The Magic Square as a Benchmark: Comparing MIP to Improved GA and to a High Performance Minimax AI Algorithm

José B. Fonseca
Department of Electrical Engineering and Computer Science
Faculty of Sciences and Technology of New University of Lisbon
Quinta da Torre, 2829-516 Caparica, Portugal
Phone: +351-93-4935940, Fax: +351-21-4262406
email:jbfo@unl.pt

ABSTRACT: We found the magic square a simple problem with a very rich combinatorics: there are $(n^2)!$ manners to fill the *n*x*n* matrix with integers between 1 and n^2, without repetitions, but only very few of them are magic squares. For n=4, generating all the 16! permutations, we found 7040 magic squares and 549504 *relaxed magic squares*. In the literature many people believe that the number of magic squares of order four is 880, but in fact these are the *canonical* magic squares from which can be generated all the other by rotation and transposition or reflection. So we use the magic square as a benchmark to compare mathematical programming, namely mixed integer programming (MIP), to Genetic Algorithms (GAs), that we will show are much more powerful to solve these kind of discrete combinatorial explosive problems. Then we got the idea to compare GAs to the human being, and we developed a prototypes of a game where the objective was to get the *relaxed magic square* in a minimum number of changes between two elements of the matrix. This game could be used for management training since in management problems we often have a limited budget, a lot of constraints and objectives to reach that could be formulated in a similar matrix form. Finally we developed an artificial intelligence minimax algorithm that imitates a human solving the magic square and show that in most cases its performance, in terms of number permutations, is better than the performance of the GA algorithm.

KEYWORDS: Magic Square as a Benchmark, MIP Solution of a Magic Square, AI Minimax Algorithm that Solves the Magic Square, Improved Evolutionary Algorithm for the Solution of the Magic Square.

INTODUCTION

In the literature 'magic square' has various meanings. Here we consider the addition magic square which is a square matrix *n*x*n* with integer elements between 1 and n^2 and where the sums of the elements of all lines, columns and the two main diagonals are equal to the magic sum which is given by [2]

$$MagicSum=\Sigma a_{ij}/n= n (n^2+1)/2 \qquad (1)$$

Since we found very difficult to reach a magic square by simple changes of pairs of elements, even for n=3, for game development we only consider the *Relaxed Magic Square* where we don't impose that the sum of the elements of the two main diagonals be equal to the magic sum. By the same argument in the magic rectangles all lines must have the sum [2]

$$LineMagicSum=\Sigma a_{ij} /n= n(m\ n +1) / 2 \qquad (2)$$

and all columns must have the sum [2] given by

$$ColumnMagicSum=\Sigma a_{ij} /m=m(m\ n +1) / 2 \qquad (3)$$

Sun in [6] showed that the magic rectangle has solution only when m and n are both odd or both even, but never n=m=2, that is a magic square 2x2 that is simple to show that it has no solution [1]. In the literature exists also another proposal of magic square, the addition-multiplication magic square [3],[4] that has also the restriction that the products of all elements of all lines, columns and the two main diagonals are equal to the magic product. This is a much more difficult and challenging problem and in the near future we will consider the *Relaxed Addition-Multiplication Magic Square*.

THE MAGIC SQUARE AS A BENCHMARK

Our initial motivation to study the magic square was to compare MIP to GAs in the solution of explosive combinatorial optimisation problems to make a decision on which method to use to solve a even much more explosive combinatorial problem. [5] reported that using a 1998 Pentium PC, may be at 200MHz clock, obtained a 100x100 magic square in about 2h, which would mean about 20 minutes in a 1GHz modern PC…and our PC at 1GHz, using the Cplex algorithm to implement MIP, took about 4.3 days to obtain a 7x7 relaxed magic square!…which is a much more simple problem than obtaining a 7x7 magic square! In table 1 you can find the solution obtained and you can verify that as a matter of fact the sums of the main diagonals are not equal to the magic sum 175.

26	2	43	17	38	10	39
23	19	12	42	13	45	21
31	48	3	14	7	44	28
1	49	40	29	25	4	27
18	36	16	33	11	41	20
30	15	37	32	34	22	5
46	6	24	8	47	9	35

Table 1. The relaxed 7x7 magic square obtained by the Cplex algorithm after *4.3 days* of computation on a PC at 1GHz. You can verify that the sums of the main diagonals are not equal to 175 and that this solution is completely different from the 2 solutions presented in tables 2 and 3.

COMPARING GA'S TO HUMANS

Using the first game based on the relaxed magic square we obtained a 7x7 one in 83 moves departing from a sequential square (see table 2) and in 57 moves departing from a random square (see table 3), which is surely a better performance than the MIP's one but we don't believe that we would be capable to obtain a 100x100 magic square!

Move 83, Objective=175

49	29	1	4	21	35	36
16	9	10	46	40	13	48
8	2	17	45	47	31	18
25	23	24	28	26	22	27
14	34	43	7	30	32	15
19	41	38	12	6	39	20
44	37	42	33	5	3	11

Error=98

[old value,new value]=[9 2]

Move 84

49	29	1	4	21	35	36
16	2	10	46	40	13	48
8	9	17	45	47	31	18
25	23	24	28	26	22	27
14	34	43	7	30	32	15
19	41	38	12	6	39	20
44	37	42	33	5	3	11

Error=0

Table 2. The 7x7 relaxed magic square obtained by the author in 83 moves from a sequential initialisation

Move 1

47	12	30	24	44	38	23
2	40	22	31	39	45	36
9	20	46	21	4	18	1
8	11	14	37	42	26	33
41	34	19	25	35	16	10
27	29	32	17	15	28	43
48	13	3	49	7	5	6

Error=10696

[old value,new value]=[28 27]

(...)

Move 57

2	35	24	23	39	16	36
17	5	31	29	42	45	6
41	20	46	25	4	18	21
10	12	14	49	44	38	8
43	30	19	1	7	27	48
15	40	32	37	13	3	22
47	33	9	11	26	28	34

Error=338

[old value,new value]=[13 26]

Move 58

2	35	24	23	39	16	36
17	5	31	29	42	45	6
41	20	46	25	4	18	21
10	12	14	49	44	38	8
43	30	19	1	7	27	48
15	40	32	37	26	3	22
47	33	9	11	13	28	34

Error=0

Table 3. The 7x7 relaxed magic square obtained by the author in 57 moves from a random initialisation

COMPARING GA'S TO MINIMAX AI ALGORITHM

Before describing in detail our artificial intelligence randomized minimax algorithm let's see how it obtains a magic square of order four from random initialization and in appendixes A and B we show how myself did obtain a 4x4 magic square from a random and sequential filling, respectively:

Move 1, Objective=34
1 9 3 13
8 6 14 12
11 5 2 15
7 10 4 16
Error=886
15 <--> 2
Move 2, Objective=34
1 9 3 13 26
8 6 14 12 40
11 5 15 2 33
7 10 4 16 37
Error=301
[…]
12 <--> 13
Move 11, Objective=34
2 12 5 14 33
9 6 10 8 33
16 3 15 1 35
7 13 4 11 35
34 34 34 34 0
Error=4
2 <--> 10
Move 12, Objective=34
10 12 5 14 41
9 6 2 8 25
16 3 15 1 35
7 13 4 11 35
42 34 26 34 0
Error=388
4 <--> 15
Move 13, Objective=34
10 12 5 14 41
9 6 2 8 25
16 3 4 1 24
7 13 15 11 46
42 34 26 34 0
Error=575
3 <--> 13
Move 14, Objective=34
10 12 5 14 41
9 6 2 8 25
16 13 4 1 34
7 3 15 11 36
42 34 26 34 0
Error=275

7 <--> 2
Move 15, Objective=34
10 12 5 14 41
9 6 7 8 30
16 13 4 1 34
2 3 15 11 31
37 34 31 34 0
Error=105
12 <--> 6
Move 16, Objective=34
10 6 5 14 35
9 12 7 8 36
16 13 4 1 34
2 3 15 11 31
37 34 31 34 0
Error=45
5 <--> 10
Move 17, Objective=34
5 6 10 14 35
9 12 7 8 36
16 13 4 1 34
2 3 15 11 31
32 34 36 34 0
Error=30
5 <--> 7
Move 18, Objective=34
7 6 10 14 37
9 12 5 8 34
16 13 4 1 34
2 3 15 11 31
34 34 34 34 0
Error=18
3 <--> 6
Move 19, Objective=34
7 3 10 14 34
9 12 5 8 34
16 13 4 1 34
2 6 15 11 34
34 34 34 34 0
Error=0

Table 4. Example of a run of AI minimax algorithm finding a magic square of order four from random initial filling.

Each change corresponds to the permutation that minimizes the error *over a set of random number of cycles of random chosen pairs of numbers*. When is detected a situation where the chosen pair of numbers to be permuted is equal to the previous, then next change corresponds to the change that, now, *maximizes* the error *over a set of random number of cycles of random chosen pairs of numbers*. This prevents the oscillation and *stagnation in local minimum.*

COMPUTATIONAL RESULTS

In table 5 we compare the AI minimax algorithm to an improved evolutionary algorithm for n=3..20, in terms of number permutations. In most cases our algorithm is much more efficient. Note that the number of permutations is obtained, in both cases, in only one run, and not averaged over a set of successive runs.

n	AI Minimax Alg	Improved GA
3	7	121
4	19	57
5	185	129
6	126	48
7	108	3645
8	48	1824
9	84	456
10	95	2985
11	2023	2766
12	320	823
13	3330	562
14	1017	3510
15	1111	893
16	415	7762
17	191	753
18	420	1922
19	625	4507
20	613	6215

Table 5. Computational results of one run of each algorithm in terms of number of permutations.

CONCLUSIONS

We showed that although very simple, our AI minimax algorithm is very powerful. Nevertheless its runtime is much greater, since each permutation results from a minimization/maximization over a relatively great number of cycles, and the calculation of the new error associated to a given permutation is time consuming. We also showed that the magic square is a good benchmark to compare optimisation algorithms since it has an explosive combinatory that increases with (n^2!).

REFERENCES

[1] Ball , W. W. R., 1963, "Mathematical Recreations and Essays", Macmillan, New York, USA.

[2] Evans, A. B., 1996, "Magic Rectangles and Modular Magic Rectangles", Journal of Statistical Planning and Inference,51, 171-180.

[3] Horner, N. N., 1952, "Addition-Multiplication Magic Squares", Scripta Math., 18, 300-303.

[4] Horner, N. N., 1955, "Addition-Multiplication Magic Squares of Order 8", Scripta Math., 21, 23-27.

[5] Rechenberg, I., 2000, "Case Studies in Evolutionary Experimentation and Computation", Comput. Methods Appl. Mech. Engrg., 186, 125-140.

[6] Sun, R. G., 1990, "Existence of Magic Rectangles", Nei Mongol Daxue Xuebao Ziran Kexue, 21, 10-16.