3P-02

分類写像による魔方陣の数え上げについて

山田穂高 † 山本修身 ‡
名城大学理工学研究科情報工学専攻 † 名城大学理工学部情報工学科 ‡

1 はじめに

本稿では $n \times n$ 魔方陣 [1]（これを n 次魔方陣と呼ぶ）の個数の数え上げについて考える．5 次までの魔方陣の個数は，1 桁目まで正確に数え上げられているが，6 次魔方陣については，正確な個数は知られていない．概数についてはマルマフ連鎖モンテカルロ法を用いて計算されている [2].

本稿では，6 次魔方陣の厳密な個数の数え上げを目指して，効率的な n 次魔方陣の数え上げ手法について考える．特に後述する ZDD (Zero-suppressed Decision Diagram)[3] を用いた厳密な個数の計算方法について考える．

ZDD を構成し，個数を得る方法は一般に高速で効率的であると考えられるが，現実には 5 次以上の魔方陣についてはメモリの制約から ZDD を構成することが困難となる．実際，64G バイトのメモリをもつ計算機では，ノード数の爆発により計算ができていない．

そこで，本稿では一気に魔方陣の総数を数え上げる巨大な ZDD を作成するのではなく，魔方陣をある写像によりいくつかの類に分類してから数え上げる方法を考えた．この場合，それぞれの類ごとに ZDD を用いて数え上げることができれば，それらを合計することで魔方陣の総数を数え上げることになる．

この方法を用いることにより，5 次魔方陣，6 次魔方陣について，いくつかの類に対してメモリに乗せることのできる ZDD を構成することができた．しかし，類の個数や 1 つの類の計算に要する時間が大きいことから，総数をすべて計算することは今のところできていない．

2 魔方陣の定義と ZDD による表現

まず本節では，魔方陣を以下のように定義し，その基本的性質について述べる．

定義 1 n 次魔方陣とは，0 から n^2-1 の整数を $n \times n$ の格子状に並べたもののうち，それぞれの行，列，対角線についての和がすべて等しい配置のことである．[1]

このとき，以下の性質が成り立つ．

性質 1 n 次魔方陣に対して，

(1) 各列，行，対角線上の要素の和は，$n(n^2-1)/2$ となる．
(2) ある配置から，90° 回転，および，反転させた配置は，元の配置を含め 8 つ存在し，常に異なる配置となる．

ここで，性質 1-(1) の和のことを魔法和，1-(2) の配置のことを対称形と呼ぶ．魔方陣の個数を数える場合，一般的に対称形を同じ 1 つの魔方陣として扱うが，本稿では対称形を区別する．また，定義 1 より，n 次の魔方陣は以下の性質を持つ．

性質 2 n 次魔方陣の n^2 個の任意の整数の並びについて，

(1) 0 から n^2-1 までの整数が必ず 1 つ存在し，かつ，2 つ以上存在しない．
(2) 各行，列，対角線の和が等しい．

ここで，性質 2-(1), 性質 2-(2) の条件をそれぞれ順列条件，魔方和条件と呼ぶ．

また，魔方陣に ZDD を用いるためにビットパターンで表現する必要がある．定義より，魔方陣は n^2 個の整数の並びとして表現できる．魔方陣をビットパターンとして表現する方法としては 2 種類のものが考えられる．まず，1 つの方法としてそれぞれの整数を n^2 ビットの 2 進数を用いて，$100\cdots00$, $010\cdots00$, ..., $000\cdots01$ と表現することにより，n^4 ビットで表現することができる．これに対し，もう 1 つの方法として，それぞれの数を 2 進数として，$\lceil \log_2(n^2-1) \rceil$ ビットで表現することができる．この場合，全部で $O(n^2 \log n)$ ビットで表現可能となる．ここでは，これらの方法を，それぞれ n^2 ビット表現法，$\log n^2$ 表現法と呼ぶ．本稿では，変数の少ない $\log n^2$ 表現法の方を用いる．

上記のビットパターンによる表現方法を用いて，ZDD によって魔方陣を表現するには前述の順列条件と魔方和条件を満たす関数を構成すればよい．使用する計算機に搭載されているメモリーの量や計算量に制約がなければ，あらゆる魔方陣は ZDD でそのまま表現することができる．その ZDD が表現する解集合の個数を計算することにより，魔方陣の総数を得ることができる．しかし 64G バイトのメモリーが搭載されている PC 上で，この方法でうまくいくのは 4 次魔方陣までで，5 次以上の魔方陣についてはそのまま計算することは困難である．

そこで，本稿では魔方陣の総数を直接数え上げる ZDD を作成するのではなく，ある方法によりいくつかの類に類別し，それぞれの類内の魔方陣の個数を ZDD で計算して，合計することで魔方陣の総数を数え上げる．本稿では，その分類方法として分類写像というものを導入する．

3 分類写像

分類写像は魔方陣を分類するために用いる写像のことである．分類写像を用いた場合，魔方陣の総数 X は，

$$X = \sum_{p \in \mathrm{Im} f} |f^{-1}(p)| \tag{1}$$

のように求めることができる．つまり，あらかじめ $f(p)$ を全て求めておけば，式 (1) を用いることで魔方陣の総和を求めることができる．分類写像として次の 4 種類のものを考える．

(i) m を法とする数字に置き換える分類写像
(ii) m 個の連続した数字の集合に区分けして，それぞれの区分ごとに数字を割り当てる分類写像
(iii) 特定の m 個の変数の数字により分類する分類写像
(iv) 特定の m 個の数字がどの変数に存在するかにより分類する分類写像

以下に各分類写像の像の特徴について述べる．

On Enumeration of Magic Squares by a Classification Mapping
† Hodaka Yamada, Division of Information Engineering, School of Science and Technology, Meijo University
‡ Osami Yamamoto, Department of Information Engineering, Faculty of Science and Technology, Meijo University
[1] 魔方陣は 1 から n^2 までの整数の並びとして定義される事が多いが，ここでは 2 進数によって効率的に表現するために前述の定義を採用する．

表 1　n 次の魔方陣ごとの m を法とする分類写像の像の個数.

法 m	3×3	4×4	5×5	6×6
2	1	102	5,348	2,226,112
3	24	400	449,164	6,375,089,124
4	4	3,584	8,331,664	1,284,326,015,232
5	8	2,336	66,262,960	—

まず (i) の分類写像の像である条件は，通常の魔方陣と同様に順列条件と魔法和条件を用いる．この分類写像における順列条件および魔法和条件は，m 個の任意の整数の並びを x_i $(i = 0, \dots n^2 - 1)$ とすると，以下の 2 つのようになる：

(I)　$x_i \in \{0, \dots, m-1\}$ であり，かつ，$\{0, \dots, m-1\}$ のそれぞれの個数が $\{0, \dots, n^2 - 1\} \bmod m$ とした場合のそれぞれの個数と等しい.

(II)　各行，列，対角線上の和と魔方和が，法 m に対してすべて合同である.

表 1 に各条件下での分類写像の像の個数を示す.

次に (ii) の分類写像の像である条件は，順列条件のみを用いる．この分類写像における順列条件は，m 個の任意の整数の並びを x_i $(i = 0, \dots n^2 - 1)$ とすると以下のようになる：

(I)　$x_i \in \{0, \dots, m-1\}$ であり，かつ，$\{0, \dots, m-1\}$ の各個数が $\{0, \dots, n^2\}$ を区分けされた整数の個数と等しい.

魔法和条件に関しては区分けの仕方により条件が変わるため，決まった魔法和条件を設けることは不可能である．分類写像の像の個数は，順列条件を満たしたす順列の個数に等しい.

(iii)，および (iv) の分類写像に関しては，0 から $n^2 - 1$ の数字を用いた m 個の順列であれば像となり得る．また，(iii) および (iv) の像の個数はともに ${}_{n^2}P_m$ である．本稿では像の個数が比較的少ない (i) の分類写像を用いて実験を行った.

4　分類写像を用いた魔方陣の数え上げ

前節の (i) の分類写像の像を用いた原像の求め方を説明する．ここでも順列条件と魔法和条件を用いる．ここで，$\{0, \dots, n^2 - 1\} \bmod m$ 内の $\{0, \dots, m-1\}$ の個数を $M_j (j = 0, \dots, m-1)$ とすると，順列条件と魔法和条件は以下のようになる：

(I)　M_j 個の任意の整数の並び $x_i (i = 0, \dots, M_j - 1)$ が $x_i \in \{0, \dots, M_j - 1\}$ であり，かつ $x_i \neq x_l$ $(i \neq l)$.

(II)　各行，列，対角線上の和がすべて $(\mathcal{S}_n - s_k)/m$ に等しい.

$\log n^2$ 表現法を用いて ZDD によって像に対する原像を表現するには，魔方陣の総数を数え上げる場合と同様に前述の順列条件と魔方和条件を満たす関数を構成すればよい.

5　分類写像を用いた効果

実験は，メモリ 64GB，CPU6-Core Intel Xeon × 2 の MacPro を用い，言語処理系は Java version1.8.0_05 を使用した.

3 次，4 次魔方陣に対して，ZDD を作成する過程の最大ノード数および最大ノード数の平均をまとめたものを表 2，表 3 に示す．この表から分類写像を用いたことによって，大幅にノード数を減らせたことがわかる．また，計算時間については分類写像を用いなかった場合と大きな変化がなかった.

また，5 次魔方陣に対しては $m = 3$，6 次魔方陣に対しては $m = 8$ の分類写像から原像を数え上げる ZDD を作成できた．そこで，5 次魔方陣に関しては $m = 3, 4$，6 次魔方陣に関しては $m = 8, 9$ の分類写像の像のうち，ランダムに選出した 100 個に対し原像を数え上げる ZDD を作成する過程の最大ノード数と原像の個数の平均および標準偏差をまとめたものを表 4，表 5 に示す.

表 2　3 次魔方陣を数え上げる際のノード数.

mod	最大ノード数		計算時間 [s]
	最大値	平均	合計
分類なし	10,948		0.246
2	775	775	0.081
3	272	208	0.367

表 3　4 次魔方陣を数え上げる際のノード数.

mod	最大ノード数		計算時間 [s]
	最大値	平均	合計
分類なし	1,601,108		10.246
2	66,809	41,554	11.394
3	41,554	6,643	7.688

表 4　5 次魔方陣を数え上げる際のノード数および原像の個数.

mod	最大ノード数		原像の個数		計算時間 [s]
	平均	標準偏差	平均	標準偏差	平均
3	2.7×10^7	1.4×10^7	5,048	1,368	362.614
4	2.0×10^6	1.2×10^7	260	79	15.501

表 5　6 次魔方陣を数え上げる際のノード数および原像の個数.

mod	最大ノード数		原像の個数		計算時間 [s]
	平均	標準偏差	平均	標準偏差	平均
8	2.8×10^7	2.0×10^7	1,919	598	548.721
9	5.1×10^6	4.8×10^6	304	117	42.248

表 5 から，分類写像 (i) を用いた場合，6 次魔方陣の総数を数え上げるには，膨大な時間がかかることになり，不可能である.

6　まとめ

本稿では 6 次魔方陣を数え上げるために，総数を一度に数え上げる ZDD を作成するのではなく，分類写像を用い，いくつかの類に類別しその類に対して魔方陣を数え上げる ZDD を作成する方法について考えた．それにより，ZDD を作成することができなかった 5 次，6 次魔方陣に対しても ZDD が作成できるようになった．しかし，実験結果から現段階の方法だけでは現実的な時間で計算することができない．更なる工夫としては，(1) の分類写像以外の分類写像を用いた場合，および併用，ZDD の作成方法などが考えられる．また，ZDD 以外の数え上げ方法に，今後この分類写像を用いた場合にどのような効果が得られるかについても調べる予定である.

参考文献

[1] Levitin，A.　and Levitin，M.，2011 : *Algorithmic Puzzle*. Oxford University Press，NewYork.

[2] Pinn, K., Wieczerkowski, C.: Number of magic squares from parallel tempering Monte Carlo. *Int. J. Mod. Phys. C*, vol. 9, Issue 4, pp. 541-546, 1998.

[3] Knuth, D. E.: *The Art of Computer Programming*. vol. 4, Fascicle 1, Generating Bitwise Tricks & Techniques, Binary Decision Diagrams. Addison-Wesley, Boston, 2005.