| \geq \cdots \geq |D_d|$.
2. $C_i \cap B_i = \emptyset$ *for* $1 \leq i \leq L$.

second theorem

The above-described technique can be modified to handle the case where n is odd. So we get the following

Theorem

For all integers $n \geq 1$, there exists a type-$1, 2$ $RPA(n)$.

latin RPAs

- Finding general constructions for $LRPAs$ seems to be quite difficult.
- In fact, we only have a few small examples at the present time (no infinite classes are known, even for type-1 $LRPA(n)$).
- We describe the method we used to construct type-$1, 2, 3, 4$ $LRPA(16)$ and type-$1, 2, 3, 4$ $LRPA(36)$, illustrating the technique by constructing a type-$1, 2, 3, 4$ $LRPA(16)$.

Lemma

Let $n \geq 2$ be even, and suppose there exists an $\frac{n}{2} \times \frac{n}{2}$ latin square S with the property that for all i with $2 \leq i \leq \frac{n}{2}$, the $i \times \lceil \frac{n}{i} \rceil$ rectangle in the upper left hand corner of S contains each of the symbols from 1 to $\frac{n}{2}$ at least twice. Then there exists a type-$1, 2, 3, 4$ $LRPA(n)$.

the construction

We construct a type-$1, 2, 3, 4$ $LRPA(n)$, A, from S in two stages as follows:

1. Each of the $i \times \lceil \frac{n}{i} \rceil$ rectangles in the upper left hand corner of S contains each symbol x with $1 \leq x \leq \frac{n}{2}$ twice. By considering each such rectangle in turn and using a graph colouring argument, we can replace appropriately chosen copies of x by $n+1-x$ and construct a new array S' for which each of the $i \times \lceil \frac{n}{i} \rceil$ rectangles in the upper left hand corner contain each of the symbols from 1 to n.
2. Now we let S' form the top left corner of A, and "reflect" it by applying the symmetry condition $a_{i,j} + a_{i,n+1-j} = n+1$, to fill in the top right corner of A. Finally, we carry out a similar reflection vertically to fill in the rest of A. The result is an $LRPA$ that is symmetric under rotation through 180 degrees.

example

We give an example of an 8×8 latin square S with the required properties. Note that the shaded cells are cells that are contained in basic rectangles in the resulting 16×16 latin square.

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

example (cont.)

We now adjust the entries in the top left rectangles so that each rectangle contains all the numbers from 1 to 16:

1	2	3	4	10	9	11	12
15	5	6	7	13	16	14	8
14	11	12	8	16	2	4	7
13	10	9	16	2	3	5	6
7	4	16	2	5	6	8	3
8	16	2	3	6	5	7	4
6	3	4	5	8	7	1	2
12	9	7	6	3	4	2	1

example (cont.)

Finally, we "reflect" the result to obtain a type-$1, 2, 3, 4$ $LRPA(16)$:

1	2	3	4	10	9	11	12	5	6	8	7	13	14	15	16
15	5	6	7	13	16	14	8	9	3	1	4	10	11	12	2
14	11	12	8	16	2	4	7	10	13	15	1	9	5	6	3
13	10	9	16	2	3	5	6	11	12	14	15	1	8	7	4
7	4	16	2	5	6	8	3	14	9	11	12	15	1	13	10
8	16	2	3	6	5	7	4	13	10	12	11	14	15	1	9
6	3	4	5	8	7	1	2	15	16	10	9	12	13	14	11
12	9	7	6	3	4	2	1	16	15	13	14	11	10	8	5
5	8	10	11	14	15	15	16	1	2	4	3	5	7	9	12
11	14	13	12	9	10	16	15	2	1	7	8	5	4	3	6
9	1	15	14	11	12	10	13	4	7	5	6	3	2	16	8
10	13	1	15	12	11	9	14	3	8	6	5	2	16	4	7
4	7	8	1	15	14	12	11	6	5	3	2	16	9	10	13
3	6	5	9	1	15	13	10	7	4	2	16	8	12	11	14
2	12	11	10	4	1	3	9	8	14	16	13	7	6	5	15
16	15	14	13	7	8	6	5	12	11	9	10	4	3	2	1

thank you for your attention!