

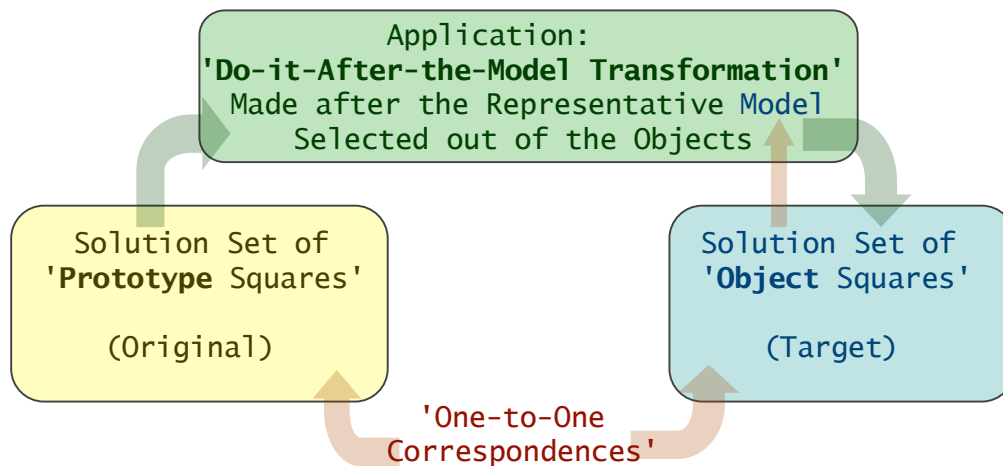
Chapter 1: Fundamental Study of 'Prototype Squares' and 'Do-it-After-the-Model Transformation': Kanji Setsuda

Section 3 : Prototype Squares of Order 5 and Reconstruction of Three Types of Magic Squares of Order 5 by DAMT

#1. Purpose of this Section

Let's study here about 'Prototype Squares' of order 5 and their structural relations with 3 types of Magic Squares of Order 5: (1) Self-complementary, (2) Pan-diagonal, and (3) Simultaneous type: both self-complementary and pan-diagonal.

**** Two Sets of Squares and 'Do-it-After-the-Model Transformation' ****



The next list shows the samples of 'Prototype Squares' of order 5 and the single application 'Do-it-After-the-Model Transformation'. They are designed to reconstruct all of our objective solutions: the Self-complementary Magic Squares of Order 5.

Apply the same 'DAMT' to each Prototype, and you should have each objective one.

**** Sample Results of the 'Do-it-After-the-Model Transformation' ****

[Prototype Squares(/P) and Self-complementary MS55 Reconstructed(/R)]

<p>1/P</p> <pre> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25</pre>	<p>(Apply the 'Do-it-After-the-Model Transformation')</p> <pre> {d[1]=c[1]; d[2]=c[23]; d[3]=c[20]; d[4]=c[12]; d[5]=c[9]; d[6]=c[15]; d[7]=c[7]; d[8]=c[4]; d[9]=c[21]; d[10]=c[18]; d[11]=c[24]; d[12]=c[16]; d[13]=c[13]; d[14]=c[10]; d[15]=c[2]; d[16]=c[8]; d[17]=c[5]; d[18]=c[22]; d[19]=c[19]; d[20]=c[11]; d[21]=c[17]; d[22]=c[14]; d[23]=c[6]; d[24]=c[3]; d[25]=c[25];}</pre>	<p>348/R</p> <pre> 1 23 20 12 9 15 7 4 21 18 24 16 13 10 2 8 5 22 19 11 17 14 6 3 25</pre>
<p>2/P</p> <pre> 1 2 3 4 5 6 8 11 14 16 9 7 13 19 17 10 12 15 18 20 21 22 23 24 25</pre>	<p>(Apply the same 'DAMT' above to the left Prototype)</p>	<p>595/R</p> <pre> 1 23 20 7 14 17 8 4 21 15 24 10 13 16 2 11 5 22 18 9 12 19 6 3 25</pre>
<p>3/P</p> <pre> 1 2 3 4 5 10 12 9 6 8 15 19 13 7 11 18 20 17 14 16 21 22 23 24 25</pre>	<p>(Apply the same 'DAMT' above to the left Prototype)</p>	<p>165/R</p> <pre> 1 23 16 19 6 11 12 4 21 17 24 18 13 8 2 9 5 22 14 15 20 7 10 3 25</pre>
<p>4/P</p> <pre> 1 2 3 4 5 10 12 15 18 20 9 7 13 19 17 6 8 11 14 16 21 22 23 24 25</pre>	<p>(Apply the same 'DAMT' above to the left Prototype)</p>	<p>856/R</p> <pre> 1 23 16 7 18 17 12 4 21 11 24 6 13 20 2 15 5 22 14 9 8 19 10 3 25</pre>

5/P				
1	2	3	4	5
16	17	18	19	20
11	12	13	14	15
6	7	8	9	10
21	22	23	24	25
.

(Apply the same 'DAMT' above to the left Prototype) →

918/R				
1	23	10	12	19
15	17	4	21	8
24	6	13	20	2
18	5	22	9	11
7	14	16	3	25
.

We want to find the complete 'One-to-one Correspondences' between two sets of Prototype Squares and of Self-complementary Magic Squares of Order 5, by our Do-it-After-the-Model Transformation, the bridge built over those two solution sets.

The next two figures below illustrate the primary forms of Prototype squares and Object solutions. The Prototype one on the left hand side is made after the regular array 5 by 5, and each value of 25 positions is set equal to its position name itself.

[Primary Form of Prototype Squares and the Model Solution]

4	5	1	2	3	4	5	1	2	12	9	1	23	20	12	9	1	23
9	10	6	7	8	9	10	6	7	21	18	15	7	4	21	18	15	7
14	15	11	12	13	14	15	11	12	10	2	24	16	13	10	2	24	16
19	20	16	17	18	19	20	16	17	19	11	8	5	22	19	11	8	5
24	25	21	22	23	24	25	21	22	3	25	17	14	6	3	25	17	14

The right figure is the primary representative solution of object squares. It should be selected not only for the 'Model Solution', but also for the purpose of getting the rules of 'DAM Transformation' after it.

This time I would like to choose this sample for the only common Model for all the three types of objects, since it originally belongs to the 'Simultaneous' type, that has the properties of both self-complementary and pan-diagonal at the same time.

#2. 'DAMT' and the Common Rules of Transformation

Do-it-After-the-Model Transformation is the way of making our object solutions from the Prototype Squares, and the rules of transformation are defined after the very Model itself and dictated as follows:

```

/* Do-it-After-the-Model Transformation */
void damt(void){
    d[1]=c[1];    d[2]=c[23];    d[3]=c[20];    d[4]=c[12];    d[5]=c[9];
    d[6]=c[15];    d[7]=c[7];    d[8]=c[4];    d[9]=c[21];    d[10]=c[18];
    d[11]=c[24];   d[12]=c[16];   d[13]=c[13];   d[14]=c[10];   d[15]=c[2];
    d[16]=c[8];    d[17]=c[5];    d[18]=c[22];   d[19]=c[19];   d[20]=c[11];
    d[21]=c[17];   d[22]=c[14];   d[23]=c[6];    d[24]=c[3];    d[25]=c[25];
}

```

Let's assume them as the only common rules for all three types, and let's apply this set of rules to each Prototype Square and get each corresponding object solution.

First of all we have to make the following three sets of Prototype Squares:

- (1) 48544 solutions for Self-complementary MS55
- (2) 3600 solutions for Pan-diagonal MS55
- (3) 16 solutions for Simultaneous MS55: both Self-complementary and Pan-diagonal.

#3. How to make the Prototype Squares for Self-Complementary MS55

At first we have to study the Model Solution and to be taught everything from it.

When we assume the next diagram as Basic Form of Prototype Squares and define such the Conditions of them as listed below for the Self-complementary Magic type, we can always get the appropriate Prototype solutions we have just wanted.

We already know how to solve these simultaneous equations, and can do that as usually as we have done many times. We should just solve the 'inverse function' of DAMT itself, I would say.

[Basic Diagram and Conditions for 'Prototype Squares 5x5']

'Prototype Squares'					The Model Solution												
4	5	1	2	3	4	5	1	2	12	9	1	23	20	12	9	1	23
9	10	6	7	8	9	10	6	7	21	18	15	7	4	21	18	15	7
14	15	11	12	13	14	15	11	12	10	2	24	16	13	10	2	24	16
19	20	16	17	18	19	20	16	17	19	11	8	5	22	19	11	8	5
24	25	21	22	23	24	25	21	22	3	25	17	14	6	3	25	17	14

*** Basic Conditions ***

- n1+n23+n20+n12+n9=C ... (1) | n1+n15+n24+n8+n17=C ... (6)
- n15+n7+n4+n21+n18=C ... (2) | n23+n7+n16+n5+n14=C ... (7)
- n24+n16+n13+n10+n2=C ... (3) | n20+n4+n13+n22+n6=C ... (8)
- n8+n5+n22+n19+n11=C ... (4) | n12+n21+n10+n19+n3=C ... (9)
- n17+n14+n6+n3+n25=C ... (5) | n9+n18+n2+n11+n25=C ... (10)

*** Self-Complementary Conditions: ***

n1+n25=n23+n3=n20+n6=n12+n14=n9+n17=n15+n11=n7+n19
 =n4+n22=n21+n5=n18+n8=n24+n2=n16+n10=n13+n13=PSM ... (cp)

*** List-forming Inequality Conditions: ***

n1<n25; n1<n9; n1<n17; and n23>n15;

Names of variables n1, n2, n3, ... , n24, n25 are just the same with the ones of primary Prototype Square. Pay your special attention to that.

#4. Apply 'DAMT' to those Prototypes and Reconstruct our Object Squares.

Let's apply the Do-it-After-the-Model Transformation to each Prototype Squares 5x5 and make it into each Self-complementary Magic Square 5x5, shall we?

The next abstract list shows the result of our transformations. Each couple has the Prototype on the left side and our reconstructed Objects on the right, accompanied with the corresponding solution numbers(/R) found in the old list of S-C MS55.

**** Reconstruction of Self-complementary Magic Squares of Order 5 ****

[Prototypes(/P) and Self-complementary MS55 Reconstructed(/R) with Sum_Checks]

1/P					348/R Rw C1					165/P					907/R Rw C1								
1	2	3	4	5	1	23	20	12	9	65	65	1	2	4	3	5	1	22	15	8	19	65	65
6	7	8	9	10	15	7	4	21	18	65	65	11	14	16	19	20	17	14	3	21	10	65	65
11	12	13	14	15	24	16	13	10	2	65	65	9	8	13	18	17	24	6	13	20	2	65	65
16	17	18	19	20	8	5	22	19	11	65	65	6	7	10	12	15	16	5	23	12	9	65	65
21	22	23	24	25	17	14	6	3	25	65	65	21	23	22	24	25	7	18	11	4	25	65	65

412/P					404/R Rw C1					524/P					493/R Rw C1								
1	2	5	3	4	1	21	19	14	10	65	65	1	2	6	3	4	1	20	19	14	11	65	65
7	8	9	10	6	15	8	3	22	17	65	65	7	21	16	11	18	9	21	3	22	10	65	65
11	14	13	12	15	24	20	13	6	2	65	65	17	14	13	12	9	24	8	13	18	2	65	65
20	16	17	18	19	9	4	23	18	11	65	65	8	15	10	5	19	16	4	23	5	17	65	65
22	23	21	24	25	16	12	7	5	25	65	65	22	23	20	24	25	15	12	7	6	25	65	65

660/P	851/R Rw C1	770/P	1030/R Rw C1
1 2 7 3 4	1 19 21 6 18 65 65	1 2 8 3 4	1 18 9 16 21 65 65
5 12 15 18 16	17 12 3 22 11 65 65	17 19 20 21 12	15 19 3 22 6 65 65
9 6 13 20 17	24 10 13 16 2 65 65	11 16 13 10 15	24 14 13 12 2 65 65
10 8 11 14 21	15 4 23 14 9 65 65	14 5 6 7 9	20 4 23 7 11 65 65
22 23 19 24 25	8 20 5 7 25 65 65	22 23 18 24 25	5 10 17 8 25 65 65
868/P	1050/R Rw C1	933/P	790/R Rw C1
1 2 9 3 7	1 17 11 14 22 65 65	1 2 10 3 4	1 16 11 20 17 65 65
15 21 20 22 18	16 21 3 19 6 65 65	15 21 19 17 8	12 21 3 22 7 65 65
10 14 13 12 16	24 8 13 18 2 65 65	14 20 13 6 12	24 18 13 8 2 65 65
8 4 6 5 11	20 7 23 5 10 65 65	18 9 7 5 11	19 4 23 5 14 65 65
19 23 17 24 25	4 12 15 9 25 65 65	22 23 16 24 25	9 6 15 10 25 65 65
988/P	867/R Rw C1	1010/P	490/R Rw C1
1 2 11 5 7	1 15 9 22 18 65 65	1 2 12 3 5	1 14 19 20 11 65 65
17 23 20 18 10	12 23 5 19 6 65 65	7 22 16 11 8	9 22 3 21 10 65 65
14 22 13 4 12	24 16 13 10 2 65 65	17 20 13 6 9	24 18 13 8 2 65 65
16 8 6 3 9	20 7 21 3 14 65 65	18 15 10 4 19	16 5 23 4 17 65 65
19 21 15 24 25	8 4 17 11 25 65 65	21 23 14 24 25	15 6 7 12 25 65 65
1072/P	797/R Rw C1	1089/P	606/R Rw C1
1 2 14 4 5	1 12 16 19 17 65 65	1 2 15 5 7	1 11 17 22 14 65 65
10 23 20 17 8	11 23 4 21 6 65 65	9 23 18 14 6	10 23 5 19 8 65 65
15 19 13 7 11	24 18 13 8 2 65 65	16 22 13 4 10	24 20 13 6 2 65 65
18 9 6 3 16	20 5 22 3 15 65 65	20 12 8 3 17	18 7 21 3 16 65 65
21 22 12 24 25	9 7 10 14 25 65 65	19 21 11 24 25	12 4 9 15 25 65 65
1104/P	623/R Rw C1	1115/P	636/R Rw C1
1 2 16 11 5	1 10 23 17 14 65 65	1 2 17 16 11	1 9 18 23 14 65 65
3 19 20 14 4	8 19 11 21 6 65 65	8 22 21 14 6	7 22 16 15 5 65 65
18 17 13 9 8	24 22 13 4 2 65 65	19 23 13 3 7	24 20 13 6 2 65 65
22 12 6 7 23	20 5 15 7 18 65 65	20 12 5 4 18	21 11 10 4 19 65 65
21 15 10 24 25	12 9 3 16 25 65 65	15 10 9 24 25	12 3 8 17 25 65 65
1119/P	1336/R Rw C1	1662/P	1993/R Rw C1
1 3 2 4 5	1 24 20 8 12 65 65	1 4 2 3 5	1 24 19 10 11 65 65
6 7 10 12 15	17 7 4 21 16 65 65	7 6 9 11 12	18 6 3 21 17 65 65
9 8 13 18 17	23 11 13 15 3 65 65	8 10 13 16 18	22 14 13 12 4 65 65
11 14 16 19 20	10 5 22 19 9 65 65	14 15 17 20 19	9 5 23 20 8 65 65
21 22 24 23 25	14 18 6 2 25 65 65	21 23 24 22 25	15 16 7 2 25 65 65
2466/P	2606/R Rw C1	2857/P	3172/R Rw C1
1 5 2 4 3	1 24 20 12 8 65 65	1 6 2 3 4	1 24 17 12 11 65 65
6 7 10 8 9	15 7 4 23 16 65 65	9 5 10 11 8	19 5 3 22 16 65 65
11 12 13 14 15	21 17 13 9 5 65 65	7 12 13 14 19	20 18 13 8 6 65 65
17 18 16 19 20	10 3 22 19 11 65 65	18 15 16 21 17	10 4 23 21 7 65 65
23 22 24 21 25	18 14 6 2 25 65 65	22 23 24 20 25	15 14 9 2 25 65 65
3479/P	3877/R Rw C1	3948/P	4109/R Rw C1
1 7 2 3 4	1 24 11 12 17 65 65	1 8 2 3 4	1 24 21 9 10 65 65
15 18 20 17 21	16 18 3 22 6 65 65	5 6 11 10 12	19 6 3 22 15 65 65
10 12 13 14 16	19 5 13 21 7 65 65	7 9 13 17 19	18 14 13 12 8 65 65
5 9 6 8 11	20 4 23 8 10 65 65	14 16 15 20 21	11 4 23 20 7 65 65
22 23 24 19 25	9 14 15 2 25 65 65	22 23 24 18 25	16 17 5 2 25 65 65
4336/P	4444/R Rw C1	4603/P	4841/R Rw C1
1 9 2 4 3	1 24 12 20 8 65 65	1 10 2 3 4	1 24 19 12 9 65 65
14 21 19 8 15	10 21 4 23 7 65 65	7 5 11 9 8	20 5 3 22 15 65 65
16 20 13 6 10	17 11 13 15 9 65 65	6 12 13 14 20	16 18 13 8 10 65 65
11 18 7 5 12	19 3 22 5 16 65 65	18 17 15 21 19	11 4 23 21 6 65 65
23 22 24 17 25	18 6 14 2 25 65 65	22 23 24 16 25	17 14 7 2 25 65 65

5067/P	5085/R Rw C1	5378/P	5728/R Rw C1
1 11 2 3 14	1 24 19 16 5 65 65	1 12 2 3 5	1 24 17 7 16 65 65
7 8 6 5 17	22 8 3 12 20 65 65	9 11 18 16 20	22 11 3 21 8 65 65
4 16 13 10 22	15 9 13 17 11 65 65	4 7 13 19 22	14 6 13 20 12 65 65
9 21 20 18 19	6 14 23 18 4 65 65	6 10 8 15 17	18 5 23 15 4 65 65
12 23 24 15 25	21 10 7 2 25 65 65	21 23 24 14 25	10 19 9 2 25 65 65
5807/P	6088/R Rw C1	6155/P	6312/R Rw C1
1 14 2 3 8	1 24 9 16 15 65 65	1 15 2 3 4	1 24 19 12 9 65 65
17 19 21 15 22	20 19 3 18 5 65 65	7 18 20 9 21	16 18 3 22 6 65 65
6 16 13 10 20	12 4 13 22 14 65 65	10 12 13 14 16	11 5 13 21 15 65 65
4 11 5 7 9	21 8 23 7 6 65 65	5 17 6 8 19	20 4 23 8 10 65 65
18 23 24 12 25	11 10 17 2 25 65 65	22 23 24 11 25	17 14 7 2 25 65 65
6394/P	6569/R Rw C1	6725/P	6782/R Rw C1
1 16 2 3 4	1 24 11 21 8 65 65	1 17 2 3 4	1 24 15 20 5 65 65
15 12 17 8 6	19 12 3 22 9 65 65	11 12 16 5 7	18 12 3 22 10 65 65
7 21 13 5 19	10 20 13 6 16 65 65	8 20 13 6 18	9 19 13 7 17 65 65
20 18 9 14 11	17 4 23 14 7 65 65	19 21 10 14 15	16 4 23 14 8 65 65
22 23 24 10 25	18 5 15 2 25 65 65	22 23 24 9 25	21 6 11 2 25 65 65
6937/P	7009/R Rw C1	7139/P	7201/R Rw C1
1 18 2 3 4	1 24 15 20 5 65 65	1 19 2 3 8	1 24 15 20 5 65 65
11 17 19 5 12	16 17 3 22 7 65 65	11 10 14 5 9	22 10 3 18 12 65 65
10 20 13 6 16	8 14 13 12 18 65 65	4 20 13 6 22	7 17 13 9 19 65 65
14 21 7 9 15	19 4 23 9 10 65 65	17 21 12 16 15	14 8 23 16 4 65 65
22 23 24 8 25	21 6 11 2 25 65 65	18 23 24 7 25	21 6 11 2 25 65 65
7282/P	7390/R Rw C1	7426/P	7430/R Rw C1
1 20 2 3 4	1 24 15 17 8 65 65	1 21 2 4 9	1 24 18 19 3 65 65
11 12 19 8 10	21 12 3 22 7 65 65	8 14 16 3 15	20 14 4 17 10 65 65
5 17 13 9 21	6 16 13 10 20 65 65	6 19 13 7 20	5 11 13 15 21 65 65
16 18 7 14 15	19 4 23 14 5 65 65	11 23 10 12 18	16 9 22 12 6 65 65
22 23 24 6 25	18 9 11 2 25 65 65	17 22 24 5 25	23 7 8 2 25 65 65
7519/P	7639/R Rw C1	7672/P	7693/R Rw C1
1 22 2 5 6	1 24 17 15 8 65 65	1 23 2 6 11	1 24 16 19 5 65 65
9 10 19 8 12	23 10 5 20 7 65 65	10 14 18 5 17	22 14 6 15 8 65 65
3 15 13 11 23	4 14 13 12 22 65 65	4 19 13 7 22	3 9 13 17 23 65 65
14 18 7 16 17	19 6 21 16 3 65 65	9 21 8 12 16	18 11 20 12 4 65 65
20 21 24 4 25	18 11 9 2 25 65 65	15 20 24 3 25	21 7 10 2 25 65 65
7793/P	8025/R Rw C1	14601/P	14853/R Rw C1
2 1 3 4 5	2 23 20 12 8 65 65	3 1 2 4 5	3 24 20 10 8 65 65
6 16 11 8 19	9 16 4 21 15 65 65	6 9 7 8 15	12 9 4 21 19 65 65
17 12 13 14 9	25 7 13 19 1 65 65	14 10 13 16 12	25 11 13 15 1 65 65
7 18 15 10 20	11 5 22 10 17 65 65	11 18 19 17 20	7 5 22 17 14 65 65
21 22 23 25 24	18 14 6 3 24 65 65	21 22 24 25 23	18 16 6 2 23 65 65
21805/P	21908/R Rw C1	27597/P	27769/R Rw C1
4 1 2 3 5	4 24 19 10 8 65 65	5 1 2 3 4	5 24 11 16 9 65 65
7 9 6 8 15	12 9 3 21 20 65 65	15 8 6 9 7	12 8 3 22 20 65 65
14 10 13 16 12	25 11 13 15 1 65 65	14 16 13 10 12	25 19 13 7 1 65 65
11 18 20 17 19	6 5 23 17 14 65 65	19 17 20 18 11	6 4 23 18 14 65 65
21 23 24 25 22	18 16 7 2 22 65 65	22 23 24 25 21	17 10 15 2 21 65 65
33193/P	33316/R Rw C1	37313/P	37447/R Rw C1
6 1 2 4 3	6 24 18 5 12 65 65	7 1 2 3 4	7 24 17 6 11 65 65
8 10 9 12 19	11 10 4 23 17 65 65	9 12 8 11 21	10 12 3 22 18 65 65
15 5 13 21 11	25 7 13 19 1 65 65	16 6 13 20 10	25 5 13 21 1 65 65
7 14 17 16 18	9 3 22 16 15 65 65	5 15 18 14 17	8 4 23 14 16 65 65
23 22 24 25 20	14 21 8 2 20 65 65	22 23 24 25 19	15 20 9 2 19 65 65

3165/P	2899/R1	3174/R2	3287/R3	?/R4
1 6 5 4 3	1 21 24 16 3	1 21 24 8 11	1 21 24 4 15	1 21 24 18 7
2 7 12 11 10	17 7 8 11 22	17 7 4 23 14	17 7 8 23 10	17 11 6 23 14
9 8 13 18 17	20 12 13 14 6	20 16 13 10 6	20 12 13 14 6	22 16 13 10 4
16 15 14 19 24	4 15 18 19 9	12 3 22 19 9	16 3 18 19 9	12 3 20 15 9
23 22 21 20 25	23 10 2 5 25	15 18 2 5 25	11 22 2 5 25	19 8 2 5 25
3470/P	3194/R1	3247/R2	3272/R3	?/R4
1 6 16 21 11	1 10 19 24 11	1 10 19 23 12	1 10 19 21 14	1 10 19 3 17
7 17 22 12 2	8 17 23 12 5	8 17 21 15 4	8 17 23 15 2	8 12 6 15 4
18 23 13 3 8	20 22 13 4 6	20 24 13 2 6	20 22 13 4 6	5 24 13 2 21
24 14 4 9 19	21 14 3 9 18	22 11 5 9 18	24 11 3 9 18	22 11 20 14 18
15 5 10 20 25	15 2 7 16 25	14 3 7 16 25	12 5 7 16 25	9 23 7 16 25
6653/P	6655/R1	6684/R2	6696/R3	?/R4
1 16 6 21 11	1 20 9 24 11	1 20 9 23 12	1 20 9 21 14	1 20 9 3 7
17 7 22 12 2	18 7 23 12 5	18 7 21 15 4	18 7 23 15 2	18 12 16 15 4
8 23 13 3 18	10 22 13 4 16	10 24 13 2 16	10 22 13 4 16	5 24 13 2 21
24 14 4 19 9	21 14 3 19 8	22 11 5 19 8	24 11 3 19 8	22 11 10 14 8
15 5 20 10 25	15 2 17 6 25	14 3 17 6 25	12 5 17 6 25	19 23 17 6 25
8098/P	7796/R1	8002/R2	8475/R3	?/R4
2 1 5 4 3	2 21 20 19 3	2 21 20 14 8	2 21 20 4 18	2 21 20 12 10
6 10 9 8 7	11 10 14 8 22	11 10 4 23 17	11 10 14 23 7	11 8 1 23 17
15 14 13 12 11	25 9 13 17 1	25 19 13 7 1	25 9 13 17 1	22 19 13 7 4
19 18 17 16 20	4 18 12 16 15	9 3 22 16 15	19 3 12 16 15	9 3 25 18 15
23 22 21 25 24	23 7 6 5 24	18 12 6 5 24	8 22 6 5 24	16 14 6 5 24
8731/P	7966/R1	8120/R2	8393/R3	?/R4
2 1 15 14 7	2 11 22 23 7	2 11 22 20 10	2 11 22 14 16	2 11 22 6 18
4 18 17 10 3	5 18 20 10 12	5 18 14 19 9	5 18 20 19 3	5 10 1 19 9
21 20 13 6 5	25 17 13 9 1	25 23 13 3 1	25 17 13 9 1	12 23 13 3 14
23 16 9 8 22	14 16 6 8 21	17 7 12 8 21	23 7 6 8 21	17 7 25 16 21
19 12 11 25 24	19 3 4 15 24	16 6 4 15 24	10 12 4 15 24	8 20 4 15 24
8775/P	8799/R1	8940/R2	9199/R3	?/R4
2 3 1 4 5	2 25 16 17 5	2 25 16 14 8	2 25 16 4 18	2 25 16 12 6
10 6 7 8 9	15 6 14 8 22	15 6 4 21 19	15 6 14 21 9	15 8 3 21 19
11 14 13 12 15	23 7 13 19 3	23 17 13 9 3	23 7 13 19 3	22 17 13 9 4
17 18 19 20 16	4 18 12 20 11	7 5 22 20 11	17 5 12 20 11	7 5 23 18 11
21 22 25 23 24	21 9 10 1 24	18 12 10 1 24	8 22 10 1 24	20 14 10 1 24
9857/P	9857/R1	10000/R2	10228/R3	?/R4
2 5 1 4 3	2 25 16 19 3	2 25 16 14 8	2 25 16 4 18	2 25 16 12 6
10 6 9 8 7	15 6 14 8 22	15 6 4 23 17	15 6 14 23 7	15 8 5 23 17
11 14 13 12 15	21 9 13 17 5	21 19 13 7 5	21 9 13 17 5	22 19 13 7 4
19 18 17 20 16	4 18 12 20 11	9 3 22 20 11	19 3 12 20 11	9 3 21 18 11
23 22 25 21 24	23 7 10 1 24	18 12 10 1 24	8 22 10 1 24	20 14 10 1 24
16472/P	7552/R1	16377/R2	16778/R3	?/R4
3 4 5 6 1	1 24 16 21 3	3 21 16 18 7	3 21 16 6 19	3 21 16 8 11
10 11 12 7 2	20 7 18 11 9	9 11 6 25 14	9 11 18 25 2	9 7 4 25 14
17 18 13 8 9	4 14 13 12 22	22 24 13 2 4	22 12 13 14 4	20 24 13 2 6
24 19 14 15 16	17 15 8 19 6	12 1 20 15 17	24 1 8 15 17	12 1 22 19 17
25 20 21 22 23	23 5 10 2 25	19 8 10 5 23	7 20 10 5 23	15 18 10 5 23
21979/P	17884/R1	21917/R2	22309/R3	?/R4
4 1 5 2 3	3 24 1 15 22	4 21 20 12 8	4 21 20 2 18	4 21 20 14 10
6 10 7 8 9	17 8 19 16 5	11 10 2 23 19	11 10 12 23 9	11 8 1 23 19
15 12 13 14 11	20 12 13 14 6	25 17 13 9 1	25 7 13 19 1	24 17 13 9 2
17 18 19 16 20	21 10 7 18 9	7 3 24 16 15	17 3 14 16 15	7 3 25 18 15
23 24 21 25 22	4 11 25 2 23	18 14 6 5 22	8 24 6 5 22	16 12 6 5 22
23237/P	3453/R1	23205/R2	23472/R3	?/R4
4 3 5 2 1	1 24 3 15 22	4 21 20 12 8	4 21 20 2 18	4 21 20 14 10
6 10 9 8 7	19 8 17 16 5	11 10 2 25 17	11 10 12 25 7	11 8 3 25 17
15 12 13 14 11	20 12 13 14 6	23 19 13 7 3	23 9 13 17 3	24 19 13 7 2
19 18 17 16 20	21 10 9 18 7	9 1 24 16 15	19 1 14 16 15	9 1 23 18 15
25 24 21 23 22	4 11 23 2 25	18 14 6 5 22	8 24 6 5 22	16 12 6 5 22

23573/P	19316/R1	23688/R2	23961/R3	?/R4
4 5 1 2 3	3 24 5 11 22	4 25 16 12 8	4 25 16 2 18	4 25 16 14 6
10 6 7 8 9	17 8 19 20 1	15 6 2 23 19	15 6 12 23 9	15 8 5 23 19
11 12 13 14 15	16 12 13 14 10	21 17 13 9 5	21 7 13 19 5	24 17 13 9 2
17 18 19 20 16	25 6 7 18 9	7 3 24 20 11	17 3 14 20 11	7 3 21 18 11
23 24 25 21 22	4 15 21 2 23	18 14 10 1 22	8 24 10 1 22	20 12 10 1 22
28229/P	18154/R1	28234/R2	28604/R3	?/R4
5 2 4 1 3	3 25 2 14 21	5 22 19 11 8	5 22 19 1 18	5 22 19 15 9
7 9 6 8 10	16 8 20 17 4	12 9 1 23 20	12 9 11 23 10	12 8 2 23 20
14 11 13 15 12	19 11 13 15 7	24 16 13 10 2	24 6 13 20 2	25 16 13 10 1
16 18 20 17 19	22 9 6 18 10	6 3 25 17 14	16 3 15 17 14	6 3 24 18 14
23 25 22 24 21	5 12 24 1 23	18 15 7 4 21	8 25 7 4 21	17 11 7 4 21
29268/P	18981/R1	29346/R2	29686/R3	?/R4
5 4 2 1 3	3 25 4 12 21	5 24 17 11 8	5 24 17 1 18	5 24 17 15 7
9 7 6 8 10	16 8 20 19 2	14 7 1 23 20	14 7 11 23 10	14 8 4 23 20
12 11 13 15 14	17 11 13 15 9	22 16 13 10 4	22 6 13 20 4	25 16 13 10 1
16 18 20 19 17	24 7 6 18 10	6 3 25 19 12	16 3 15 19 12	6 3 22 18 12
23 25 24 22 21	5 14 22 1 23	18 15 9 2 21	8 25 9 2 21	19 11 9 2 21
29320/P	29648/R1	29347/R2	29688/R3	?/R4
5 4 2 15 17	5 24 3 16 17	5 24 3 25 8	5 24 3 15 18	5 24 3 1 7
23 7 6 8 10	14 7 25 8 11	14 7 15 9 20	14 7 25 9 10	14 8 4 9 20
12 25 13 1 14	22 6 13 20 4	22 16 13 10 4	22 6 13 20 4	11 16 13 10 15
16 18 20 19 3	15 18 1 19 12	6 17 11 19 12	16 17 1 19 12	6 17 22 18 12
9 11 24 22 21	9 10 23 2 21	18 1 23 2 21	8 11 23 2 21	19 25 23 2 21
29370/P	5046/R1	29296/R2	29733/R3	28292/R4
5 4 3 2 1	1 24 4 15 21	5 23 16 14 7	5 23 16 2 19	5 23 16 12 9
10 9 8 7 6	20 7 18 17 3	11 9 2 25 18	11 9 14 25 6	11 7 4 25 18
15 14 13 12 11	16 14 13 12 10	22 20 13 6 4	22 8 13 18 4	24 20 13 6 2
20 19 18 17 16	23 9 8 19 6	8 1 24 17 15	20 1 12 17 15	8 1 22 19 15
25 24 23 22 21	5 11 22 2 25	19 12 10 3 21	7 24 10 3 21	17 14 10 3 21
30908/P	30776/R1	30792/R2	30814/R3	?/R4
5 9 18 22 11	5 8 16 25 11	5 8 16 24 12	5 8 16 22 14	5 8 16 2 19
10 19 23 12 1	6 19 24 12 4	6 19 22 15 3	6 19 24 15 1	6 12 9 15 3
20 24 13 2 6	17 23 13 3 9	17 25 13 1 9	17 23 13 3 9	4 25 13 1 22
25 14 3 7 16	22 14 2 7 20	23 11 4 7 20	25 11 2 7 20	23 11 17 14 20
15 4 8 17 21	15 1 10 18 21	14 2 10 18 21	12 4 10 18 21	7 24 10 18 21
32651/P	32632/R1	32642/R2	32656/R3	?/R4
5 19 8 22 11	5 18 6 25 11	5 18 6 24 12	5 18 6 22 14	5 18 6 2 9
20 9 23 12 1	16 9 24 12 4	16 9 22 15 3	16 9 24 15 1	16 12 19 15 3
10 24 13 2 16	7 23 13 3 19	7 25 13 1 19	7 23 13 3 19	4 25 13 1 22
25 14 3 17 6	22 14 2 17 10	23 11 4 17 10	25 11 2 17 10	23 11 7 14 10
15 4 18 7 21	15 1 20 8 21	14 2 20 8 21	12 4 20 8 21	17 24 20 8 21
33532/P	33293/R1	33337/R2	33391/R3	?/R4
6 1 21 16 11	6 5 24 19 11	6 5 24 18 12	6 5 24 16 14	4 18 2 21 20
2 22 17 12 7	3 22 18 12 10	3 22 16 15 9	3 22 18 15 7	17 11 25 14 23
23 18 13 8 3	25 17 13 9 1	25 19 13 7 1	25 17 13 9 1	10 19 13 7 16
19 14 9 4 24	16 14 8 4 23	17 11 10 4 23	19 11 8 4 23	3 12 1 15 9
15 10 5 25 20	15 7 2 21 20	14 8 2 21 20	12 10 2 21 20	6 5 24 8 22
35676/P	31314/R1	35710/R2	35733/R3	?/R4
6 11 1 16 21	5 17 22 1 20	6 25 4 18 12	6 25 4 16 14	2 19 16 3 20
22 2 7 12 17	16 14 8 24 3	23 2 16 5 19	23 2 18 5 17	8 5 17 14 1
3 18 13 8 23	15 7 13 19 11	15 9 13 17 11	15 7 13 19 11	4 11 13 15 22
9 14 19 24 4	23 2 18 12 10	7 21 10 24 3	9 21 8 24 3	25 12 9 21 18
5 10 25 15 20	6 25 4 9 21	14 8 22 1 20	12 10 22 1 20	6 23 10 7 24
36813/P	36839/R1	36847/R2	36877/R3	?/R4
6 21 1 16 11	6 25 4 19 11	6 25 4 18 12	6 25 4 16 14	2 9 16 3 20
22 2 17 12 7	23 2 18 12 10	23 2 16 15 9	23 2 18 15 7	8 15 7 14 1
3 18 13 8 23	5 17 13 9 21	5 19 13 7 21	5 17 13 9 21	4 21 13 5 22
19 14 9 24 4	16 14 8 24 3	17 11 10 24 3	19 11 8 24 3	25 12 19 11 18
15 10 25 5 20	15 7 22 1 20	14 8 22 1 20	12 10 22 1 20	6 23 10 17 24

```

      46948/P      46858/R1      46890/R2      46929/R3      ?/R4
10  4 23 17 11  10  3 21 20 11  10  3 21 19 12  10  3 21 17 14  2 19  5 23 16
 5 24 18 12  6   1 24 19 12  9   1 24 17 15  8   1 24 19 15  6  18 11 22 14 25
25 19 13  7  1  22 18 13  8  4   22 20 13  6  4   22 18 13  8  4   9 20 13  6 17
20 14  8  2 21  17 14  7  2 25  18 11  9  2 25  20 11  7  2 25  1 12  4 15  8
15  9  3 22 16  15  6  5 23 16  14  7  5 23 16  12  9  5 23 16  10  3 21  7 24

      47956/P      392/R1      47954/R2      47970/R3      ?/R4
10 24  3  7  1   1 20 11 23 10  10 23 11  9 12  10 23 11  7 14  4 17 11 23 10
15  4 18 12  6   19 12  9  4 21  21  4  7 25  8   21  4  9 25  6   8 25 24 12 21
 5  9 13 17 21  24  8 13 18  2   2 20 13  6 24   2 18 13  8 24   7  6 13 20 19
20 14  8 22 11   5 22 17 14  7   18  1 19 22  5   20  1 17 22  5   5 14  2  1 18
25 19 23  2 16  16  3 15  6 25  14 17 15  3 16  12 19 15  3 16  16  3 15  9 22

      47966/P      47936/R1      47955/R2      47971/R3      ?/R4
10 24  3 17 11  10 23  1 20 11  10 23  1 19 12  10 23  1 17 14  4  8 17  5 16
25  4 18 12  6   21  4 19 12  9   21  4 17 15  8   21  4 19 15  6   7 15  6 14  3
 5 19 13  7 21   2 18 13  8 24   2 20 13  6 24   2 18 13  8 24   1 24 13  2 25
20 14  8 22  1   17 14  7 22  5   18 11  9 22  5   20 11  7 22  5   23 12 20 11 19
15  9 23  2 16  15  6 25  3 16  14  7 25  3 16  12  9 25  3 16  10 21  9 18 22
. . . . .

[Reconstruction ERRORS: 48517/R1, 0/R2, 48378/R3, 48435/R4]
[Count = 48544]

```

I took the same Prototype set and did not change anything about it, but prepared 4 different Models and 4 different sets of DAMT rules according to them. I applied those DAMT to the same Prototype solutions one after another and tried to get the Object solutions. The abstract list above shows the results of my recent calculation.

Though I showed only the successful samples as many as possible in the list above, there are really many errors found in reconstruction after other Models except the Model #348, more than 99% of results fail to be the Object solutions.

When I tested it between the Primitive sets of both Prototypes and Object solutions, I had to meet the same kind of errors for other Models except #348, too many.

Only in the case of Model #348 all the results prove to be the Self-complementary MS55. It is because the Prototype solutions themselves are made after the very Model #348 and One-to-one Correspondences are realized only between the two sets.

If you want to have all the reconstructions successfully for every Model, you should prepare the new Prototype sets making after the new Models, each by each.

It means the Prototype set is actually dependent on the Model and subordinate to it.

#6. Case of Pan-diagonal Magic Squares of Order 5

Let's go on to study about Prototype Squares of Order 5 and Do-it-After-the-Model Transformation for the next case now: 'Pan-diagonal' Magic Squares of Order 5.

You know how to define that type, don't you? I prepared the basic things as listed below. You may also know there are 3600 Standard Solutions of that type.

```

/* Make Pan-diagonal Magic Squares of Order 5 */
/** Basic Form: **

  2  3  4  5 | 1| 2| 3| 4| 5| 1  2  3  4   | 1|23|20|12| 9|
  7  8  9 10 | 6| 7| 8| 9|10| 6  7  8  9   | 15| 7| 4|21|18|
12 13 14 15 |11|12|13|14|15|11 12 13 14   | 24|16|13|10| 2|
17 18 19 20 |16|17|18|19|20|16 17 18 19   |  8| 5|22|19|11|
22 23 24 25 |21|22|23|24|25|21 22 23 24   |17|14| 6| 3|25|

** Basic Conditions: C=65 **

```

```

n1+n2+n3+n4+n5=C      ...rw1; | n1+n6+n11+n16+n21=C    ...c11;
n6+n7+n8+n9+n10=C    ...rw2; | n2+n7+n12+n17+n22=C    ...c12;
n11+n12+n13+n14+n15=C ...rw3; | n3+n8+n13+n18+n23=C    ...c13;
n16+n17+n18+n19+n20=C ...rw4; | n4+n9+n14+n19+n24=C    ...c14;
n21+n22+n23+n24+n25=C ...rw5; | n5+n10+n15+n20+n25=C   ...c15;
** Pan-diagonal Conditions: C=65 **
n1+n7+n13+n19+n25=C   ...pd1; | n1+n10+n14+n18+n22=C   ...pb1;
n2+n8+n14+n20+n21=C   ...pd2; | n2+n6+n15+n19+n23=C   ...pb2;
n3+n9+n15+n16+n22=C   ...pd3; | n3+n7+n11+n20+n24=C   ...pb3;
n4+n10+n11+n17+n23=C  ...pd4; | n4+n8+n12+n16+n25=C   ...pb4;
n5+n6+n12+n18+n24=C   ...pd5; | n5+n9+n13+n17+n21=C   ...pb5;
** List-forming Inequality Conditions **
n1<n5; n1<n21; n1<n25; and n2>n6;
/**/

```

The next abstract list shows the result of my recent calculation for PDMS55.

[Standard Solutions of Pan-diagonal Magic Squares of Order 5: with Check-Sums]

1/Rw P1\ -----		37/Rw P1\ -----		73/Rw P1\ -----		109/Rw P1\ -----	
1 23 20 12 9 65 65\ 15 7 4 21 18 65 65\ 24 16 13 10 2 65 65\ 8 5 22 19 11 65 65\ 17 14 6 3 25 65 65\ -----	1 23 19 12 10 65 65\ 14 7 5 21 18 65 65\ 25 16 13 9 2 65 65\ 8 4 22 20 11 65 65\ 17 15 6 3 24 65 65\ -----	1 24 18 12 10 65 65\ 13 7 5 21 19 65 65\ 25 16 14 8 2 65 65\ 9 3 22 20 11 65 65\ 17 15 6 4 23 65 65\ -----	1 24 17 13 10 65 65\ 12 8 5 21 19 65 65\ 25 16 14 7 3 65 65\ 9 2 23 20 11 65 65\ 18 15 6 4 22 65 65\ -----	-----	-----	-----	-----
65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	-----	-----	-----	-----
145/Rw P1\ -----		181/Rw P1\ -----		217/Rw P1\ -----		253/Rw P1\ -----	
1 18 25 12 9 65 65\ 15 7 4 16 23 65 65\ 19 21 13 10 2 65 65\ 8 5 17 24 11 65 65\ 22 14 6 3 20 65 65\ -----	1 18 24 12 10 65 65\ 14 7 5 16 23 65 65\ 20 21 13 9 2 65 65\ 8 4 17 25 11 65 65\ 22 15 6 3 19 65 65\ -----	1 19 23 12 10 65 65\ 13 7 5 16 24 65 65\ 20 21 14 8 2 65 65\ 9 3 17 25 11 65 65\ 22 15 6 4 18 65 65\ -----	1 19 22 13 10 65 65\ 12 8 5 16 24 65 65\ 20 21 14 7 3 65 65\ 9 2 18 25 11 65 65\ 23 15 6 4 17 65 65\ -----	-----	-----	-----	-----
65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	-----	-----	-----	-----
289/Rw P1\ -----		325/Rw P1\ -----		361/Rw P1\ -----		397/Rw P1\ -----	
1 20 14 8 22 65 65\ 13 7 21 5 19 65 65\ 25 4 18 12 6 65 65\ 17 11 10 24 3 65 65\ 9 23 2 16 15 65 65\ -----	1 19 15 8 22 65 65\ 13 7 21 4 20 65 65\ 24 5 18 12 6 65 65\ 17 11 9 25 3 65 65\ 10 23 2 16 14 65 65\ -----	1 18 15 9 22 65 65\ 14 7 21 3 20 65 65\ 23 5 19 12 6 65 65\ 17 11 8 25 4 65 65\ 10 24 2 16 13 65 65\ -----	1 17 15 9 23 65 65\ 14 8 21 2 20 65 65\ 22 5 19 13 6 65 65\ 18 11 7 25 4 65 65\ 10 24 3 16 12 65 65\ -----	-----	-----	-----	-----
65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	-----	-----	-----	-----
433/Rw P1\ -----		469/Rw P1\ -----		505/Rw P1\ -----		541/Rw P1\ -----	
1 20 9 13 22 65 65\ 8 12 21 5 19 65 65\ 25 4 18 7 11 65 65\ 17 6 15 24 3 65 65\ 14 23 2 16 10 65 65\ -----	1 19 10 13 22 65 65\ 8 12 21 4 20 65 65\ 24 5 18 7 11 65 65\ 17 6 14 25 3 65 65\ 15 23 2 16 9 65 65\ -----	1 18 10 14 22 65 65\ 9 12 21 3 20 65 65\ 23 5 19 7 11 65 65\ 17 6 13 25 4 65 65\ 15 24 2 16 8 65 65\ -----	1 17 10 14 23 65 65\ 9 13 21 2 20 65 65\ 22 5 19 8 11 65 65\ 18 6 12 25 4 65 65\ 15 24 3 16 7 65 65\ -----	-----	-----	-----	-----
65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	-----	-----	-----	-----
577/Rw P1\ -----		1153/Rw P1\ -----		1729/Rw P1\ -----		2305/Rw P1\ -----	
2 23 20 11 9 65 65\ 15 6 4 22 18 65 65\ 24 17 13 10 1 65 65\ 8 5 21 19 12 65 65\ 16 14 7 3 25 65 65\ -----	3 22 20 11 9 65 65\ 15 6 4 23 17 65 65\ 24 18 12 10 1 65 65\ 7 5 21 19 13 65 65\ 16 14 8 2 25 65 65\ -----	4 22 20 11 8 65 65\ 15 6 3 24 17 65 65\ 23 19 12 10 1 65 65\ 7 5 21 18 14 65 65\ 16 13 9 2 25 65 65\ -----	5 22 19 11 8 65 65\ 14 6 3 25 17 65 65\ 23 20 12 9 1 65 65\ 7 4 21 18 15 65 65\ 16 13 10 2 24 65 65\ -----	-----	-----	-----	-----
65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	65 65 65 65 65 Co1umn /65/65/65/65/65 P2Sums	-----	-----	-----	-----

```

      2881/|Rw|P1\      3025/|Rw|P1\      3169/|Rw|P1\      3313/|Rw|P1\
  6 23 20  2 14|65|65\  7 23 20  1 14|65|65\  8 22 20  1 14|65|65\  9 22 20  1 13|65|65\
  5 12  9 21 18|65|65\  5 11  9 22 18|65|65\  5 11  9 23 17|65|65\  5 11  8 24 17|65|65\
 24 16  3 15  7|65|65\ 24 17  3 15  6|65|65\ 24 18  2 15  6|65|65\ 23 19  2 15  6|65|65\
 13 10 22 19  1|65|65\ 13 10 21 19  2|65|65\ 12 10 21 19  3|65|65\ 12 10 21 18  4|65|65\
 17  4 11  8 25|65|65\ 16  4 12  8 25|65|65\ 16  4 13  7 25|65|65\ 16  3 14  7 25|65|65\
-----+-----+-----+-----+
|65|65|65|65|65|CoTumn |65|65|65|65|65|CoTumn |65|65|65|65|65|CoTumn |65|65|65|65|65|CoTumn
/65/65/65/65/65|P2Sums /65/65/65/65/65|P2Sums /65/65/65/65/65|P2Sums /65/65/65/65/65|P2Sums

      3457/|Rw|P1\
 10 22 19  1 13|65|65\
   4 11  8 25 17|65|65\
 23 20  2 14  6|65|65\
 12  9 21 18  5|65|65\
 16  3 15  7 24|65|65\
-----+-----
|65|65|65|65|65|CoTumn
/65/65/65/65/65|P2Sums . . . . . [Count = 3600]

```

Let's try to reconstruct all those 3600 Pan-diagonal Magic Squares of Order 5 above, only by the solution set of Prototype Squares and our DAM Transformation.

What and how do we have to do to make the Prototype Squares for them, then?

[Basic Diagrams and Conditions for Prototype Squares 5x5]

'Prototype Squares'										The Model Solution																			
24	25	21	22	23	24	25	21	22		3	25	17	14	6	3	25	17	14											
4	5		1		2		3		4		5		1	2	12	9		1		23		20		12		9		1	23
9	10		6		7		8		9		10		6	7	21	18		15		7		4		21		18		15	7
14	15		11		12		13		14		15		11	12	10	2		24		16		13		10		2	24	16	
19	20		16		17		18		19		20		16	17	19	11		8		5		22		19		11		8	5
24	25		21		22		23		24		25		21	22	3	25		17		14		6		3		25		17	14
4	5	1	2	3	4	5	1	2		12	9	1	23	20	12	9	1	23											

* Basic Conditions *

- | | | | | |
|------------------------|---------|--|------------------------|----------|
| $n1+n23+n20+n12+n9=C$ | ... (1) | | $n1+n15+n24+n8+n17=C$ | ... (6) |
| $n15+n7+n4+n21+n18=C$ | ... (2) | | $n23+n7+n16+n5+n14=C$ | ... (7) |
| $n24+n16+n13+n10+n2=C$ | ... (3) | | $n20+n4+n13+n22+n6=C$ | ... (8) |
| $n8+n5+n22+n19+n11=C$ | ... (4) | | $n12+n21+n10+n19+n3=C$ | ... (9) |
| $n17+n14+n6+n3+n25=C$ | ... (5) | | $n9+n18+n2+n11+n25=C$ | ... (10) |

* Pan-diagonal Conditions: *

- | | | | | |
|------------------------|----------|--|------------------------|----------|
| $n1+n7+n13+n19+n25=C$ | ... (11) | | $n1+n18+n10+n22+n14=C$ | ... (12) |
| $n23+n4+n10+n11+n17=C$ | ... (13) | | $n23+n15+n2+n19+n6=C$ | ... (14) |
| $n20+n21+n2+n8+n14=C$ | ... (15) | | $n20+n7+n24+n11+n3=C$ | ... (16) |
| $n12+n18+n24+n5+n6=C$ | ... (17) | | $n12+n4+n16+n8+n25=C$ | ... (18) |
| $n9+n15+n16+n22+n3=C$ | ... (19) | | $n9+n21+n13+n5+n17=C$ | ... (20) |

* List-forming Inequality Conditions: *

- $n1 < n25$; $n1 < n9$; $n1 < n17$; and $n23 > n15$... (1fc)

Self-complementary conditions are no more necessary at all this time, since all the 25 pairs are not requested to place symmetrically in the Pan-diagonal object, although the Model itself is selected from the Simultaneous type.

Let's do the same jobs as before: (1) Make all the Prototype Squares 5x5, (2) Apply our DAMT to them one after another, and (3) Get every solution of Pan-diagonal Magic Squares 5x5 reconstructed, shall we?

145/P	9/R1	5/R2	35/R3	31/R4
1 4 2 3 5	1 22 20 14 8	1 22 20 13 9	1 22 10 14 18	1 22 10 13 19
6 9 7 8 10	15 9 3 21 17	15 8 4 21 17	15 19 3 21 7	15 18 4 21 7
11 14 12 13 15	23 16 12 10 4	24 16 12 10 3	23 6 12 20 4	24 6 12 20 3
16 19 17 18 20	7 5 24 18 11	7 5 23 19 11	17 5 24 8 11	17 5 23 9 11
21 24 22 23 25	19 13 6 2 25	18 14 6 2 25	9 13 16 2 25	8 14 16 2 25
217/P	45/R1	41/R2	71/R3	67/R4
1 5 2 3 4	1 22 19 15 8	1 22 19 13 10	1 22 9 15 18	1 22 9 13 20
6 10 7 8 9	14 10 3 21 17	14 8 5 21 17	14 20 3 21 7	14 18 5 21 7
11 15 12 13 14	23 16 12 9 5	25 16 12 9 3	23 6 12 19 5	25 6 12 19 3
16 20 17 18 19	7 4 25 18 11	7 4 23 20 11	17 4 25 8 11	17 4 23 10 11
21 25 22 23 24	20 13 6 2 24	18 15 6 2 24	10 13 16 2 24	8 15 16 2 24
289/P	289/R1	315/R2	298/R3	324/R4
1 6 16 21 11	1 20 14 8 22	1 20 14 23 7	1 20 12 8 24	1 20 12 23 9
2 7 17 22 12	13 7 21 5 19	13 22 6 5 19	13 9 21 5 17	13 24 6 5 17
3 8 18 23 13	25 4 18 12 6	10 4 18 12 21	25 2 18 14 6	10 2 18 14 21
4 9 19 24 14	17 11 10 24 3	17 11 25 9 3	19 11 10 22 3	19 11 25 7 3
5 10 20 25 15	9 23 2 16 15	24 8 2 16 15	7 23 4 16 15	22 8 4 16 15
361/P	433/R1	459/R2	442/R3	468/R4
1 11 16 21 6	1 20 9 13 22	1 20 9 23 12	1 20 7 13 24	1 20 7 23 14
2 12 17 22 7	8 12 21 5 19	8 22 11 5 19	8 14 21 5 17	8 24 11 5 17
3 13 18 23 8	25 4 18 7 11	15 4 18 7 21	25 2 18 9 11	15 2 18 9 21
4 14 19 24 9	17 6 15 24 3	17 6 25 14 3	19 6 15 22 3	19 6 25 12 3
5 15 20 25 10	14 23 2 16 10	24 13 2 16 10	12 23 4 16 10	22 13 4 16 10
433/P	445/R1	457/R2	454/R3	466/R4
1 16 11 21 6	1 15 9 18 22	1 15 9 23 17	1 15 7 18 24	1 15 7 23 19
2 17 12 22 7	8 17 21 5 14	8 22 16 5 14	8 19 21 5 12	8 24 16 5 12
3 18 13 23 8	25 4 13 7 16	20 4 13 7 21	25 2 13 9 16	20 2 13 9 21
4 19 14 24 9	12 6 20 24 3	12 6 25 19 3	14 6 20 22 3	14 6 25 17 3
5 20 15 25 10	19 23 2 11 10	24 18 2 11 10	17 23 4 11 10	22 18 4 11 10
505/P	457/R1	445/R2	466/R3	454/R4
1 21 11 16 6	1 15 9 23 17	1 15 9 18 22	1 15 7 23 19	1 15 7 18 24
2 22 12 17 7	8 22 16 5 14	8 17 21 5 14	8 24 16 5 12	8 19 21 5 12
3 23 13 18 8	20 4 13 7 21	25 4 13 7 16	20 2 13 9 21	25 2 13 9 16
4 24 14 19 9	12 6 25 19 3	12 6 20 24 3	14 6 25 17 3	14 6 20 22 3
5 25 15 20 10	24 18 2 11 10	19 23 2 11 10	22 18 4 11 10	17 23 4 11 10
577/P	577/R1	586/R2	603/R3	612/R4
2 1 3 4 5	2 23 20 11 9	2 23 20 14 6	2 23 10 11 19	2 23 10 14 16
7 6 8 9 10	15 6 4 22 18	15 9 1 22 18	15 16 4 22 8	15 19 1 22 8
12 11 13 14 15	24 17 13 10 1	21 17 13 10 4	24 7 13 20 1	21 7 13 20 4
17 16 18 19 20	8 5 21 19 12	8 5 24 16 12	18 5 21 9 12	18 5 24 6 12
22 21 23 24 25	16 14 7 3 25	19 11 7 3 25	6 14 17 3 25	9 11 17 3 25
1153/P	1153/R1	1162/R2	1179/R3	1188/R4
3 1 2 4 5	3 22 20 11 9	3 22 20 14 6	3 22 10 11 19	3 22 10 14 16
8 6 7 9 10	15 6 4 23 17	15 9 1 23 17	15 16 4 23 7	15 19 1 23 7
13 11 12 14 15	24 18 12 10 1	21 18 12 10 4	24 8 12 20 1	21 8 12 20 4
18 16 17 19 20	7 5 21 19 13	7 5 24 16 13	17 5 21 9 13	17 5 24 6 13
23 21 22 24 25	16 14 8 2 25	19 11 8 2 25	6 14 18 2 25	9 11 18 2 25
1729/P	1729/R1	1738/R2	1755/R3	1764/R4
4 1 2 3 5	4 22 20 11 8	4 22 20 13 6	4 22 10 11 18	4 22 10 13 16
9 6 7 8 10	15 6 3 24 17	15 8 1 24 17	15 16 3 24 7	15 18 1 24 7
14 11 12 13 15	23 19 12 10 1	21 19 12 10 3	23 9 12 20 1	21 9 12 20 3
19 16 17 18 20	7 5 21 18 14	7 5 23 16 14	17 5 21 8 14	17 5 23 6 14
24 21 22 23 25	16 13 9 2 25	18 11 9 2 25	6 13 19 2 25	8 11 19 2 25
2305/P	2305/R1	2314/R2	2331/R3	2340/R4
5 1 2 3 4	5 22 19 11 8	5 22 19 13 6	5 22 9 11 18	5 22 9 13 16
10 6 7 8 9	14 6 3 25 17	14 8 1 25 17	14 16 3 25 7	14 18 1 25 7
15 11 12 13 14	23 20 12 9 1	21 20 12 9 3	23 10 12 19 1	21 10 12 19 3
20 16 17 18 19	7 4 21 18 15	7 4 23 16 15	17 4 21 8 15	17 4 23 6 15
25 21 22 23 24	16 13 10 2 24	18 11 10 2 24	6 13 20 2 24	8 11 20 2 24

2881/P	2881/R1	2886/R2	2887/R3	2892/R4
6 7 8 9 10	6 23 20 2 14	6 23 20 4 12	6 23 15 2 19	6 23 15 4 17
11 12 13 14 15	5 12 9 21 18	5 14 7 21 18	5 17 9 21 13	5 19 7 21 13
1 2 3 4 5	24 16 3 15 7	22 16 3 15 9	24 11 3 20 7	22 11 3 20 9
16 17 18 19 20	13 10 22 19 1	13 10 24 17 1	18 10 22 14 1	18 10 24 12 1
21 22 23 24 25	17 4 11 8 25	19 2 11 8 25	12 4 16 8 25	14 2 16 8 25
3025/P	3025/R1	3030/R2	3031/R3	3036/R4
7 6 8 9 10	7 23 20 1 14	7 23 20 4 11	7 23 15 1 19	7 23 15 4 16
12 11 13 14 15	5 11 9 22 18	5 14 6 22 18	5 16 9 22 13	5 19 6 22 13
2 1 3 4 5	24 17 3 15 6	21 17 3 15 9	24 12 3 20 6	21 12 3 20 9
17 16 18 19 20	13 10 21 19 2	13 10 24 16 2	18 10 21 14 2	18 10 24 11 2
22 21 23 24 25	16 4 12 8 25	19 1 12 8 25	11 4 17 8 25	14 1 17 8 25
3169/P	3169/R1	3174/R2	3175/R3	3180/R4
8 6 7 9 10	8 22 20 1 14	8 22 20 4 11	8 22 15 1 19	8 22 15 4 16
13 11 12 14 15	5 11 9 23 17	5 14 6 23 17	5 16 9 23 12	5 19 6 23 12
3 1 2 4 5	24 18 2 15 6	21 18 2 15 9	24 13 2 20 6	21 13 2 20 9
18 16 17 19 20	12 10 21 19 3	12 10 24 16 3	17 10 21 14 3	17 10 24 11 3
23 21 22 24 25	16 4 13 7 25	19 1 13 7 25	11 4 18 7 25	14 1 18 7 25
3313/P	3313/R1	3318/R2	3319/R3	3324/R4
9 6 7 8 10	9 22 20 1 13	9 22 20 3 11	9 22 15 1 18	9 22 15 3 16
14 11 12 13 15	5 11 8 24 17	5 13 6 24 17	5 16 8 24 12	5 18 6 24 12
4 1 2 3 5	23 19 2 15 6	21 19 2 15 8	23 14 2 20 6	21 14 2 20 8
19 16 17 18 20	12 10 21 18 4	12 10 23 16 4	17 10 21 13 4	17 10 23 11 4
24 21 22 23 25	16 3 14 7 25	18 1 14 7 25	11 3 19 7 25	13 1 19 7 25
3457/P	3457/R1	3462/R2	3463/R3	3468/R4
10 6 7 8 9	10 22 19 1 13	10 22 19 3 11	10 22 14 1 18	10 22 14 3 16
15 11 12 13 14	4 11 8 25 17	4 13 6 25 17	4 16 8 25 12	4 18 6 25 12
5 1 2 3 4	23 20 2 14 6	21 20 2 14 8	23 15 2 19 6	21 15 2 19 8
20 16 17 18 19	12 9 21 18 5	12 9 23 16 5	17 9 21 13 5	17 9 23 11 5
25 21 22 23 24	16 3 15 7 24	18 1 15 7 24	11 3 20 7 24	13 1 20 7 24
.				
[Reconstruction ERRORS:	0/R1,	0/R2,	0/R3,	0/R4]

How beautiful those reconstructions are!

[List 3: Prototypes(/P) and SCMS55 Reconstructed(/R5~R8): after another Models]

1/P	364/R5	368/R6	376/R7	380/R8
1 2 3 4 5	1 23 14 10 17	1 23 12 10 19	1 23 14 20 7	1 23 12 20 9
6 7 8 9 10	15 7 16 3 24	15 9 16 3 22	15 17 6 3 24	15 19 6 3 22
11 12 13 14 15	18 4 25 12 6	18 2 25 14 6	8 4 25 12 16	8 2 25 14 16
16 17 18 19 20	22 11 8 19 5	24 11 8 17 5	22 11 18 9 5	24 11 18 7 5
21 22 23 24 25	9 20 2 21 13	7 20 4 21 13	19 10 2 21 13	17 10 4 21 13
73/P	400/R5	404/R6	412/R7	416/R8
1 3 2 4 5	1 22 14 10 18	1 22 13 10 19	1 22 14 20 8	1 22 13 20 9
6 8 7 9 10	15 8 16 2 24	15 9 16 2 23	15 18 6 2 24	15 19 6 2 23
11 13 12 14 15	17 4 25 13 6	17 3 25 14 6	7 4 25 13 16	7 3 25 14 16
16 18 17 19 20	23 11 7 19 5	24 11 7 18 5	23 11 17 9 5	24 11 17 8 5
21 23 22 24 25	9 20 3 21 12	8 20 4 21 12	19 10 3 21 12	18 10 4 21 12
145/P	404/R5	400/R6	416/R7	412/R8
1 4 2 3 5	1 22 13 10 19	1 22 14 10 18	1 22 13 20 9	1 22 14 20 8
6 9 7 8 10	15 9 16 2 23	15 8 16 2 24	15 19 6 2 23	15 18 6 2 24
11 14 12 13 15	17 3 25 14 6	17 4 25 13 6	7 3 25 14 16	7 4 25 13 16
16 19 17 18 20	24 11 7 18 5	23 11 7 19 5	24 11 17 8 5	23 11 17 9 5
21 24 22 23 25	8 20 4 21 12	9 20 3 21 12	18 10 4 21 12	19 10 3 21 12
217/P	408/R5	399/R6	420/R7	411/R8
1 5 2 3 4	1 22 13 9 20	1 22 15 9 18	1 22 13 19 10	1 22 15 19 8
6 10 7 8 9	14 10 16 2 23	14 8 16 2 25	14 20 6 2 23	14 18 6 2 25
11 15 12 13 14	17 3 24 15 6	17 5 24 13 6	7 3 24 15 16	7 5 24 13 16
16 20 17 18 19	25 11 7 18 4	23 11 7 20 4	25 11 17 8 4	23 11 17 10 4
21 25 22 23 24	8 19 5 21 12	10 19 3 21 12	18 9 5 21 12	20 9 3 21 12

289/P	218/R5	244/R6	222/R7	248/R8
1 6 16 21 11	1 20 23 12 9	1 20 8 12 24	1 20 23 14 7	1 20 8 14 22
2 7 17 22 12	13 7 4 16 25	13 22 4 16 10	13 9 2 16 25	13 24 2 16 10
3 8 18 23 13	19 21 15 8 2	19 6 15 23 2	17 21 15 8 4	17 6 15 23 4
4 9 19 24 14	10 3 17 24 11	25 3 17 9 11	10 3 19 22 11	25 3 19 7 11
5 10 20 25 15	22 14 6 5 18	7 14 21 5 18	24 12 6 5 18	9 12 21 5 18
361/P	230/R5	242/R6	234/R7	246/R8
1 11 16 21 6	1 20 23 7 14	1 20 13 7 24	1 20 23 9 12	1 20 13 9 22
2 12 17 22 7	8 12 4 16 25	8 22 4 16 15	8 14 2 16 25	8 24 2 16 15
3 13 18 23 8	19 21 10 13 2	19 11 10 23 2	17 21 10 13 4	17 11 10 23 4
4 14 19 24 9	15 3 17 24 6	25 3 17 14 6	15 3 19 22 6	25 3 19 12 6
5 15 20 25 10	22 9 11 5 18	12 9 21 5 18	24 7 11 5 18	14 7 21 5 18
433/P	374/R5	386/R6	378/R7	390/R8
1 16 11 21 6	1 15 23 7 19	1 15 18 7 24	1 15 23 9 17	1 15 18 9 22
2 17 12 22 7	8 17 4 11 25	8 22 4 11 20	8 19 2 11 25	8 24 2 11 20
3 18 13 23 8	14 21 10 18 2	14 16 10 23 2	12 21 10 18 4	12 16 10 23 4
4 19 14 24 9	20 3 12 24 6	25 3 12 19 6	20 3 14 22 6	25 3 14 17 6
5 20 15 25 10	22 9 16 5 13	17 9 21 5 13	24 7 16 5 13	19 7 21 5 13
505/P	386/R5	374/R6	390/R7	378/R8
1 21 11 16 6	1 15 18 7 24	1 15 23 7 19	1 15 18 9 22	1 15 23 9 17
2 22 12 17 7	8 22 4 11 20	8 17 4 11 25	8 24 2 11 20	8 19 2 11 25
3 23 13 18 8	14 16 10 23 2	14 21 10 18 2	12 16 10 23 4	12 21 10 18 4
4 24 14 19 9	25 3 12 19 6	20 3 12 24 6	25 3 14 17 6	20 3 14 22 6
5 25 15 20 10	17 9 21 5 13	22 9 16 5 13	19 7 21 5 13	24 7 16 5 13
577/P	940/R5	944/R6	952/R7	956/R8
2 1 3 4 5	2 23 14 10 16	2 23 11 10 19	2 23 14 20 6	2 23 11 20 9
7 6 8 9 10	15 6 17 3 24	15 9 17 3 21	15 16 7 3 24	15 19 7 3 21
12 11 13 14 15	18 4 25 11 7	18 1 25 14 7	8 4 25 11 17	8 1 25 14 17
17 16 18 19 20	21 12 8 19 5	24 12 8 16 5	21 12 18 9 5	24 12 18 6 5
22 21 23 24 25	9 20 1 22 13	6 20 4 22 13	19 10 1 22 13	16 10 4 22 13
1153/P	1516/R5	1520/R6	1528/R7	1532/R8
3 1 2 4 5	3 22 14 10 16	3 22 11 10 19	3 22 14 20 6	3 22 11 20 9
8 6 7 9 10	15 6 18 2 24	15 9 18 2 21	15 16 8 2 24	15 19 8 2 21
13 11 12 14 15	17 4 25 11 8	17 1 25 14 8	7 4 25 11 18	7 1 25 14 18
18 16 17 19 20	21 13 7 19 5	24 13 7 16 5	21 13 17 9 5	24 13 17 6 5
23 21 22 24 25	9 20 1 23 12	6 20 4 23 12	19 10 1 23 12	16 10 4 23 12
1729/P	2092/R5	2096/R6	2104/R7	2108/R8
4 1 2 3 5	4 22 13 10 16	4 22 11 10 18	4 22 13 20 6	4 22 11 20 8
9 6 7 8 10	15 6 19 2 23	15 8 19 2 21	15 16 9 2 23	15 18 9 2 21
14 11 12 13 15	17 3 25 11 9	17 1 25 13 9	7 3 25 11 19	7 1 25 13 19
19 16 17 18 20	21 14 7 18 5	23 14 7 16 5	21 14 17 8 5	23 14 17 6 5
24 21 22 23 25	8 20 1 24 12	6 20 3 24 12	18 10 1 24 12	16 10 3 24 12
2305/P	2668/R5	2672/R6	2680/R7	2684/R8
5 1 2 3 4	5 22 13 9 16	5 22 11 9 18	5 22 13 19 6	5 22 11 19 8
10 6 7 8 9	14 6 20 2 23	14 8 20 2 21	14 16 10 2 23	14 18 10 2 21
15 11 12 13 14	17 3 24 11 10	17 1 24 13 10	7 3 24 11 20	7 1 24 13 20
20 16 17 18 19	21 15 7 18 4	23 15 7 16 4	21 15 17 8 4	23 15 17 6 4
25 21 22 23 24	8 19 1 25 12	6 19 3 25 12	18 9 1 25 12	16 9 3 25 12
2881/P	?/R5	?/R6	?/R7	?/R8
6 7 8 9 10	6 23 4 15 17	6 23 2 15 19	6 23 4 20 12	6 23 2 20 14
11 12 13 14 15	5 12 16 8 24	5 14 16 8 22	5 17 11 8 24	5 19 11 8 22
1 2 3 4 5	18 9 25 2 11	18 7 25 4 11	13 9 25 2 16	13 7 25 4 16
16 17 18 19 20	22 1 13 19 10	24 1 13 17 10	22 1 18 14 10	24 1 18 12 10
21 22 23 24 25	14 20 7 21 3	12 20 9 21 3	19 15 7 21 3	17 15 9 21 3
3025/P	?/R5	?/R6	?/R7	?/R8
7 6 8 9 10	7 23 4 15 16	7 23 1 15 19	7 23 4 20 11	7 23 1 20 14
12 11 13 14 15	5 11 17 8 24	5 14 17 8 21	5 16 12 8 24	5 19 12 8 21
2 1 3 4 5	18 9 25 1 12	18 6 25 4 12	13 9 25 1 17	13 6 25 4 17
17 16 18 19 20	21 2 13 19 10	24 2 13 16 10	21 2 18 14 10	24 2 18 11 10
22 21 23 24 25	14 20 6 22 3	11 20 9 22 3	19 15 6 22 3	16 15 9 22 3

3169/P	?/R5	?/R6	?/R7	?/R8
8 6 7 9 10	8 22 4 15 16	8 22 1 15 19	8 22 4 20 11	8 22 1 20 14
13 11 12 14 15	5 11 18 7 24	5 14 18 7 21	5 16 13 7 24	5 19 13 7 21
3 1 2 4 5	17 9 25 1 13	17 6 25 4 13	12 9 25 1 18	12 6 25 4 18
18 16 17 19 20	21 3 12 19 10	24 3 12 16 10	21 3 17 14 10	24 3 17 11 10
23 21 22 24 25	14 20 6 23 2	11 20 9 23 2	19 15 6 23 2	16 15 9 23 2
3313/P	?/R5	?/R6	?/R7	?/R8
9 6 7 8 10	9 22 3 15 16	9 22 1 15 18	9 22 3 20 11	9 22 1 20 13
14 11 12 13 15	5 11 19 7 23	5 13 19 7 21	5 16 14 7 23	5 18 14 7 21
4 1 2 3 5	17 8 25 1 14	17 6 25 3 14	12 8 25 1 19	12 6 25 3 19
19 16 17 18 20	21 4 12 18 10	23 4 12 16 10	21 4 17 13 10	23 4 17 11 10
24 21 22 23 25	13 20 6 24 2	11 20 8 24 2	18 15 6 24 2	16 15 8 24 2
3457/P	?/R5	?/R6	?/R7	?/R8
10 6 7 8 9	10 22 3 14 16	10 22 1 14 18	10 22 3 19 11	10 22 1 19 13
15 11 12 13 14	4 11 20 7 23	4 13 20 7 21	4 16 15 7 23	4 18 15 7 21
5 1 2 3 4	17 8 24 1 15	17 6 24 3 15	12 8 24 1 20	12 6 24 3 20
20 16 17 18 19	21 5 12 18 9	23 5 12 16 9	21 5 17 13 9	23 5 17 11 9
25 21 22 23 24	13 19 6 25 2	11 19 8 25 2	18 14 6 25 2	16 14 8 25 2
.				
[Reconstruction ERRORS:	720/R5,	720/R6,	720/R7,	720/R8]
[Count = 3600/3600x4]				

** Calculated and Listed by Kanji Setsuda on Apr.18, 2011 with MacOSX & Xcode3 **

Do you notice these results look quite different from those of Self-complementary type? The first 4 DAM Transformations could reconstruct all the Object solutions in each case. And each of the last 4 DAMT [List 3] could reconstruct 80% of Objects. The rest 20% of reconstruction results are only against the List-forming Inequality Conditions for the Standard solutions: $n1 < n25$; $n1 < n9$; $n1 < n17$; and $n23 > n15$

If you prepare the Primitive sets of 28800 solutions of both Prototype and Object instead of Standard ones, you should have the complete reconstructions of them.

It means the Prototype Squares are really independent from the Models and DAMT, and they seem to be objective and universal by themselves, I would say.

#8. Another Experiment for Cooperative Reconstruction of Pan-Diagonal MS55

Can we reconstruct all the solutions of Objects with multiple DAM Transformations and with fewer Prototypes than 3600 for the Standard solutions? We want to have the cooperative power of reconstruction by taking any duplication out of the whole list.

Let me report here about the recent result of such the experiment as follows:

** Cooperative Reconstruction of Pan-diagonal Magic Squares of Order 5: **
 ** With the Set 1 of Prototype Squares and Multiple DAM Transformations **

[List 1: Prototypes(/P) and 4 Models of Pan-diagonal MS55(/R1~R4)]

1/P	1/R1	9/P	27/R3
1 2 3 4 5	1 23 20 12 9	1 2 3 4 5	1 23 10 12 19
6 7 8 9 10	15 7 4 21 18	16 17 18 19 20	15 17 4 21 8
11 12 13 14 15	24 16 13 10 2	11 12 13 14 15	24 6 13 20 2
16 17 18 19 20	8 5 22 19 11	6 7 8 9 10	18 5 22 9 11
21 22 23 24 25	17 14 6 3 25	21 22 23 24 25	7 14 16 3 25
169/P	10/R2	177/P	36/R4
1 4 3 2 5	1 23 20 14 7	1 4 3 2 5	1 23 10 14 17
6 9 8 7 10	15 9 2 21 18	16 19 18 17 20	15 19 2 21 8
11 14 13 12 15	22 16 13 10 4	11 14 13 12 15	22 6 13 20 4
16 19 18 17 20	8 5 24 17 11	6 9 8 7 10	18 5 24 7 11
21 24 23 22 25	19 12 6 3 25	21 24 23 22 25	9 12 16 3 25

[List 2: Prototypes(/P) and Pan-diagonal MS55 Reconstructed(/R1~R4)]

[1]	1/P	1/R1	10/R2	27/R3	36/R4
1 2 3 4 5	1 23 20 12 9	1 23 20 14 7	1 23 10 12 19	1 23 10 14 17	
6 7 8 9 10	15 7 4 21 18	15 9 2 21 18	15 17 4 21 8	15 19 2 21 8	
11 12 13 14 15	24 16 13 10 2	22 16 13 10 4	24 6 13 20 2	22 6 13 20 4	
16 17 18 19 20	8 5 22 19 11	8 5 24 17 11	18 5 22 9 11	18 5 24 7 11	
21 22 23 24 25	17 14 6 3 25	19 12 6 3 25	7 14 16 3 25	9 12 16 3 25	
[2]	2/P	145/R1	154/R2	171/R3	180/R4
1 2 3 4 5	1 18 25 12 9	1 18 25 14 7	1 18 10 12 24	1 18 10 14 22	
6 7 8 9 10	15 7 4 16 23	15 9 2 16 23	15 22 4 16 8	15 24 2 16 8	
11 12 13 14 15	19 21 13 10 2	17 21 13 10 4	19 6 13 25 2	17 6 13 25 4	
16 17 18 19 20	8 5 17 24 11	8 5 19 22 11	23 5 17 9 11	23 5 19 7 11	
21 22 23 24 25	22 14 6 3 20	24 12 6 3 20	7 14 21 3 20	9 12 21 3 20	
[3]	3/P	3/R1	12/R2	15/R3	24/R4
1 2 3 4 5	1 23 15 17 9	1 23 15 19 7	1 23 10 17 14	1 23 10 19 12	
6 7 8 9 10	20 7 4 21 13	20 9 2 21 13	20 12 4 21 8	20 14 2 21 8	
16 17 18 19 20	24 11 18 10 2	22 11 18 10 4	24 6 18 15 2	22 6 18 15 4	
11 12 13 14 15	8 5 22 14 16	8 5 24 12 16	13 5 22 9 16	13 5 24 7 16	
21 22 23 24 25	12 19 6 3 25	14 17 6 3 25	7 19 11 3 25	9 17 11 3 25	
[4]	4/P	13/R1	22/R2	25/R3	34/R4
1 2 3 4 5	1 23 20 7 14	1 23 20 9 12	1 23 15 7 19	1 23 15 9 17	
11 12 13 14 15	10 12 4 21 18	10 14 2 21 18	10 17 4 21 13	10 19 2 21 13	
6 7 8 9 10	24 16 8 15 2	22 16 8 15 4	24 11 8 20 2	22 11 8 20 4	
16 17 18 19 20	13 5 22 19 6	13 5 24 17 6	18 5 22 14 6	18 5 24 12 6	
21 22 23 24 25	17 9 11 3 25	19 7 11 3 25	12 9 16 3 25	14 7 16 3 25	
[5]	5/P	157/R1	166/R2	169/R3	178/R4
1 2 3 4 5	1 18 25 7 14	1 18 25 9 12	1 18 15 7 24	1 18 15 9 22	
11 12 13 14 15	10 12 4 16 23	10 14 2 16 23	10 22 4 16 13	10 24 2 16 13	
6 7 8 9 10	19 21 8 15 2	17 21 8 15 4	19 11 8 25 2	17 11 8 25 4	
16 17 18 19 20	13 5 17 24 6	13 5 19 22 6	23 5 17 14 6	23 5 19 12 6	
21 22 23 24 25	22 9 11 3 20	24 7 11 3 20	12 9 21 3 20	14 7 21 3 20	
[6]	8/P	301/R1	310/R2	313/R3	322/R4
1 2 3 4 5	1 13 25 7 19	1 13 25 9 17	1 13 20 7 24	1 13 20 9 22	
16 17 18 19 20	10 17 4 11 23	10 19 2 11 23	10 22 4 11 18	10 24 2 11 18	
6 7 8 9 10	14 21 8 20 2	12 21 8 20 4	14 16 8 25 2	12 16 8 25 4	
11 12 13 14 15	18 5 12 24 6	18 5 14 22 6	23 5 12 19 6	23 5 14 17 6	
21 22 23 24 25	22 9 16 3 15	24 7 16 3 15	17 9 21 3 15	19 7 21 3 15	
[13]	25/P	2/R1	6/R2	28/R3	32/R4
1 2 4 3 5	1 24 20 12 8	1 24 20 13 7	1 24 10 12 18	1 24 10 13 17	
6 7 9 8 10	15 7 3 21 19	15 8 2 21 19	15 17 3 21 9	15 18 2 21 9	
11 12 14 13 15	23 16 14 10 2	22 16 14 10 3	23 6 14 20 2	22 6 14 20 3	
16 17 19 18 20	9 5 22 18 11	9 5 23 17 11	19 5 22 8 11	19 5 23 7 11	
21 22 24 23 25	17 13 6 4 25	18 12 6 4 25	7 13 16 4 25	8 12 16 4 25	
[25]	49/P	38/R1	42/R2	64/R3	68/R4
1 2 5 3 4	1 25 19 12 8	1 25 19 13 7	1 25 9 12 18	1 25 9 13 17	
6 7 10 8 9	14 7 3 21 20	14 8 2 21 20	14 17 3 21 10	14 18 2 21 10	
11 12 15 13 14	23 16 15 9 2	22 16 15 9 3	23 6 15 19 2	22 6 15 19 3	
16 17 20 18 19	10 4 22 18 11	10 4 23 17 11	20 4 22 8 11	20 4 23 7 11	
21 22 25 23 24	17 13 6 5 24	18 12 6 5 24	7 13 16 5 24	8 12 16 5 24	
[37]	73/P	5/R1	9/R2	31/R3	35/R4
1 3 2 4 5	1 22 20 13 9	1 22 20 14 8	1 22 10 13 19	1 22 10 14 18	
6 8 7 9 10	15 8 4 21 17	15 9 3 21 17	15 18 4 21 7	15 19 3 21 7	
11 13 12 14 15	24 16 12 10 3	23 16 12 10 4	24 6 12 20 3	23 6 12 20 4	
16 18 17 19 20	7 5 23 19 11	7 5 24 18 11	17 5 23 9 11	17 5 24 8 11	
21 23 22 24 25	18 14 6 2 25	19 13 6 2 25	8 14 16 2 25	9 13 16 2 25	
[61]	157/P	77/R1	81/R2	103/R3	107/R4
1 4 2 5 3	1 22 18 14 10	1 22 18 15 9	1 22 8 14 20	1 22 8 15 19	
6 9 7 10 8	13 9 5 21 17	13 10 4 21 17	13 19 5 21 7	13 20 4 21 7	
11 14 12 15 13	25 16 12 8 4	24 16 12 8 5	25 6 12 18 4	24 6 12 18 5	
16 19 17 20 18	7 3 24 20 11	7 3 25 19 11	17 3 24 10 11	17 3 25 9 11	
21 24 22 25 23	19 15 6 2 23	20 14 6 2 23	9 15 16 2 23	10 14 16 2 23	

[73]	289/P	289/R1	315/R2	298/R3	324/R4
1 6 16 21 11	1 20 14 8 22	1 20 14 23 7	1 20 12 8 24	1 20 12 23 9	
2 7 17 22 12	13 7 21 5 19	13 22 6 5 19	13 9 21 5 17	13 24 6 5 17	
3 8 18 23 13	25 4 18 12 6	10 4 18 12 21	25 2 18 14 6	10 2 18 14 21	
4 9 19 24 14	17 11 10 24 3	17 11 25 9 3	19 11 10 22 3	19 11 25 7 3	
5 10 20 25 15	9 23 2 16 15	24 8 2 16 15	7 23 4 16 15	22 8 4 16 15	
[109]	361/P	433/R1	459/R2	442/R3	468/R4
1 11 16 21 6	1 20 9 13 22	1 20 9 23 12	1 20 7 13 24	1 20 7 23 14	
2 12 17 22 7	8 12 21 5 19	8 22 11 5 19	8 14 21 5 17	8 24 11 5 17	
3 13 18 23 8	25 4 18 7 11	15 4 18 7 21	25 2 18 9 11	15 2 18 9 21	
4 14 19 24 9	17 6 15 24 3	17 6 25 14 3	19 6 15 22 3	19 6 25 12 3	
5 15 20 25 10	14 23 2 16 10	24 13 2 16 10	12 23 4 16 10	22 13 4 16 10	
[133]	433/P	445/R1	457/R2	454/R3	466/R4
1 16 11 21 6	1 15 9 18 22	1 15 9 23 17	1 15 7 18 24	1 15 7 23 19	
2 17 12 22 7	8 17 21 5 14	8 22 16 5 14	8 19 21 5 12	8 24 16 5 12	
3 18 13 23 8	25 4 13 7 16	20 4 13 7 21	25 2 13 9 16	20 2 13 9 21	
4 19 14 24 9	12 6 20 24 3	12 6 25 19 3	14 6 20 22 3	14 6 25 17 3	
5 20 15 25 10	19 23 2 11 10	24 18 2 11 10	17 23 4 11 10	22 18 4 11 10	
[145]	577/P	577/R1	586/R2	603/R3	612/R4
2 1 3 4 5	2 23 20 11 9	2 23 20 14 6	2 23 10 11 19	2 23 10 14 16	
7 6 8 9 10	15 6 4 22 18	15 9 1 22 18	15 16 4 22 8	15 19 1 22 8	
12 11 13 14 15	24 17 13 10 1	21 17 13 10 4	24 7 13 20 1	21 7 13 20 4	
17 16 18 19 20	8 5 21 19 12	8 5 24 16 12	18 5 21 9 12	18 5 24 6 12	
22 21 23 24 25	16 14 7 3 25	19 11 7 3 25	6 14 17 3 25	9 11 17 3 25	
[289]	1153/P	1153/R1	1162/R2	1179/R3	1188/R4
3 1 2 4 5	3 22 20 11 9	3 22 20 14 6	3 22 10 11 19	3 22 10 14 16	
8 6 7 9 10	15 6 4 23 17	15 9 1 23 17	15 16 4 23 7	15 19 1 23 7	
13 11 12 14 15	24 18 12 10 1	21 18 12 10 4	24 8 12 20 1	21 8 12 20 4	
18 16 17 19 20	7 5 21 19 13	7 5 24 16 13	17 5 21 9 13	17 5 24 6 13	
23 21 22 24 25	16 14 8 2 25	19 11 8 2 25	6 14 18 2 25	9 11 18 2 25	
[433]	1729/P	1729/R1	1738/R2	1755/R3	1764/R4
4 1 2 3 5	4 22 20 11 8	4 22 20 13 6	4 22 10 11 18	4 22 10 13 16	
9 6 7 8 10	15 6 3 24 17	15 8 1 24 17	15 16 3 24 7	15 18 1 24 7	
14 11 12 13 15	23 19 12 10 1	21 19 12 10 3	23 9 12 20 1	21 9 12 20 3	
19 16 17 18 20	7 5 21 18 14	7 5 23 16 14	17 5 21 8 14	17 5 23 6 14	
24 21 22 23 25	16 13 9 2 25	18 11 9 2 25	6 13 19 2 25	8 11 19 2 25	
[577]	2305/P	2305/R1	2314/R2	2331/R3	2340/R4
5 1 2 3 4	5 22 19 11 8	5 22 19 13 6	5 22 9 11 18	5 22 9 13 16	
10 6 7 8 9	14 6 3 25 17	14 8 1 25 17	14 16 3 25 7	14 18 1 25 7	
15 11 12 13 14	23 20 12 9 1	21 20 12 9 3	23 10 12 19 1	21 10 12 19 3	
20 16 17 18 19	7 4 21 18 15	7 4 23 16 15	17 4 21 8 15	17 4 23 6 15	
25 21 22 23 24	16 13 10 2 24	18 11 10 2 24	6 13 20 2 24	8 11 20 2 24	
[721]	2881/P	2881/R1	2886/R2	2887/R3	2892/R4
6 7 8 9 10	6 23 20 2 14	6 23 20 4 12	6 23 15 2 19	6 23 15 4 17	
11 12 13 14 15	5 12 9 21 18	5 14 7 21 18	5 17 9 21 13	5 19 7 21 13	
1 2 3 4 5	24 16 3 15 7	22 16 3 15 9	24 11 3 20 7	22 11 3 20 9	
16 17 18 19 20	13 10 22 19 1	13 10 24 17 1	18 10 22 14 1	18 10 24 12 1	
21 22 23 24 25	17 4 11 8 25	19 2 11 8 25	12 4 16 8 25	14 2 16 8 25	
[757]	3025/P	3025/R1	3030/R2	3031/R3	3036/R4
7 6 8 9 10	7 23 20 1 14	7 23 20 4 11	7 23 15 1 19	7 23 15 4 16	
12 11 13 14 15	5 11 9 22 18	5 14 6 22 18	5 16 9 22 13	5 19 6 22 13	
2 1 3 4 5	24 17 3 15 6	21 17 3 15 9	24 12 3 20 6	21 12 3 20 9	
17 16 18 19 20	13 10 21 19 2	13 10 24 16 2	18 10 21 14 2	18 10 24 11 2	
22 21 23 24 25	16 4 12 8 25	19 1 12 8 25	11 4 17 8 25	14 1 17 8 25	
[793]	3169/P	3169/R1	3174/R2	3175/R3	3180/R4
8 6 7 9 10	8 22 20 1 14	8 22 20 4 11	8 22 15 1 19	8 22 15 4 16	
13 11 12 14 15	5 11 9 23 17	5 14 6 23 17	5 16 9 23 12	5 19 6 23 12	
3 1 2 4 5	24 18 2 15 6	21 18 2 15 9	24 13 2 20 6	21 13 2 20 9	
18 16 17 19 20	12 10 21 19 3	12 10 24 16 3	17 10 21 14 3	17 10 24 11 3	
23 21 22 24 25	16 4 13 7 25	19 1 13 7 25	11 4 18 7 25	14 1 18 7 25	

It also means you could classify the 3600 solutions of Object Squares into 4 groups of 900 solutions (or 4 groups of 144 for the set of 576 solutions when $n_1=1$).

When you take the Primitive sets of both Prototype and Object, you could expect to classify them such as 4×7200 , or 8×3600 , 16×1800 , ..., $144 \times 200 = 28800$ by increasing the numbers of Models and DAMT according to them.

It implies the new possibility of studying the structure of our solution set.

It is because our objective Pan-diagonal MS55 solutions should have any highest 'symmetry' in themselves, I suppose.

#9. Case of Simultaneous MS55: Both Self-complementary and Pan-diagonal

The Simultaneous MS55 of both Self-complementary and Pan-diagonal are so rare that they are thought to be one of the most precious jewels we have ever had.

I must mention here all those of order 5 were once discovered and announced with the complete solution list by Prof. Mutsumi Suzuki in his famous homepage of magic squares. I personally used to call them "Suzuki's Squares".

They are defined just as follows:

* Make the Simultaneous MS55: Self-complementary and Pan-diagonal */

** Basic Form for Suzuki's Squares 5x5 **

```

 2  3  4  5 | 1| 2| 3| 4| 5| 1  2  3  4
 7  8  9 10 | 6| 7| 8| 9|10| 6  7  8  9
12 13 14 15 |11|12|13|14|15|11 12 13 14
17 18 19 20 |16|17|18|19|20|16 17 18 19
22 23 24 25 |21|22|23|24|25|21 22 23 24

```

** Basic Conditions for Rows and Columns: C=65 **

```

n1+n2+n3+n4+n5=C      ...rw1; | n1+n6+n11+n16+n21=C      ...c1;
n6+n7+n8+n9+n10=C     ...rw2; | n2+n7+n12+n17+n22=C      ...c2;
n11+n12+n13+n14+n15=C ...rw3; | n3+n8+n13+n18+n23=C      ...c3;
n16+n17+n18+n19+n20=C ...rw4; | n4+n9+n14+n19+n24=C      ...c4;
n21+n22+n23+n24+n25=C ...rw5; | n5+n10+n15+n20+n25=C      ...c5;

```

** Pan-diagonal Conditions: C=65 **

```

n1+n7+n13+n19+n25=C   ...pd1; | n1+n10+n14+n18+n22=C   ...pb1;
n2+n8+n14+n20+n21=C   ...pd2; | n2+n6+n15+n19+n23=C   ...pb2;
n3+n9+n15+n16+n22=C   ...pd3; | n3+n7+n11+n20+n24=C   ...pb3;
n4+n10+n11+n17+n23=C  ...pd4; | n4+n8+n12+n16+n25=C   ...pb4;
n5+n6+n12+n18+n24=C   ...pd5; | n5+n9+n13+n17+n21=C   ...pb5;

```

** Self-complementary Conditions: SC=26 **

```

n1+n25=n2+n24=n3+n23=n4+n22=n5+n21=n6+n20=
n7+n19=n8+n18=n9+n17=n10+n16=n11+n15=n12+n14=n13+n13=
n14+n12=n15+n11=n16+n10=n17+n9=n18+n8=n19+n7=n20+n6=
n21+n5=n22+n4=n23+n3=n24+n2=n25+n1=SC ...scc

```

** List-forming Inequality Conditions **

$n_1 < n_5$; $n_1 < n_{21}$; $n_1 < n_{25}$; and $n_2 > n_6$;

[Standard Solutions of Simultaneous Suzuki's Squares of Order 5: with Check_Sums]

1/ Rw P1\					2/ Rw P1\					3/ Rw P1\					4/ Rw P1\				
1	23	20	12	9 65 65\	1	23	20	14	7 65 65\	1	23	10	12	19 65 65\	1	23	10	14	17 65 65\
15	7	4	21	18 65 65\	15	9	2	21	18 65 65\	15	17	4	21	8 65 65\	15	19	2	21	8 65 65\
24	16	13	10	2 65 65\	22	16	13	10	4 65 65\	24	6	13	20	2 65 65\	22	6	13	20	4 65 65\
8	5	22	19	11 65 65\	8	5	24	17	11 65 65\	18	5	22	9	11 65 65\	18	5	24	7	11 65 65\
17	14	6	3	25 65 65\	19	12	6	3	25 65 65\	7	14	16	3	25 65 65\	9	12	16	3	25 65 65\
65 65 65 65 65 Co1umn					65 65 65 65 65 Co1umn					65 65 65 65 65 Co1umn					65 65 65 65 65 Co1umn				
/65/65/65/65/65 P2Sums					/65/65/65/65/65 P2Sums					/65/65/65/65/65 P2Sums					/65/65/65/65/65 P2Sums				

5/ Rw P1\ 2 23 19 11 10 65 65\ 14 6 5 22 18 65 65\ 25 17 13 9 1 65 65\ 8 4 21 20 12 65 65\ 16 15 7 3 24 65 65\ -----+----- 65 65 65 65 65 Column /65/65/65/65/65 P2Sums					6/ Rw P1\ 2 23 19 15 6 65 65\ 14 10 1 22 18 65 65\ 21 17 13 9 5 65 65\ 8 4 25 16 12 65 65\ 20 11 7 3 24 65 65\ -----+----- 65 65 65 65 65 Column /65/65/65/65/65 P2Sums					7/ Rw P1\ 2 23 9 11 20 65 65\ 14 16 5 22 8 65 65\ 25 7 13 19 1 65 65\ 18 4 21 10 12 65 65\ 6 15 17 3 24 65 65\ -----+----- 65 65 65 65 65 Column /65/65/65/65/65 P2Sums					8/ Rw P1\ 2 23 9 15 16 65 65\ 14 20 1 22 8 65 65\ 21 7 13 19 5 65 65\ 18 4 25 6 12 65 65\ 10 11 17 3 24 65 65\ -----+----- 65 65 65 65 65 Column /65/65/65/65/65 P2Sums				
9/ Rw P1\ 4 23 17 11 10 65 65\ 12 6 5 24 18 65 65\ 25 19 13 7 1 65 65\ 8 2 21 20 14 65 65\ 16 15 9 3 22 65 65\ -----+----- 65 65 65 65 65 Column /65/65/65/65/65 P2Sums					10/ Rw P1\ 4 23 17 15 6 65 65\ 12 10 1 24 18 65 65\ 21 19 13 7 5 65 65\ 8 2 25 16 14 65 65\ 20 11 9 3 22 65 65\ -----+----- 65 65 65 65 65 Column /65/65/65/65/65 P2Sums					11/ Rw P1\ 4 23 7 11 20 65 65\ 12 16 5 24 8 65 65\ 25 9 13 17 1 65 65\ 18 2 21 10 14 65 65\ 6 15 19 3 22 65 65\ -----+----- 65 65 65 65 65 Column /65/65/65/65/65 P2Sums					12/ Rw P1\ 4 23 7 15 16 65 65\ 12 20 1 24 8 65 65\ 21 9 13 17 5 65 65\ 18 2 25 6 14 65 65\ 10 11 19 3 22 65 65\ -----+----- 65 65 65 65 65 Column /65/65/65/65/65 P2Sums				
13/ Rw P1\ 5 23 16 12 9 65 65\ 11 7 4 25 18 65 65\ 24 20 13 6 2 65 65\ 8 1 22 19 15 65 65\ 17 14 10 3 21 65 65\ -----+----- 65 65 65 65 65 Column /65/65/65/65/65 P2Sums					14/ Rw P1\ 5 23 16 14 7 65 65\ 11 9 2 25 18 65 65\ 22 20 13 6 4 65 65\ 8 1 24 17 15 65 65\ 19 12 10 3 21 65 65\ -----+----- 65 65 65 65 65 Column /65/65/65/65/65 P2Sums					15/ Rw P1\ 5 23 6 12 19 65 65\ 11 17 4 25 8 65 65\ 24 10 13 16 2 65 65\ 18 1 22 9 15 65 65\ 7 14 20 3 21 65 65\ -----+----- 65 65 65 65 65 Column /65/65/65/65/65 P2Sums					16/ Rw P1\ 5 23 6 14 17 65 65\ 11 19 2 25 8 65 65\ 22 10 13 16 4 65 65\ 18 1 24 7 15 65 65\ 9 12 20 3 21 65 65\ -----+----- 65 65 65 65 65 Column /65/65/65/65/65 P2Sums				

[Solution Counts = 16]

It contains such the 16 standard solutions as above. I listed them with sum_checks of their rows, columns and pan-diagonals.

Let's try to reconstruct all these 16 jewels with our DAMT and Prototype Squares.

We want to use the same DAMT and its same rules for this case.

Now we have to make another original set of Prototype Squares for them according to the conditions listed as follows.

**** Basic Diagrams and Conditions for Prototype Squares 5x5 ****

Prototype Squares					The Model Solution				
4 5 1 2 3 4 5	1 2	12 9 1 23 20 12 9	1 23						
9 10 6 7 8 9 10	6 7	21 18 15 7 4 21 18	15 7						
14 15 11 12 13 14 15	11 12	10 2 24 16 13 10 2	24 16						
19 20 16 17 18 19 20	16 17	19 11 8 5 22 19 11	8 5						
24 25 21 22 23 24 25	21 22	3 25 17 14 6 3 25	17 14						

*** Basic Conditions: C=65 ***

n1+n23+n20+n12+n9=C	...rw1;		n1+n15+n24+n8+n17=C	...c1;
n15+n7+n4+n21+n18=C	...rw2;		n23+n7+n16+n5+n14=C	...c2;
n24+n16+n13+n10+n2=C	...rw3;		n20+n4+n13+n22+n6=C	...c3;
n8+n5+n22+n19+n11=C	...rw4;		n12+n21+n10+n19+n3=C	...c4;
n17+n14+n6+n3+n25=C	...rw5;		n9+n18+n2+n11+n25=C	...c5;

*** Pan-diagonal Conditions: C=65 ***

n1+n7+n13+n19+n25=C	...pd1;		n1+n18+n10+n22+n14=C	...pb1;
n23+n4+n10+n11+n17=C	...pd2;		n23+n15+n2+n19+n6=C	...pb2;
n20+n21+n2+n8+n14=C	...pd3;		n20+n7+n24+n11+n3=C	...pb3;
n12+n18+n24+n5+n6=C	...pd4;		n12+n4+n16+n8+n25=C	...pb4;
n9+n15+n16+n22+n3=C	...pd5;		n9+n21+n13+n5+n17=C	...pb5;

*** Self-Complementary Conditions: SC=26 ***

```
n1+n25=n23+n3=n20+n6=n12+n14=n9+n17=n15+n11=
n7+n19=n4+n22=n21+n5=n18+n8=n24+n2=n16+n10=
n13+n13=n10+n16= ... =SC ...scc
```

```
* List-forming Inequality Conditions: *
n1<n25; n1<n9; n1<n17; and n23>n15;
```

#10. Parallel Reconstructions of Simultaneous MS55 with multiple DAMT

The next list shows the multiple Models and the DAM Transformation rules we have to take for the parallel reconstructions of objects.

```
** Multiple DAMT Rules for Simultaneous MS55: Self-complementary & Pan-diagonal **
```

```
void damt1(void){
q[ 1]=p[ 1]; q[ 2]=p[23]; q[ 3]=p[20]; q[ 4]=p[12]; q[ 5]=p[ 9]; // 1 23 20 12 9
q[ 6]=p[15]; q[ 7]=p[ 7]; q[ 8]=p[ 4]; q[ 9]=p[21]; q[10]=p[18]; // 15 7 4 21 18
q[11]=p[24]; q[12]=p[16]; q[13]=p[13]; q[14]=p[10]; q[15]=p[ 2]; // 24 16 13 10 2
q[16]=p[ 8]; q[17]=p[ 5]; q[18]=p[22]; q[19]=p[19]; q[20]=p[11]; // 8 5 22 19 11
q[21]=p[17]; q[22]=p[14]; q[23]=p[ 6]; q[24]=p[ 3]; q[25]=p[25];} // 17 14 6 3 25
```

```
void damt2(void){
q[ 1]=p[ 1]; q[ 2]=p[23]; q[ 3]=p[20]; q[ 4]=p[14]; q[ 5]=p[ 7]; // 1 23 20 14 7
q[ 6]=p[15]; q[ 7]=p[ 9]; q[ 8]=p[ 2]; q[ 9]=p[21]; q[10]=p[18]; // 15 9 2 21 18
q[11]=p[22]; q[12]=p[16]; q[13]=p[13]; q[14]=p[10]; q[15]=p[ 4]; // 22 16 13 10 4
q[16]=p[ 8]; q[17]=p[ 5]; q[18]=p[24]; q[19]=p[17]; q[20]=p[11]; // 8 5 24 17 11
q[21]=p[19]; q[22]=p[12]; q[23]=p[ 6]; q[24]=p[ 3]; q[25]=p[25];} // 19 12 6 3 25
```

```
void damt3(void){
q[ 1]=p[ 1]; q[ 2]=p[23]; q[ 3]=p[10]; q[ 4]=p[12]; q[ 5]=p[19]; // 1 23 10 12 19
q[ 6]=p[15]; q[ 7]=p[17]; q[ 8]=p[ 4]; q[ 9]=p[21]; q[10]=p[ 8]; // 15 17 4 21 8
q[11]=p[24]; q[12]=p[ 6]; q[13]=p[13]; q[14]=p[20]; q[15]=p[ 2]; // 24 6 13 20 2
q[16]=p[18]; q[17]=p[ 5]; q[18]=p[22]; q[19]=p[ 9]; q[20]=p[11]; // 18 5 22 9 11
q[21]=p[ 7]; q[22]=p[14]; q[23]=p[16]; q[24]=p[ 3]; q[25]=p[25];} // 7 14 16 3 25
```

```
void damt4(void){
q[ 1]=p[ 1]; q[ 2]=p[23]; q[ 3]=p[10]; q[ 4]=p[14]; q[ 5]=p[17]; // 1 23 10 14 17
q[ 6]=p[15]; q[ 7]=p[19]; q[ 8]=p[ 2]; q[ 9]=p[21]; q[10]=p[ 8]; // 15 19 2 21 8
q[11]=p[22]; q[12]=p[ 6]; q[13]=p[13]; q[14]=p[20]; q[15]=p[ 4]; // 22 6 13 20 4
q[16]=p[18]; q[17]=p[ 5]; q[18]=p[24]; q[19]=p[ 7]; q[20]=p[11]; // 18 5 24 7 11
q[21]=p[ 9]; q[22]=p[12]; q[23]=p[16]; q[24]=p[ 3]; q[25]=p[25];} // 9 12 16 3 25
```

Let me list out all the results of my experiment next as follows.

```
* Parallel Reconstructions of Simultaneous MS55 with multiple DAM Transformations *
```

```
** With the Prototype Set made after the Model #1 **
```

```
[Prototypes(/P) and Simultaneous MS55 Reconstructed(/R1~R4)]
```

1/P	1/R1	2/R2	3/R3	4/R4
1 2 3 4 5	1 23 20 12 9	1 23 20 14 7	1 23 10 12 19	1 23 10 14 17
6 7 8 9 10	15 7 4 21 18	15 9 2 21 18	15 17 4 21 8	15 19 2 21 8
11 12 13 14 15	24 16 13 10 2	22 16 13 10 4	24 6 13 20 2	22 6 13 20 4
16 17 18 19 20	8 5 22 19 11	8 5 24 17 11	18 5 22 9 11	18 5 24 7 11
21 22 23 24 25	17 14 6 3 25	19 12 6 3 25	7 14 16 3 25	9 12 16 3 25
2/P	3/R1	4/R2	1/R3	2/R4
1 2 3 4 5	1 23 10 12 19	1 23 10 14 17	1 23 20 12 9	1 23 20 14 7
16 17 18 19 20	15 17 4 21 8	15 19 2 21 8	15 7 4 21 18	15 9 2 21 18
11 12 13 14 15	24 6 13 20 2	22 6 13 20 4	24 16 13 10 2	22 16 13 10 4
6 7 8 9 10	18 5 22 9 11	18 5 24 7 11	8 5 22 19 11	8 5 24 17 11
21 22 23 24 25	7 14 16 3 25	9 12 16 3 25	17 14 6 3 25	19 12 6 3 25
3/P	2/R1	1/R2	4/R3	3/R4
1 4 3 2 5	1 23 20 14 7	1 23 20 12 9	1 23 10 14 17	1 23 10 12 19
6 9 8 7 10	15 9 2 21 18	15 7 4 21 18	15 19 2 21 8	15 17 4 21 8
11 14 13 12 15	22 16 13 10 4	24 16 13 10 2	22 6 13 20 4	24 6 13 20 2
16 19 18 17 20	8 5 24 17 11	8 5 22 19 11	18 5 24 7 11	18 5 22 9 11
21 24 23 22 25	19 12 6 3 25	17 14 6 3 25	9 12 16 3 25	7 14 16 3 25

4/P					4/R1					3/R2					2/R3					1/R4				
1	4	3	2	5	1	23	10	14	17	1	23	10	12	19	1	23	20	14	7	1	23	20	12	9
16	19	18	17	20	15	19	2	21	8	15	17	4	21	8	15	9	2	21	18	15	7	4	21	18
11	14	13	12	15	22	6	13	20	4	24	6	13	20	2	22	16	13	10	4	24	16	13	10	2
6	9	8	7	10	18	5	24	7	11	18	5	22	9	11	8	5	24	17	11	8	5	22	19	11
21	24	23	22	25	9	12	16	3	25	7	14	16	3	25	19	12	6	3	25	17	14	6	3	25
5/P					5/R1					6/R2					7/R3					8/R4				
2	1	3	5	4	2	23	19	11	10	2	23	19	15	6	2	23	9	11	20	2	23	9	15	16
7	6	8	10	9	14	6	5	22	18	14	10	1	22	18	14	16	5	22	8	14	20	1	22	8
12	11	13	15	14	25	17	13	9	1	21	17	13	9	5	25	7	13	19	1	21	7	13	19	5
17	16	18	20	19	8	4	21	20	12	8	4	25	16	12	18	4	21	10	12	18	4	25	6	12
22	21	23	25	24	16	15	7	3	24	20	11	7	3	24	6	15	17	3	24	10	11	17	3	24
6/P					7/R1					8/R2					5/R3					6/R4				
2	1	3	5	4	2	23	9	11	20	2	23	9	15	16	2	23	19	11	10	2	23	19	15	6
17	16	18	20	19	14	16	5	22	8	14	20	1	22	8	14	6	5	22	18	14	10	1	22	18
12	11	13	15	14	25	7	13	19	1	21	7	13	19	5	25	17	13	9	1	21	17	13	9	5
7	6	8	10	9	18	4	21	10	12	18	4	25	6	12	8	4	21	20	12	8	4	25	16	12
22	21	23	25	24	6	15	17	3	24	10	11	17	3	24	16	15	7	3	24	20	11	7	3	24
7/P					6/R1					5/R2					8/R3					7/R4				
2	5	3	1	4	2	23	19	15	6	2	23	19	11	10	2	23	9	15	16	2	23	9	11	20
7	10	8	6	9	14	10	1	22	18	14	6	5	22	18	14	20	1	22	8	14	16	5	22	8
12	15	13	11	14	21	17	13	9	5	25	17	13	9	1	21	7	13	19	5	25	7	13	19	1
17	20	18	16	19	8	4	25	16	12	8	4	21	20	12	18	4	25	6	12	18	4	21	10	12
22	25	23	21	24	20	11	7	3	24	16	15	7	3	24	10	11	17	3	24	6	15	17	3	24
8/P					8/R1					7/R2					6/R3					5/R4				
2	5	3	1	4	2	23	9	15	16	2	23	9	11	20	2	23	19	15	6	2	23	19	11	10
17	20	18	16	19	14	20	1	22	8	14	16	5	22	8	14	10	1	22	18	14	6	5	22	18
12	15	13	11	14	21	7	13	19	5	25	7	13	19	1	21	17	13	9	5	25	17	13	9	1
7	10	8	6	9	18	4	25	6	12	18	4	21	10	12	8	4	25	16	12	8	4	21	20	12
22	25	23	21	24	10	11	17	3	24	6	15	17	3	24	20	11	7	3	24	16	15	7	3	24
9/P					9/R1					10/R2					11/R3					12/R4				
4	1	3	5	2	4	23	17	11	10	4	23	17	15	6	4	23	7	11	20	4	23	7	15	16
9	6	8	10	7	12	6	5	24	18	12	10	1	24	18	12	16	5	24	8	12	20	1	24	8
14	11	13	15	12	25	19	13	7	1	21	19	13	7	5	25	9	13	17	1	21	9	13	17	5
19	16	18	20	17	8	2	21	20	14	8	2	25	16	14	18	2	21	10	14	18	2	25	6	14
24	21	23	25	22	16	15	9	3	22	20	11	9	3	22	6	15	19	3	22	10	11	19	3	22
10/P					11/R1					12/R2					9/R3					10/R4				
4	1	3	5	2	4	23	7	11	20	4	23	7	15	16	4	23	17	11	10	4	23	17	15	6
19	16	18	20	17	12	16	5	24	8	12	20	1	24	8	12	6	5	24	18	12	10	1	24	18
14	11	13	15	12	25	9	13	17	1	21	9	13	17	5	25	19	13	7	1	21	19	13	7	5
9	6	8	10	7	18	2	21	10	14	18	2	25	6	14	8	2	21	20	14	8	2	25	16	14
24	21	23	25	22	6	15	19	3	22	10	11	19	3	22	16	15	9	3	22	20	11	9	3	22
11/P					10/R1					9/R2					12/R3					11/R4				
4	5	3	1	2	4	23	17	15	6	4	23	17	11	10	4	23	7	15	16	4	23	7	11	20
9	10	8	6	7	12	10	1	24	18	12	6	5	24	18	12	20	1	24	8	12	16	5	24	8
14	15	13	11	12	21	19	13	7	5	25	19	13	7	1	21	9	13	17	5	25	9	13	17	1
19	20	18	16	17	8	2	25	16	14	8	2	21	20	14	18	2	25	6	14	18	2	21	10	14
24	25	23	21	22	20	11	9	3	22	16	15	9	3	22	10	11	19	3	22	6	15	19	3	22
12/P					12/R1					11/R2					10/R3					9/R4				
4	5	3	1	2	4	23	7	15	16	4	23	7	11	20	4	23	17	15	6	4	23	17	11	10
19	20	18	16	17	12	20	1	24	8	12	16	5	24	8	12	10	1	24	18	12	6	5	24	18
14	15	13	11	12	21	9	13	17	5	25	9	13	17	1	21	19	13	7	5	25	19	13	7	1
9	10	8	6	7	18	2	25	6	14	18	2	21	10	14	8	2	25	16	14	8	2	21	20	14
24	25	23	21	22	10	11	19	3	22	6	15	19	3	22	20	11	9	3	22	16	15	9	3	22
13/P					13/R1					14/R2					15/R3					16/R4				
5	2	3	4	1	5	23	16	12	9	5	23	16	14	7	5	23	6	12	19	5	23	6	14	17
10	7	8	9	6	11	7	4	25	18	11	9	2	25	18	11	17	4	25	8	11	19	2	25	8
15	12	13	14	11	24	20	13	6	2	22	20	13	6	4	24	10	13	16	2	22	10	13	16	4
20	17	18	19	16	8	1	22	19	15	8	1	24	17	15	18	1	22	9	15	18	1	24	7	15
25	22	23	24	21	17	14	10	3	21	19	12	10	3	21	7	14	20	3	21	9	12	20	3	21

14/P	15/R1	16/R2	13/R3	14/R4
5 2 3 4 1	5 23 6 12 19	5 23 6 14 17	5 23 16 12 9	5 23 16 14 7
20 17 18 19 16	11 17 4 25 8	11 19 2 25 8	11 7 4 25 18	11 9 2 25 18
15 12 13 14 11	24 10 13 16 2	22 10 13 16 4	24 20 13 6 2	22 20 13 6 4
10 7 8 9 6	18 1 22 9 15	18 1 24 7 15	8 1 22 19 15	8 1 24 17 15
25 22 23 24 21	7 14 20 3 21	9 12 20 3 21	17 14 10 3 21	19 12 10 3 21
15/P	14/R1	13/R2	16/R3	15/R4
5 4 3 2 1	5 23 16 14 7	5 23 16 12 9	5 23 6 14 17	5 23 6 12 19
10 9 8 7 6	11 9 2 25 18	11 7 4 25 18	11 19 2 25 8	11 17 4 25 8
15 14 13 12 11	22 20 13 6 4	24 20 13 6 2	22 10 13 16 4	24 10 13 16 2
20 19 18 17 16	8 1 24 17 15	8 1 22 19 15	18 1 24 7 15	18 1 22 9 15
25 24 23 22 21	19 12 10 3 21	17 14 10 3 21	9 12 20 3 21	7 14 20 3 21
16/P	16/R1	15/R2	14/R3	13/R4
5 4 3 2 1	5 23 6 14 17	5 23 6 12 19	5 23 16 14 7	5 23 16 12 9
20 19 18 17 16	11 19 2 25 8	11 17 4 25 8	11 9 2 25 18	11 7 4 25 18
15 14 13 12 11	22 10 13 16 4	24 10 13 16 2	22 20 13 6 4	24 20 13 6 2
10 9 8 7 6	18 1 24 7 15	18 1 22 9 15	8 1 24 17 15	8 1 22 19 15
25 24 23 22 21	9 12 20 3 21	7 14 20 3 21	19 12 10 3 21	17 14 10 3 21
[Reconstruction ERRORS: 0/R1, 0/R2, 0/R3, 0/R4]				
[Count = 16/16x4]				

** Calculated and Listed by Kanji Setsuda on Apr.22, 2011 with MacOSX & Xcode3 **

As you see, our Prototype solutions are always effective even to the other Models and DAMT than #1, enough to reconstruct all the object solutions without any fails. This means they are essentially independent from the Models and 'universal'.

What a 'symmetry' those Suzuki's Squares have!

#11. Cooperative Reconstructions of Simultaneous MS55 with multiple DAMT

The next list shows the same objects with Latin Units of 'Complete Euler Squares'. Yes. They could be created for such type by the 5-th Increment Number System.

** 'Complete Euler Squares' of Order 5 for the Simultaneous MS55: **
 ** Self-complementary & Pan-diagonal; Made by New Euler's Method **

[Standard Solutions with Latin Units/H/L Solution_Number#|Check_Sums|]

1/H	2/L	1# Rw C1 P1\P2/	1/H	4/L	2# Rw C1 P1\P2/
04321	02413	1 23 20 12 9 65 65 65\65/	04321	02431	1 23 20 14 7 65 65 65\65/
21043	41302	15 7 4 21 18 65 65 65\65/	21043	43102	15 9 2 21 18 65 65 65\65/
43210	30241	24 16 13 10 2 65 65 65\65/	43210	10243	22 16 13 10 4 65 65 65\65/
10432	24130	8 5 22 19 11 65 65 65\65/	10432	24310	8 5 24 17 11 65 65 65\65/
32104	13024	17 14 6 3 25 65 65 65\65/	32104	31024	19 12 6 3 25 65 65 65\65/
3/H	2/L	3# Rw C1 P1\P2/	3/H	4/L	4# Rw C1 P1\P2/
04123	02413	1 23 10 12 19 65 65 65\65/	04123	02431	1 23 10 14 17 65 65 65\65/
23041	41302	15 17 4 21 8 65 65 65\65/	23041	43102	15 19 2 21 8 65 65 65\65/
41230	30241	24 6 13 20 2 65 65 65\65/	41230	10243	22 6 13 20 4 65 65 65\65/
30412	24130	18 5 22 9 11 65 65 65\65/	30412	24310	18 5 24 7 11 65 65 65\65/
12304	13024	7 14 16 3 25 65 65 65\65/	12304	31024	9 12 16 3 25 65 65 65\65/
1/H	6/L	5# Rw C1 P1\P2/	1/H	8/L	6# Rw C1 P1\P2/
04321	12304	2 23 19 11 10 65 65 65\65/	04321	12340	2 23 19 15 6 65 65 65\65/
21043	30412	14 6 5 22 18 65 65 65\65/	21043	34012	14 10 1 22 18 65 65 65\65/
43210	41230	25 17 13 9 1 65 65 65\65/	43210	01234	21 17 13 9 5 65 65 65\65/
10432	23041	8 4 21 20 12 65 65 65\65/	10432	23401	8 4 25 16 12 65 65 65\65/
32104	04123	16 15 7 3 24 65 65 65\65/	32104	40123	20 11 7 3 24 65 65 65\65/
3/H	6/L	7# Rw C1 P1\P2/	3/H	8/L	8# Rw C1 P1\P2/
04123	12304	2 23 9 11 20 65 65 65\65/	04123	12340	2 23 9 15 16 65 65 65\65/
23041	30412	14 16 5 22 8 65 65 65\65/	23041	34012	14 20 1 22 8 65 65 65\65/
41230	41230	25 7 13 19 1 65 65 65\65/	41230	01234	21 7 13 19 5 65 65 65\65/
30412	23041	18 4 21 10 12 65 65 65\65/	30412	23401	18 4 25 6 12 65 65 65\65/
12304	04123	6 15 17 3 24 65 65 65\65/	12304	40123	10 11 17 3 24 65 65 65\65/

1/H 9/L	9# Rw C1 P1\P2/	1/H 11/L	10# Rw C1 P1\P2/
04321 32104 4 23 17 11 10 65 65 65\65/		04321 32140 4 23 17 15 6 65 65 65\65/	
21043 10432 12 6 5 24 18 65 65 65\65/		21043 14032 12 10 1 24 18 65 65 65\65/	
43210 43210 25 19 13 7 1 65 65 65\65/		43210 03214 21 19 13 7 5 65 65 65\65/	
10432 21043 8 2 21 20 14 65 65 65\65/		10432 21403 8 2 25 16 14 65 65 65\65/	
32104 04321 16 15 9 3 22 65 65 65\65/		32104 40321 20 11 9 3 22 65 65 65\65/	
3/H 9/L	11# Rw C1 P1\P2/	3/H 11/L	12# Rw C1 P1\P2/
04123 32104 4 23 7 11 20 65 65 65\65/		04123 32140 4 23 7 15 16 65 65 65\65/	
23041 10432 12 16 5 24 8 65 65 65\65/		23041 14032 12 20 1 24 8 65 65 65\65/	
41230 43210 25 9 13 17 1 65 65 65\65/		41230 03214 21 9 13 17 5 65 65 65\65/	
30412 21043 18 2 21 10 14 65 65 65\65/		30412 21403 18 2 25 6 14 65 65 65\65/	
12304 04321 6 15 19 3 22 65 65 65\65/		12304 40321 10 11 19 3 22 65 65 65\65/	
1/H 13/L	13# Rw C1 P1\P2/	1/H 15/L	14# Rw C1 P1\P2/
04321 42013 5 23 16 12 9 65 65 65\65/		04321 42031 5 23 16 14 7 65 65 65\65/	
21043 01342 11 7 4 25 18 65 65 65\65/		21043 03142 11 9 2 25 18 65 65 65\65/	
43210 34201 24 20 13 6 2 65 65 65\65/		43210 14203 22 20 13 6 4 65 65 65\65/	
10432 20134 8 1 22 19 15 65 65 65\65/		10432 20314 8 1 24 17 15 65 65 65\65/	
32104 13420 17 14 10 3 21 65 65 65\65/		32104 31420 19 12 10 3 21 65 65 65\65/	
3/H 13/L	15# Rw C1 P1\P2/	3/H 15/L	16# Rw C1 P1\P2/
04123 42013 5 23 6 12 19 65 65 65\65/		04123 42031 5 23 6 14 17 65 65 65\65/	
23041 01342 11 17 4 25 8 65 65 65\65/		23041 03142 11 19 2 25 8 65 65 65\65/	
41230 34201 24 10 13 16 2 65 65 65\65/		41230 14203 22 10 13 16 4 65 65 65\65/	
30412 20134 18 1 22 9 15 65 65 65\65/		30412 20314 18 1 24 7 15 65 65 65\65/	
12304 13420 7 14 20 3 21 65 65 65\65/		12304 31420 9 12 20 3 21 65 65 65\65/	

[Solution Counts(n1==1/A11) = 4/16] OK!

Let's make all those 16 objects with multiple DAMT and fewer Prototype squares. We want to reconstruct the single set of 16 solutions with them, eliminating any duplication out of the whole list. The next report shows my newest result.

**** Cooperative Reconstructions of Simultaneous MS55 with Multiple DAM Transformations and fewer Prototype Squares made after the Model #1: ****

[List 1: Prototypes(/P) and Simultaneous Models Reconstructed(n1==1)]

[1]	1/P	1/H 2/L	1/R	[2]	2/P	3/H 2/L	3/R
1 2 3 4 5	04321	02413 1 23 20 12 9		1 2 3 4 5	04123	02413 1 23 10 12 19	
6 7 8 9 10	21043	41302 15 7 4 21 18		16 17 18 19 20	23041	41302 15 17 4 21 8	
11 12 13 14 15	43210	30241 24 16 13 10 2		11 12 13 14 15	41230	30241 24 6 13 20 2	
16 17 18 19 20	10432	24130 8 5 22 19 11		6 7 8 9 10	30412	24130 18 5 22 9 11	
21 22 23 24 25	32104	13024 17 14 6 3 25		21 22 23 24 25	12304	13024 7 14 16 3 25	
[3]	3/P	1/H 4/L	2/R	[4]	4/P	3/H 4/L	4/R
1 4 3 2 5	04321	02431 1 23 20 14 7		1 4 3 2 5	04123	02431 1 23 10 14 17	
6 9 8 7 10	21043	43102 15 9 2 21 18		16 19 18 17 20	23041	43102 15 19 2 21 8	
11 14 13 12 15	43210	10243 22 16 13 10 4		11 14 13 12 15	41230	10243 22 6 13 20 4	
16 19 18 17 20	10432	24310 8 5 24 17 11		6 9 8 7 10	30412	24310 18 5 24 7 11	
21 24 23 22 25	32104	31024 19 12 6 3 25		21 24 23 22 25	12304	31024 9 12 16 3 25	

[List 2: Prototypes(/P) and Suzuki's Squares Reconstructed(/R1~R4)]

[1]	1/P	1H 2L 1/R1	1H 4L 2/R2	3H 2L 3/R3	3H 4L 4/R4
1 2 3 4 5		1 23 20 12 9	1 23 20 14 7	1 23 10 12 19	1 23 10 14 17
6 7 8 9 10		15 7 4 21 18	15 9 2 21 18	15 17 4 21 8	15 19 2 21 8
11 12 13 14 15		24 16 13 10 2	22 16 13 10 4	24 6 13 20 2	22 6 13 20 4
16 17 18 19 20		8 5 22 19 11	8 5 24 17 11	18 5 22 9 11	18 5 24 7 11
21 22 23 24 25		17 14 6 3 25	19 12 6 3 25	7 14 16 3 25	9 12 16 3 25
[2]	5/P	1H 6L 5/R2	1H 8L 6/R3	3H 6L 7/R4	3H 8L 8/R5
2 1 3 5 4		2 23 19 11 10	2 23 19 15 6	2 23 9 11 20	2 23 9 15 16
7 6 8 10 9		14 6 5 22 18	14 10 1 22 18	14 16 5 22 8	14 20 1 22 8
12 11 13 15 14		25 17 13 9 1	21 17 13 9 5	25 7 13 19 1	21 7 13 19 5
17 16 18 20 19		8 4 21 20 12	8 4 25 16 12	18 4 21 10 12	18 4 25 6 12
22 21 23 25 24		16 15 7 3 24	20 11 7 3 24	6 15 17 3 24	10 11 17 3 24

[3]	9/P	1H 9L 9/R3	1H 11L 10/R4	3H 9L 11/R5	3H 11L 12/R6
4 1 3 5 2	4 23 17 11 10	4 23 17 15 6	4 23 7 11 20	4 23 7 15 16	
9 6 8 10 7	12 6 5 24 18	12 10 1 24 18	12 16 5 24 8	12 20 1 24 8	
14 11 13 15 12	25 19 13 7 1	21 19 13 7 5	25 9 13 17 1	21 9 13 17 5	
19 16 18 20 17	8 2 21 20 14	8 2 25 16 14	18 2 21 10 14	18 2 25 6 14	
24 21 23 25 22	16 15 9 3 22	20 11 9 3 22	6 15 19 3 22	10 11 19 3 22	
[4]	13/P	1H 13L 13/R4	1H 15L 14/R5	3H 13L 15/R6	3H 15L 16/R7
5 2 3 4 1	5 23 16 12 9	5 23 16 14 7	5 23 6 12 19	5 23 6 14 17	
10 7 8 9 6	11 7 4 25 18	11 9 2 25 18	11 17 4 25 8	11 19 2 25 8	
15 12 13 14 11	24 20 13 6 2	22 20 13 6 4	24 10 13 16 2	22 10 13 16 4	
20 17 18 19 16	8 1 22 19 15	8 1 24 17 15	18 1 22 9 15	18 1 24 7 15	
25 22 23 24 21	17 14 10 3 21	19 12 10 3 21	7 14 20 3 21	9 12 20 3 21	

[Solution Counts: $1 \times 4 \times 4 = 16$]

[Monitor List of Reconstructions by the Solution Numbers]

?: 0

1: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

OK!

** Calculated and Listed by Kanji Setsuda on Apr.22, 2011 with MacOSX & Xcode 3 **

This result tells us about the new way of classifying the solutions into 4 groups.

It could be regarded that the Prototype Squares illustrate the conceptual diagrams of basic transformations.

Please read the article in the Section 4, Chapter 3 of Part 3. I wrote about "Analysis on the 'Classic Composition' of Order 5". Will you compare the last results of that with the ones of this article? You will find some inspiration from your comparison, I hope.

#12. Comment

We could have finally succeeded in making all the three types of Prototype Squares of order 5, and finding the complete One-to-one Correspondences between their Prototype Squares and objective Magic Squares of order 5.

If you only have a single set of simple transformation rules and an appropriate set of Prototype Squares, you can always reconstruct the corresponding set of magic squares automatically, whenever you apply the 'Do-it-After-the-Model Transformation'.

It is really amazing. But why could we have done that? What makes it possible?

What do the Prototype Squares really mean?

I guess they are the basis of all object squares. The first representative form of Prototype Squares is just the same with Regular Array 5x5, and it has the master key to open all doors of every world, because we always make any type of magic squares on the Basic Array itself.

Isn't it the Mother of all magic squares?

Don't you think we have to study more about this life-giving Mother herself?

We have to start studying the structures of everything.

The solution set of Prototype Squares must be thought to be the set of Basic Forms of object squares, when I extend its concept to that which should contain all the conceptual diagrams of various basic transformations.

You cannot deny that the top solution of the Prototype Squares is just the Basic Form of all, can you? Each of the successive ones should illustrate how the Basic Form is transformed. Therefore, you should be able to make it into each object solution by our DAM Transformation.

I feel one of the most wonderful things in our magic square world is the very Model and DAM Transformation.

A single Model Solution and the simple rules of Transformation defined after that are really capable of making all the Object squares when provided with the solution set of Prototype Squares.

In the case of order 5, the only common Model could make all the three types of magic squares 5x5: (1) Self-complementary, (2) Pan-diagonal and (3) Simultaneous type: both self-complementary and pan-diagonal.

It means that the single common Model selected is just the only one Fundamental Solution and the representative of all. How amazing it is!

(The original article was written in 2001, 2003, 2005 and 2006 by Kanji Setsuda.

Revised on May 4, 2011 by Kanji Setsuda, working with MacOSX 10.5.8 & Xcode 3.1.2)

E-Mail Address: <jag12001@nifty.com>

Appendix: Program List for the Case of Simultaneous MS55

```
/** 'Prototype Squares' of Order 5 and Reconstructions of the Simultaneous **/  
/** Magic Squares: Self-complementary and Pan-diagonal; by multiple 'DAMT' **/  
/** 'ProtoT55SmlF.c' Dictated by Kanji Setsuda in **/  
/** 2005, 2006; and on Dec. 7, 2010; Apr.22, 2011 **/  
/** Working with MacOSX 10.5.8 & Xcode3.1.2 **/  
/**/  
/* Using Library */  
#include <stdio.h>  
/**/  
/* Global Variables */  
short int cnt, cnt1, cnt2, cnt3;  
short LSM, PSM, mdl;  
short nm[26], uflg[26];  
short dnm[26];  
short ln[17];  
short d[6][46];  
short st[17][26]; //for Solution Table  
short pn[17][26]; //for Prototype Squares  
/**/  
/* Sub-Routine: */  
void sstp01(void), sstp02(void), sstp03(void), sstp04(void);  
void sstp05(void), sstp06(void), sstp07(void), sstp08(void);  
void sstp09(void), sstp10(void), sstp11(void), sstp12(void);  
void sstp13(void);  
void solrecord(void);  
void prsol(short x);  
void chksum(void);  
/**/  
void pstp01(void), pstp02(void), pstp03(void), pstp04(void);  
void pstp05(void), pstp06(void), pstp07(void), pstp08(void);  
void pstp09(void), pstp10(void), pstp11(void), pstp12(void);  
void pstp13(void);  
void protorecord(void);  
void sortp(void);  
void exc(short x);  
void damt5(short x);  
/**/  
void pr4mdls(void);  
void mulmdl(short x1, short x2, short x3, short x4);  
void muldamt(short x1, short x2, short x3, short x4);  
void pr5ans(void);  
short findoldn(void);  
void prmon(void);  
/**/
```

```

/* Main Program */
int main(){
short n;
printf("\n** Simultaneous Magic Squares of Order 5: Self-complementary & Pan-diagonal; **\n");
printf(" [Standard Solutions of Suzuki's Squares of Order 5: with Check_Sums]\n");
for(n=0;n<26;n++){nm[n]=0; uflg[n]=0; st[0][n]=0;}
LSM=65; PSM=26; cnt=0; cnt3=0;
nm[13]=13; uflg[13]=1; //Set n13=13;
sstp01(); //Make the Table of Solutions
nm[13]=0; uflg[13]=0; //Reset n13;
printf(" [Solution Counts = %d]\n",cnt);
for(n=0;n<26;n++){nm[n]=0; uflg[n]=0;}
LSM=65; PSM=26; cnt1=0;
nm[13]=13; uflg[13]=1;
pstp01(); //Make the Table of Prototype Solutions
nm[13]=0; uflg[13]=0;
sortp(); //Sort Prototype Solutions for Smart List
printf("\n ** Cooperative Reconstruction of Simultaneous Magic Squares of Order 5: **\n");
printf(" ** Self-complementary & Pan-diagonal; with Multiple DAM Transformations **\n",1);
printf("\n [List 1: Prototypes(/P) and 4 Models Reconstructed(/R1~R4)]\n");
for(n=0;n<=16;n++){ln[n]=0;}; cnt2=0;
mdl=1; mulmdl(mdl,2,3,4); //Define 4 Models for 'Multiple DAMT'
printf("\n [List 2: Prototypes(/P) and Pan-diagonal MS55 Reconstructed(/R1~R4)]\n");
for(n=0;n<=16;n++){ln[n]=0;}; cnt2=0;
mdl=1; muldamt(mdl,2,3,4); //Reconstruct Objects with multiple DAMT
printf(" [Solution Counts: 1 x %d x %d = %d]\n",4,cnt2,cnt1);
printf("\n [Reconstruction Monitor by the Solution Numbers between Old and New]\n");
prmon();
printf(" OK!\n");
printf("** Calculated and Listed by Kanji Setsuda on Apr.22, 2011 with MacOSX & Xcode3 **\n");
return 0;
}
/**/
/* Make the Simultaneous MS55: Self-complementary and Pan-diagonal */
/* Level 1: */
/* Set n1 & n25 & n1<n25 */
void sstp01(){
short a,b;
for(a=1;a<13;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[1]=a; nm[25]=b;
uflg[a]=1; uflg[b]=1;
sstp02();
uflg[b]=0; uflg[a]=0;}
}
}
/* Set n7 & n19 */
void sstp02(){
short a,b;
for(a=1;a<26;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[7]=a; nm[19]=b;
uflg[a]=1; uflg[b]=1;
if(nm[1]+nm[7]+nm[13]+nm[19]+nm[25]==LSM){sstp03();}
uflg[b]=0; uflg[a]=0;}
}
}
/* Level 2: */
/* Set n2 & n24 */
void sstp03(){
short a,b;
for(a=25;a>0;a--){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[2]=a; nm[24]=b;
uflg[a]=1; uflg[b]=1;
}
}
}

```

```

        sstp04();
        uflg[b]=0; uflg[a]=0;}
    }
}
/* Set n3 & n23 */
void sstp04(){
    short a,b;
    for(a=25;a>0;a--){b=PSM-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[3]=a; nm[23]=b;
            uflg[a]=1; uflg[b]=1;
            sstp05();
            uflg[b]=0; uflg[a]=0;}
        }
}
/* Set n8 & n18 */
void sstp05(){
    short a,b;
    for(a=1;a<26;a++){b=PSM-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[8]=a; nm[18]=b;
            uflg[a]=1; uflg[b]=1;
            if(nm[3]+nm[8]+nm[13]+nm[18]+nm[23]==LSM){sstp06();}
            uflg[b]=0; uflg[a]=0;}
        }
}
/* Set n4 & n22 */
void sstp06(){
    short a,b;
    for(a=1;a<26;a++){b=PSM-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[4]=a; nm[22]=b;
            uflg[a]=1; uflg[b]=1;
            sstp07();
            uflg[b]=0; uflg[a]=0;}
        }
}
/* Set n5=65-n1-n2-n3-n4(>n1) & n21(>n1) */
void sstp07(){
    short a,b;
    a=LSM-nm[1]-nm[2]-nm[3]-nm[4];
    b=LSM-nm[25]-nm[24]-nm[23]-nm[22];
    if((nm[1]<a)&&(a<26)&&(a+b==PSM)){
        if((b>nm[1])&&(uflg[a]==0)&&(uflg[b]==0)){
            nm[5]=a; nm[21]=b;
            uflg[a]=1; uflg[b]=1;
            sstp08();
            uflg[b]=0; uflg[a]=0;}}
}
/* Level 3: */
/* Set n11 & n15 */
void sstp08(){
    short a,b;
    for(a=25;a>0;a--){b=PSM-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[11]=a; nm[15]=b;
            uflg[a]=1; uflg[b]=1;
            sstp09();
            uflg[b]=0; uflg[a]=0;}
        }
}
/* Set n6=65-n2-n15-n19-n23 (n6<n2) & n20 */
void sstp09(){
    short a,b;
    a=LSM-nm[2]-nm[15]-nm[19]-nm[23];

```

```

b=LSM-nm[24]-nm[11]-nm[7]-nm[3];
if((0<a)&&(a<nm[2])&&(a+b==PSM)){
    if((uflg[a]==0)&&(uflg[b]==0)){
        nm[6]=a; nm[20]=b;
        uflg[a]=1; uflg[b]=1;
        sstp10();
        uflg[b]=0; uflg[a]=0;}}
}
/* Set n16=65-n1-n6-n11-n21 & n10 */
void sstp10(){
    short a,b;
    a=LSM-nm[1]-nm[6]-nm[11]-nm[21];
    b=LSM-nm[25]-nm[20]-nm[15]-nm[5];
    if((0<a)&&(a<26)&&(a+b==PSM)){
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[16]=a; nm[10]=b;
            uflg[a]=1; uflg[b]=1;
            sstp11();
            uflg[b]=0; uflg[a]=0;}}
}
/* Level 4: */
/* Set n9=65-n6-n7-n8-n10 & n17 */
void sstp11(){
    short a,b,c,d;
    a=LSM-nm[6]-nm[7]-nm[8]-nm[10];
    b=LSM-nm[20]-nm[19]-nm[18]-nm[16];
    c=LSM-nm[3]-nm[15]-nm[16]-nm[22];
    d=LSM-nm[4]-nm[10]-nm[11]-nm[23];
    if((0<a)&&(a<26)){
        if((a==c)&&(b==d)&&(a+b==PSM)){
            if((uflg[a]==0)&&(uflg[b]==0)){
                nm[9]=a; nm[17]=b;
                uflg[a]=1; uflg[b]=1;
                if(nm[5]+nm[9]+nm[13]+nm[17]+nm[21]==LSM){sstp12();}
                uflg[b]=0; uflg[a]=0;}}}
}
/* Set n12=65-n2-n7-n17-n22 & n14 */
void sstp12(){
    short a,b,c,d;
    a=LSM-nm[2]-nm[7]-nm[17]-nm[22];
    b=LSM-nm[24]-nm[19]-nm[9]-nm[4];
    c=LSM-nm[4]-nm[8]-nm[16]-nm[25];
    d=LSM-nm[1]-nm[10]-nm[18]-nm[22];
    if((0<a)&&(a<26)){
        if((a==c)&&(b==d)&&(a+b==PSM)){
            if((uflg[a]==0)&&(uflg[b]==0)){
                nm[12]=a; nm[14]=b;
                uflg[a]=1; uflg[b]=1;
                if(nm[11]+nm[12]+nm[13]+nm[14]+nm[15]==LSM){sstp13();}
                uflg[b]=0; uflg[a]=0;}}}
}
/* Check Sums of 4 Entries */
void sstp13(void){
    short sm1, sm2;
    sm1=nm[5]+nm[6]+nm[12]+nm[18]+nm[24];
    sm2=nm[2]+nm[8]+nm[14]+nm[20]+nm[21];
    if((sm1==LSM)&&(sm2==LSM)){solrecord();}
}
/**/
/* Record the Solutions */
void solrecord(){
    short n;
    cnt++;
    for(n=1;n<26;n++){st[cnt][n]=nm[n];}
    d[cnt3][0]=cnt;
}

```



```

for(n=1;n<26;n++){d[cnt3][n]=nm[n];}
chksum();
cnt3++; if(cnt3==4){prsol(cnt3); cnt3=0;}
}
/**/
/* Print the Solutions */
void prsol(short x){
short l,l5,m,n;
for(n=0;n<x;n++){printf("%15d/|Rw|P1\\",d[n][0]);}
printf("\n");
for(l=0;l<5;l++){l5=l*5;
for(m=0;m<x;m++){
printf(" %3d%3d%3d%3d%3d%2d%2d\\",
d[m][l5+1],d[m][l5+2],d[m][l5+3],d[m][l5+4],d[m][l5+5],d[m][l+26],d[m][l+31]);
}
printf("\n");
}
for(n=0;n<x;n++){printf(" -----+-----");}
printf("\n");
for(m=0;m<x;m++){
printf(" |%2d|%2d|%2d|%2d|%2d|Column",
d[m][31],d[m][32],d[m][33],d[m][34],d[m][35]);}
printf("\n");
for(m=0;m<x;m++){
printf(" /%2d/%2d/%2d/%2d/%2d|P2Sums",
d[m][41],d[m][42],d[m][43],d[m][44],d[m][45]);}
printf("\n");
}
}
/**/
/* Check Sums of Rows, Columns and Pan-diagonals */
void chksum(){
d[cnt3][26]=nm[1]+nm[2]+nm[3]+nm[4]+nm[5];
d[cnt3][27]=nm[6]+nm[7]+nm[8]+nm[9]+nm[10];
d[cnt3][28]=nm[11]+nm[12]+nm[13]+nm[14]+nm[15];
d[cnt3][29]=nm[16]+nm[17]+nm[18]+nm[19]+nm[20];
d[cnt3][30]=nm[21]+nm[22]+nm[23]+nm[24]+nm[25];
d[cnt3][31]=nm[1]+nm[6]+nm[11]+nm[16]+nm[21];
d[cnt3][32]=nm[2]+nm[7]+nm[12]+nm[17]+nm[22];
d[cnt3][33]=nm[3]+nm[8]+nm[13]+nm[18]+nm[23];
d[cnt3][34]=nm[4]+nm[9]+nm[14]+nm[19]+nm[24];
d[cnt3][35]=nm[5]+nm[10]+nm[15]+nm[20]+nm[25];
/**/
d[cnt3][36]=nm[1]+nm[7]+nm[13]+nm[19]+nm[25];
d[cnt3][37]=nm[2]+nm[8]+nm[14]+nm[20]+nm[21];
d[cnt3][38]=nm[3]+nm[9]+nm[15]+nm[16]+nm[22];
d[cnt3][39]=nm[4]+nm[10]+nm[11]+nm[17]+nm[23];
d[cnt3][40]=nm[5]+nm[6]+nm[12]+nm[18]+nm[24];
d[cnt3][41]=nm[1]+nm[10]+nm[14]+nm[18]+nm[22];
d[cnt3][42]=nm[2]+nm[6]+nm[15]+nm[19]+nm[23];
d[cnt3][43]=nm[3]+nm[7]+nm[11]+nm[20]+nm[24];
d[cnt3][44]=nm[4]+nm[8]+nm[12]+nm[16]+nm[25];
d[cnt3][45]=nm[5]+nm[9]+nm[13]+nm[17]+nm[21];
}
}
/**/
/** Make the Set #1 of Prototype Squares of Order 5 */
/* Define Level 1 */
/* Set n1 and n25=PSM-n1 & n1<n25 */
void pstp01(){
short a,b;
for(a=1;a<13;a++){b=PSM-a;
if((uflg[a]==0)&&(uflg[b]==0)){
nm[1]=a; nm[25]=b;
uflg[a]=1; uflg[b]=1;
pstp02();
uflg[b]=0; uflg[a]=0;}
}
}

```

```

}
}
/* Set n23 and n3=PSM-n23 */
void pstp02(){
  short a,b;
  for(a=25;a>0;a--){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[23]=a; nm[3]=b;
      uflg[a]=1; uflg[b]=1;
      pstp03();
      uflg[b]=0; uflg[a]=0;}
    }
}
/* Set n20 and n6=PSM-n20 */
void pstp03(){
  short a,b;
  for(a=25;a>0;a--){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[20]=a; nm[6]=b;
      uflg[a]=1; uflg[b]=1;
      pstp04();
      uflg[b]=0; uflg[a]=0;}
    }
}
/* Set n12 and n14=PSM-n12 */
void pstp04(){
  short a,b;
  for(a=1;a<26;a++){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[12]=a; nm[14]=b;
      uflg[a]=1; uflg[b]=1;
      pstp05();
      uflg[b]=0; uflg[a]=0;}
    }
}
/* Set n9=LSM-n1-n23-n20-n12(>n1) and n17=LSM-n25-n3-n6-n14(>n1) */
void pstp05(){
  short a,b;
  a=LSM-nm[1]-nm[23]-nm[20]-nm[12];
  if((nm[1]<a)&&(a<26)){
    b=LSM-nm[25]-nm[3]-nm[6]-nm[14];
    if((a+b==PSM)&&(b>nm[1])){
      if((uflg[a]==0)&&(uflg[b]==0)){
        nm[9]=a; nm[17]=b;
        uflg[a]=1; uflg[b]=1;
        pstp06();
        uflg[b]=0; uflg[a]=0;}}}
}
/* Set n15(<n23) and n11=PSM-n15 */
void pstp06(){
  short a,b;
  for(a=nm[23]-1;a>0;a--){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[15]=a; nm[11]=b;
      uflg[a]=1; uflg[b]=1;
      pstp07();
      uflg[b]=0; uflg[a]=0;}
    }
}
/* Set n24 and n2=PSM-n24 */
void pstp07(){
  short a,b;
  for(a=25;a>0;a--){b=PSM-a;
    if((uflg[a]==0)&&(uflg[b]==0)){
      nm[24]=a; nm[2]=b;

```

```

        uflg[a]=1; uflg[b]=1;
        pstp08();
        uflg[b]=0; uflg[a]=0;}}
    }
}
/* Set n8=LSM-n1-n15-n24-n17 and n18=LSM-n25-n11-n2-n9 */
void pstp08(){
short a,b;
a=LSM-nm[1]-nm[15]-nm[24]-nm[17];
if((0<a)&&(a<26)){
    b=LSM-nm[25]-nm[11]-nm[2]-nm[9];
    if((a+b==PSM)&&(uflg[a]==0)&&(uflg[b]==0)){
        nm[8]=a; nm[18]=b;
        uflg[a]=1; uflg[b]=1;
        pstp09();
        uflg[b]=0; uflg[a]=0;}}
}
/* Define Level 2 */
/* Set n7=LSM-n20-n24-n11-n3 and n19=LSM-n6-n2-n15-n23 */
void pstp09(){
short a,b;
a=LSM-nm[20]-nm[24]-nm[11]-nm[3];
if((0<a)&&(a<26)){
    b=LSM-nm[6]-nm[2]-nm[15]-nm[23];
    if((a+b==PSM)&&(uflg[a]==0)&&(uflg[b]==0)){
        nm[7]=a; nm[19]=b;
        uflg[a]=1; uflg[b]=1;
        pstp10();
        uflg[b]=0; uflg[a]=0;}}
}
/* Set n21=LSM-n20-n2-n8-n14 and n5=LSM-n6-n24-n18-n12 */
void pstp10(){
short a,b;
a=LSM-nm[20]-nm[2]-nm[8]-nm[14];
if((0<a)&&(a<26)){
    b=LSM-nm[6]-nm[24]-nm[18]-nm[12];
    if((a+b==PSM)&&(uflg[a]==0)&&(uflg[b]==0)){
        nm[21]=a; nm[5]=b;
        uflg[a]=1; uflg[b]=1;
        pstp11();
        uflg[b]=0; uflg[a]=0;}}
}
/* Set n4=LSM-n15-n7-n21-n18 and n22=LSM-n11-n19-n5-n8 */
void pstp11(){
short a,b;
a=LSM-nm[15]-nm[7]-nm[21]-nm[18];
if((0<a)&&(a<26)){
    b=LSM-nm[11]-nm[19]-nm[5]-nm[8];
    if((a+b==PSM)&&(uflg[a]==0)&&(uflg[b]==0)){
        nm[4]=a; nm[22]=b;
        uflg[a]=1; uflg[b]=1;
        pstp12();
        uflg[b]=0; uflg[a]=0;}}
}
/* Set n16=LSM-n23-n7-n5-n14 and n10=LSM-n3-n19-n21-n12 */
void pstp12(){
short a,b;
a=LSM-nm[23]-nm[7]-nm[5]-nm[14];
if((0<a)&&(a<26)){
    b=LSM-nm[3]-nm[19]-nm[21]-nm[12];
    if((a+b==PSM)&&(uflg[a]==0)&&(uflg[b]==0)){
        nm[16]=a; nm[10]=b;
        uflg[a]=1; uflg[b]=1;
        pstp13();
        uflg[b]=0; uflg[a]=0;}}
}

```

```

}
/* The Last Sum-Checks */
/* Check Sums of the Lines */
void pstp13(){
    short sm1, sm2, sm3, sm4;
    sm1=nm[1]+nm[18]+nm[10]+nm[22]+nm[14];
    sm2=nm[25]+nm[8]+nm[16]+nm[4]+nm[12];
    sm3=nm[23]+nm[4]+nm[10]+nm[11]+nm[17];
    sm4=nm[3]+nm[22]+nm[16]+nm[15]+nm[9];
    if((sm1==LSM)&&(sm2==LSM)&&(sm3==LSM)&&(sm4==LSM)){protorecord();}
}
/**/
/* Record the Prototype Solutions */
void protorecord(){
    short n;
    cnt1++;
    for(n=1;n<26;n++){pn[cnt1][n]=nm[n];}
}
/**/
/* Sort the Prototypes */
void sortp(){
    long d1,d2,d3,d4,d5,d6;
    short m,mx,f;
    mx=cnt1-1;
    do{f=0;
        for(m=1;m<=mx;m++){
            d1=(((pn[m][1]*25+pn[m][2])*25+pn[m][3])*25+pn[m][4])*25+pn[m][5];
            d2=(((pn[m+1][1]*25+pn[m+1][2])*25+pn[m+1][3])*25+pn[m+1][4])*25+pn[m+1][5];
            if(d1>d2){exc(m); f=1;}
            else if(d1==d2){
                d3=(((pn[m][6]*25+pn[m][7])*25+pn[m][8])*25+pn[m][9])*25+pn[m][10];
                d4=(((pn[m+1][6]*25+pn[m+1][7])*25+pn[m+1][8])*25+pn[m+1][9])*25+pn[m+1][10];
                if(d3>d4){exc(m); f=1;}
                else if(d3==d4){
                    d5=(((pn[m][11]*25+pn[m][16])*25+pn[m][21])*25+pn[m][12])*25+pn[m][17];
                    d6=(((pn[m+1][11]*25+pn[m+1][16])*25+pn[m+1][21])*25+pn[m+1][12])*25+pn[m+1][17];
                    if(d5>d6){exc(m); f=1;}
                }
            }
        }
        mx--;
    }while(f>0);
}
/**/
void exc(short x){
    short n,d;
    for(n=0;n<26;n++){
        d=pn[x][n]; pn[x][n]=pn[x+1][n]; pn[x+1][n]=d;
    }
}
/**/
/** Make Prototype Squares 5x5 into Objects */
/** 'Do-it-After-the-Model Transformation' */
void damt5(short x){
    short n;
    for(n=1;n<26;n++){dnm[n]=nm[st[x][n]];}
}
/**/
/* Print the 4 Models */
void pr4mdl5(void){
    short l,l5,m;
    printf("%14d/P %12d/R1 %14d/P %12d/R3\n",d[0][0],d[1][0],d[2][0],d[3][0]);
    for(l=0;l<5;l++){l5=l*5;
        for(m=0;m<4;m++){

```

```

        if(m==2){printf(" ");}
        printf(" %3d%3d%3d%3d%3d",d[m][l5+1],d[m][l5+2],d[m][l5+3],d[m][l5+4],d[m][l5+5]);
    }
    printf("\n");
}
printf("%14d/P %12d/R2 %14d/P %12d/R4\n",d[4][0],d[5][0],d[6][0],d[7][0]);
for(l=0;l<5;l++){l5=l*5;
    for(m=4;m<8;m++){
        if(m==6){printf(" ");}
        printf(" %3d%3d%3d%3d%3d",d[m][l5+1],d[m][l5+2],d[m][l5+3],d[m][l5+4],d[m][l5+5]);
    }
    printf("\n");
}
}
/**/
/* Make the Multiple Models of Pan-diagonal MS55 */
void mulmdl(short x1, short x2, short x3, short x4){
    short f,f2,m,n;
    f=0;
    for(m=1;m<=576;m++){
        f2=f*2; d[f2][0]=m;
        for(n=1;n<26;n++){nm[n]=pn[m][n]; d[f2][n]=nm[n];}
        damt5(1);
        n=findoldn();
        if((n==x1)&&(ln[n]==0)){
            d[f2+1][0]=n; ln[n]++;
            for(n=1;n<26;n++){d[f2+1][n]=dnm[n];}
            f++;}
        if((n==x2)&&(ln[n]==0)){
            d[f2+1][0]=n; ln[n]++;
            for(n=1;n<26;n++){d[f2+1][n]=dnm[n];}
            f++;}
        if((n==x3)&&(ln[n]==0)){
            d[f2+1][0]=n; ln[n]++;
            for(n=1;n<26;n++){d[f2+1][n]=dnm[n];}
            f++;}
        if((n==x4)&&(ln[n]==0)){
            d[f2+1][0]=n; ln[n]++;
            for(n=1;n<26;n++){d[f2+1][n]=dnm[n];}
            f++;}
        if(f==4){pr4mdl5(); break;}
    }
}
/**/
/* Cooperative Reconstruction of Objects with multiple DAMT */
void muldamt(short x1, short x2, short x3, short x4){
    short m,n;
    for(m=1;m<=cnt1;m++){
        d[0][0]=m;
        for(n=1;n<26;n++){nm[n]=pn[m][n]; d[0][n]=nm[n];}
        damt5(x1);
        n=findoldn(); if((n==0)|| (ln[n]>0)){goto nextstp;}
        d[1][0]=n; ln[n]++;
        for(n=1;n<26;n++){d[1][n]=dnm[n];}
        damt5(x2);
        n=findoldn(); if((n==0)|| (ln[n]>0)){goto nextstp;}
        d[2][0]=n; ln[n]++;
        for(n=1;n<26;n++){d[2][n]=dnm[n];}
        damt5(x3);
        n=findoldn(); if((n==0)|| (ln[n]>0)){goto nextstp;}
        d[3][0]=n; ln[n]++;
        for(n=1;n<26;n++){d[3][n]=dnm[n];}
        damt5(x4);
        n=findoldn(); if((n==0)|| (ln[n]>0)){goto nextstp;}
        d[4][0]=n; ln[n]++;
    }
}

```

```

        for(n=1;n<26;n++){d[4][n]=dnm[n];}
        cnt2++;
        pr5ans();
nxtstp: ;
    }
}
/**/
/* Print the Answers */
void pr5ans(){
    short l,l5,m,n;
    if(md1==1){printf(" [%2d]%9d/P %14d/R1%14d/R2%14d/R3%14d/R4\n",
        cnt2,d[0][0],d[1][0],d[2][0],d[3][0],d[4][0]);}
    else{printf(" [%2d]%9d/P %14d/R5%14d/R6%14d/R7%14d/R8\n",
        cnt2,d[0][0],d[1][0],d[2][0],d[3][0],d[4][0]);}
    for(l=0;l<5;l++){l5=l*5;
        for(m=0;m<5;m++){
            printf(" ");
            for(n=1;n<6;n++){
                printf("%3d",d[m][l5+n]);}
            if(m==0){printf(" ");}
            if(m<4){printf(" ");}
        }
        printf("\n");
    }
}
/**/
/* Find its Old Number */
short findoldn(){
    short int fn,cn,m,n;
    fn=0;
    for(m=1;m<=cnt;m++){cn=0;
        for(n=1;n<26;n++){if(dnm[n]==st[m][n]){cn++;}else{break;}}
        if(cn==25){fn=m; break;}
    }
    if(cn<25){fn=0;}
    return fn;
}
/**/
/* Reconstruction Monitor */
void prmon(){
    short m;
    printf(" ?? : %2d\n",ln[0]);
    for(m=1;m<=cnt1;m++){
        if(m%16==1){printf("%4d: ",m);}
        printf("%2d",ln[m]);
        if(m%16==0){printf("\n");}
    }
}
/**/

```