社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－MPS－68 (16)
2008／3／5

緩やかな制約によるディスクレパンシの検討

河村 泰之

愛媛大学 教育学部

概要 ディスクレパンシは数論で発達した考え方である．連続測度と離散測度の差を考えることで，"ズレ"を定量的に表す尺度である．最近では情報科学の分野でもディスクレパンシの概念が重要視されてきている．情報科学では主に離散空間だけで考えることも多く，ディスクレパンシでもまた離散測度だけで考えた概念が生み出された．例えば，ディスクレパンシをハーフトーニングに応用するために作られた最適化問題がある．ある行列の中のすべての小領域で要素の合計をとり，最大値と最小値の差を小さくすることを目的としている．本研究では，ハーフトーニングに応用されたディスクレパンシの基準に焦点を当て，新しい考え方を検討する．

A Discrepancy under Loose Constraints

Yasuyuki KAWAMURA

Faculty of Education
Ehime University

Abstract The discrepancy is developed in number theory. We calculate the difference between a continuous measure and a discrete one and use it as a quantitatively scale for the disagreement. In computer science, the concept of a discrepancy is becoming important recently and we yield the concept of a discrepancy which is calculated from a discrete measure only. For instance, there is an optimization problem to apply a discrepancy to a halftoning. The purpose of that is to minimize the difference between a maximum summation and a minimum one of elements in each small region in a matrix. In this paper, we focus such a discrepancy which is applied to a halftoning and discuss a new idea.

1 はじめに

計算機科学の各分野で discrepancy の考え方を応用した技術が利用されてきている．例えば，数値積分では，通常のモンテカルロ法では収束の遅い問題に対して，discrepancy に基づく高速化アルゴリズムが研究されている [1]．計算幾何学の分野では，近年，ハーフトーニングの性能評価に discrepancy が利用されている [4]．コンピュータグラフィクス技術としては discrepancy のレンダリングへの応用が実用化されている．

数学で生まれた discrepancy の概念が実用的な場面で利用されるようになり，discrepancy にもいくつか定義が生まれた [2, 3]．どの定義も分布の仕方を表す尺度として用いられ，応用する場合は discrepancy が低くなる性質を利用することが多い．本稿では，新

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

図 1: $n=5, k=2$ のとき，k-discrepancy が 8 の例 ($max = 19+15+5+17 = 56, min = 3+16+25+4 = 48$).

しい discrepancy の概念を提案し，その性質を考察する.

2 k-discrepansy

まず，Aronov et al. がハーフトーニングに応用するために定式化した最適化問題をとりあげる [5]:

> 整数 $1,\cdots,n^2$ を $n\times n$ 行列に並べる．そのとき，すべての $k\times k$ 領域内の合計を均一にする ($k=1,\cdots,n$).

この問題で discrepancy を考えてみる．正整数 n に対して $1,\cdots,n^2$ がそれぞれ一回ずつ要素として現れるような $n\times n$ 行列を M とする．k を n 以下の正整数として，行列 M 内のすべての $k\times k$ 領域の要素の合計を計算し，最大値と最小値の差を k-discrepancy とする．ただし，index は mod をとるので，行列の上下，左右はそれぞれ巡回する．図 1 に，$n=5, k=2$ としたときの k-discrepancy が 8 となる例を示した.

Aronov et al. の最適化問題で $k=2$ のときは，Kawamura が k-discrepancy を 0 にする必要十分条件を示し，それ以外の (0 にできない) 場合については k-discrepancy を低くするような良い行列を heuristic に構成する方法を提案し，実験的に正当性を示した [6].

k-discrepancy は本来 すべての正整数 $k(<n)$ に対して考えるべきであるが，Aronov et al. [5] も Kawamura [6] も $k=2$ に限定した実験で成果をまとめている．すべての $k<n$ に対して，$n\times n$ 行列の中の $k\times k$ 領域の要素の合計を計算することは容易ではないからである．著者の持つ Core Solo CPU 1.33GHz, メモリ 2GB のパソコンを用いて，$n=201$ の 2-discrepancy [1] を計算したところ出力まで約 5 時間を要した．(k を大きくとって実験をすることはあまり現実的ではない．) そこで，現実的に計算を省略するための新しい discrepancy が必要となり，本研究に着手した．以下に，k-discrepancy の k を大きく計算するための新しい discerepancy を提案する.

3 提案

話を簡単にするため，1 次元で次の問題を考える:

> $[1,n]$ のすべての整数がただ一度ずつ現れるような数列 $\{a_i\}$ を考える ($i=1,\cdots,n$). この中から k 個の数を選んで合計を均一にする.

k-discrepancy で $k=n$ とするのと同じで，すべての組み合わせをとると実用上の意味がないので，適切に組合せを選ぶ必要がある．数列 $\{a_i\}$ 内でそれぞれの a_i は同じ回数 v だけ選ばれるように工夫して何度か選び，k 個の要素の合計を計算し，合計の最大値と最小値の差を discrepancy とする.

図 2 に提案した discrepancy のアイディアを示した．$n=8, k=4$ のとき，すべての組み合わせを取ると ${}_8C_4 = 70$ 個の合計の計算が必要となるが，図 2 では 14 回の合計だけ求めている．それでいて，ある意味での均一性を保ちつつ，数列の discrepancy を計算できていることがわかる．ここで均一性とは，各 a_i が合計のために選ばれる回数を a_i の価値 (value) と考え，a_i が用いられた回数 v_i を統一することで価値を平等に考えていることを指している．図 2 の場合，どの a_i も 7 回選ばれている.

実際には，次のような行列を考えることに相当する : 要素が 0 か 1 のどちらかをとる $m\times n$ 行列 $A=$

[1] 文献 [6] で計算した最大の数が $n=201$ の 2-discrepncy である.

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	
4	5	3	1	2	7	6	8	sum
4	5	3	1					<u>13</u>
4	5			2	7			18
4	5					6	8	23
4		3		2		6		15
4		3			7		8	22
4			1	2			8	15
4			1		7	6		18
	5	3		2			8	18
	5	3			7	6		21
	5		1	2		6		14
	5		1		7		8	21
		3	1	2	7			13
		3	1			6	8	18
				2	7	6	8	<u>23</u>

図 2: $n = 8, k = 4$ での提案する discrepancy が 10 となる例 ($max - min = 23 - 13 = 10$).

$(a_{j,i})$ において，各行ベクトルの要素の和 (1 の個数) をそれぞれ $\sum_{i=1}^{n} a_{j,i} = k_j$，各列ベクトルの要素の和 (1 の個数) をそれぞれ $\sum_{j=1}^{m} a_{j,i} = v_i$ とする．

このような行列を考え，ある k, v がとれて，

$$k = k_1 = \cdots = k_m \text{ かつ } v = v_1 = \cdots = v_n \quad (1)$$

という制約を満たすとき，その行列を緩やかな制約によるディスクレパンシ行列と定義し，本稿では単にディスクレパンシ行列と呼ぶ．全ての要素が 0 の行列と全ての要素が 1 の行列は制約 (1) を満たすのではあるが，ディスクレパンシ行列から除外する．

さて，いくつか trivial な性質が観察できる．

観察 1 すべての n, k, v に対して，ディスクレパンシ行列が存在するとは限らない．

観察 2 ディスクレパンシ行列が存在しても一意性はない．

観察 3 任意のディスクレパンシ行列に対して，$m = nv/k$.

観察 4 制約 *(1)* を満たす行列の行の交換，列の交換を行っても制約 *(1)* を満たす．

観察 5 (自明なディスクレパンシ行列) 任意の $n \times n$ 行列 *($n = m$ のとき)* に対して，どんな正整数 $k < n$ をとってもディスクレパンシ行列は存在する．

次にディスクレパンシ行列の存在性について考える．

定理 1 n と m が互いに素でないとき，ディスクレパンシ行列が存在する．

証明 n と m の共通の約数を p として，$p \times p$ のディスクレパンシ行列を作る．*(少なくとも自明なものは存在する．)* この行列を並べるだけで，$n \times m$ 行列のディスクレパンシ行列ができる．

□

定理 2 m と n が互いに素なら，ディスクレパンシ行列は存在しない．

証明 $m > n$ のとき，$m \times n$ 行列のディスクレパンシ行列が存在することと $m \times n$ 行列から $n \times n$ 行列を取り除いて残る $(m - n) \times n$ 行列で存在することは必要十分である．($m < n$ の場合は $m \times m$ 行列を取り除く．)

$$\begin{array}{cccc}
a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\
a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n,1} & a_{n,2} & \cdots & a_{n,n} \\
\hline
a_{n+1,1} & a_{n+1,2} & \cdots & a_{n+1,n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m,1} & a_{m,2} & \cdots & a_{m,n}
\end{array}$$

図 3: $m \times n$ 行列 $(m > n)$ から $n \times n$ 行列を取り除く様子．

これを再帰的に繰り返すと狭義単調減少で 1×2 行列 *(もしくは 2×1 行列)* に帰着する．1×2 行列は制約 *(1)* を満たさない． □

本研究で提案するディスクレパンシの利用では，観察 3 に注目する．当然，$m < {}_nC_k$ であるから discrepancy の高速計算として意義がある．${}_nC_k$ 通りの

すべての合計を計算するには $O({}_nC_k) \times O(k)$ の計算が必要で，$k = n/2$ 辺りが最も複雑である．

4 自明ではない構成法

自明なディスクレパンシ行列は，必ず $v = k$ に限定されてしまう．v が k の倍数となるように増やすことは容易である (これも自明であろう) が，実用的には意味が無い．

自明ではないディスクレパンシ行列を作る方法を一つ紹介する．行列 $(a_{j,i})$ を，n が奇数で $k = \lfloor n/2 \rfloor$ の $n \times n$ ディスクレパンシ行列 $(i, j = 1, \cdots, n)$ だとすると，次の手順に従うことにより自明でないディスクレパンシ行列を得る (図 4 参照)．

1. すべての $1 \leq j \leq n$ に対して，$a_{j,0} = 1$ とする．
2. すべての $0 \leq i \leq n, 1 \leq j \leq n$ 対して，
$$b_{j,i} = \begin{cases} 1 & if \ \ a_{j,i} = 0 \\ 0 & if \ \ a_{j,i} = 1 \end{cases}$$
とする．
3. $a_{j,i}, b_{j,i}$ をつなげて $2n \times (n+1)$ 行列を作る．

このように作ると，$m' \times n'$ 行列で $k = \frac{n'}{2}$ で，$v \neq k$ であるようなディスクレパンシ行列を作ることができる．(このとき，$v = n' - 1$ である．)

$$\begin{array}{ccccc}
a_{1,0} & a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\
a_{2,0} & a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
a_{n,0} & a_{n,1} & a_{n,2} & \cdots & a_{n,n} \\
b_{1,0} & b_{1,1} & b_{1,2} & \cdots & b_{1,n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
b_{n,0} & b_{n,1} & b_{n,2} & \cdots & b_{n,n}
\end{array}$$

図 4: 自明でない $m' \times n'$ 行列の作り方 $(m' = 2n, n' = n+1)$.

5 おわりに

新しい discrepancy を提案した．1 次元に限定して論じたが，自然な拡張によって高次元の定義が可能である．ハーフトーニングに応用するなら，三次元を考えると良いだろう．このような discrepancy を考えることで，k-discrepancy より広い範囲で"ズレ"を計算できるようになるので，k-discrepancy のときより大きな k で実験が可能となる．実際には，提案した discrepancy は k-discrepancy の拡張になっていることがわかる．

参考文献

[1] 室田一雄編: **離散構造とアルゴリズム** *IV*, pp.99–126, 近代科学社 (1995).

[2] B.Chazelle: *The Discrepancy method*, Cambridge University Press (2000).

[3] M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf: *Computational Geometry*, Springer (1997).

[4] T.Asano, N.Katoh, K.Obokata, T.Tokuyama: *Matrix rounding under the Lp-discrepancy measure and its application to digital halftoning*, Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms, pp.896–904 (2002).

[5] B.Aronov, T.Asano, Y.Kikuchi, S.Nandy, S.Sasahara, and T.Uno: *A Generalization of magic squares with applications to digital halftoning*, Proc. ISAAC 2004, LNCS 3341, 89-100, Springer (2004).

[6] A.Kawamura: *Some Improvements on Low-Discrepancy Matrices*, Korea-Japan Joint Workshop on Algorithms and Computation (2005).