

# 魔女の最適調合問題— 非減少部分列の最適化に基づく多次元ソーティング問題

藤原直紀<sup>1</sup>徳山豪<sup>2</sup>

## 1 はじめに

実数や整数などの全順序集合の $n$ 個の要素からなるデータ列が与えられたとき、それを非減少列になるように並べ替える問題は、計算理論では基本問題であるソーティング（整列問題）として効率的なアルゴリズムが数多く与えられている[4]。一方で、全順序が与えられていないときのデータ列をどのように並べるか、あるいは構造化するかという問題は、定式化は応用によって異なり、統一的な手法は与えられていない。

特に、与えられる入力が $d$ 次元のベクトルの列であるときは、多次元ソーティングといい、様々な応用で重要な問題である。実用的なデータベースの設計では、データの主キーとなる次元を決め、そのキーをもとにソーティングを行うのが一般的である。また、次元に優先度がついていれば、辞書式順序による整列も広く利用されている。一方で、そのような主キーや次元の優先度が与えられていない場合は、多次元データ検索を目的にした、木構造などを用いたデータ構造化、すなわち多次元データ構造に関する理論研究が広くおこなわれており、計算幾何学([8], [9])の大きなテーマとされている。しかしながら、多次元データ構造は、データの並べ替えとは異なった問題の定式化であり、整列問題の直接の一般化としては別の定式化が望まれるところである。

本論文では、多次元ソーティングを $d \times n$ の行列の列の並べ替えと見なし、行の非減少部分列の長さの和が最大となるように並べ替える問題を提案し、その計算複雑度を考察する。非減少部分列の選び方にはいくつものバリエーションが考えられるが、本論文では、行の初めから一要素ずつ見ていったときにグリーディに構成できる非減少部分列、すなわち FF(First-Fit)非減少部分列に焦点を当てる。

### 1.1 問題の提起

全順序集合の要素からなる列 $\mathbf{a} = (a_1, a_2, \dots, a_n)$ が与えられたとき、その FF(First-Fit) 非減少部分列とは、列の先頭から順に見ていったときにグリーディに構成される非減少部分列である。たとえば、 $\mathbf{a} = (3, 1, 3, 5, 2, 4, 6)$ のとき、 $\mathbf{a}$ の FF 非減少部分列は $(3, 3, 5, 6)$ となる。ここで、 $\mathbf{a}$ の要素に対する効用関数 $\varphi$ を考える。 $\varphi$ は $\mathbf{a}$ の要素から $\mathbb{R}^+$ への関数である。 $\mathbf{a}$ に対する効用 $\Phi(\mathbf{a})$ は $\mathbf{a}$ の要素の効用の和で定義する。上の例では、 $\Phi(\mathbf{a}) = \varphi(3) + \varphi(3) + \varphi(5) + \varphi(6)$ となる。各行が全順序であるような $d \times n$ の行列 $A$ を考える。 $A$ に対する効用 $\Phi(A)$ を各行の効用の和で定義する。行列の列の順列 $\sigma$ を $A$ に作用させ、効用を最適化するような $\sigma A$ を見つきたい。 $\Phi(\sigma A)$ を最大化/最小化する $\sigma$ を求める問題をそれぞれ効用最大化順序付け問題 (MaxCA) および効用最小化順序付け問題 (MinCA) という。

### 1.2 問題の背景

この問題の着想はクラフティングゲームにある。クラフティングゲームとは、複数の素材アイテムを元に新しい一つのアイテムをクラフトするゲームである。マインクラフトなどが代表例にあげられる。このクラフトのシステムにおいて、得られる成果物がクラフトする素材の順番によって、たとえば品質の良し悪しが決まるようにすれば、よりゲーム性が増すのではないかと筆者の考えがあった。具体例を挙げて説明するには長くなるため、詳細は[6]を参考にされたい。

### 1.3 本論文の成果

本論文の成果は以下のものである。

(1) ソーティング問題と最大値問題の多次元化を与える新しい最適化問題の提案。

(2)  $d$  (ベクトルの次元) が定数のときの多項式時間アルゴリズムの設計。

- 各行に同一要素が重複して現れない場合は、入力となるベクトル列が与えられたとき、重み付き有向グラフを構成し、そのグラフ上で最長および最短経路問題として説明できる。
- 行に同一の要素を含む場合では、グラフの構成に工夫が必要となる。その差異と手法を詳しく説明する。

## 2 FF 非減少部分列の最適化に基づく多次元ソーティング問題

本論文では、多次元ソーティング問題に対して組合せ最適化の観点から定式化し、この問題がベクトルの次元数が定数のときには多項式時間で解けることを示す。

### 2.1 数学的定式化

全順序集合 $X$ とは、順序 $<$ が定義された集合で、相異なる要素 $x \neq y \in X$ に対して $x < y$ または $y < x$ が成り立つものを指す。したがって、集合 $X$ の部分集合 $S$ に対して、その最大要素 $\max\{x \in S\}$ が常に存在する。非負整数集合 $\mathbb{Z} \geq 0$ が典型的な例であり、以降の議論における全順序集合としては $\mathbb{Z} \geq 0$ であると仮定して考えればその一般化は容易である。全順序集合の要素からなる列 $\mathbf{a} = (a_1, a_2, \dots, a_n)$ が与えられたとき、その $i$ 番目までの接頭部分列における最大要素を $\text{Max}(\mathbf{a}, i) = \max_{j \leq i} a_j$ で定義する。 $\text{Max}(\mathbf{a}, i) = a_i$ であるとき、 $a_i$ は列 $\mathbf{a}$ の接頭最大要素であるという。 $M(\mathbf{a})$ を $\mathbf{a}$ のすべての接頭最大要素から構成される部分列とする。つまり、その要素を追加したときに非減少部分列になれるならば追加し、そうでないならば破棄するというアルゴリズムで得られる部分列である。列の初めから処理し、処理順に部分列を拡張するためにファーストフィット (FF) アルゴリズムのように動作することから、 $M(\mathbf{a})$ は $\mathbf{a}$ の FF 最大非減少部分列であるという。 $\mathbf{a}$ の要素について、効用と呼ばれる非負値関数 $\varphi$ を導入し、 $\Phi(\mathbf{a}) = \sum_{e \in M(\mathbf{a})} \varphi(e)$ とする。各行が全順序であるような $d \times n$ の行列 $A$ を考える。効用関数 $\varphi$ は $A$ の要素の集合に対して定義されていると仮定する。 $A(i)$ を $A$ の $i$ 番目の行ベクトルとする。 $\Phi(A) = \sum_{1 \leq i \leq d} \Phi(A(i))$ を $A$ の効用と呼ぶ。 $S_n = \{1, 2, \dots, n\}$ のすべての順列の集合とする。 $\sigma \in S_n$ が与えられたとき、 $\sigma A$ は $A$ の列のインデックスを $\sigma$ で並べ替えて得られる行列となる。すなわち、 $A$ の $(i, j)$ 要素は、 $\sigma A$ において $(i, \sigma(j))$ 要素になる。

#### 効用最適化順序付け問題

$d$ は定数とする。

- (1) MaxCA:  $\sum_{1 \leq i \leq d} \Phi(\sigma A(i))$ を最大化する $\sigma \in S_n$ を見つける。
- (2) MinCA:  $\sum_{1 \leq i \leq d} \Phi(\sigma A(i))$ を最小化する $\sigma \in S_n$ を見つける。

例 1.  $2 \times 6$ の行列 $A = \begin{pmatrix} 123456 \\ 623541 \end{pmatrix}$ を考え、

すべての $x$ に対して $\varphi(x) = 1$ とする。入力での目的関数値は $\Phi(A) = 6 + 1 = 7$ となることが分かる。ここでは、赤色で強調している数値が非減少部分列を示している。このとき、置換として $\sigma = (2, 3, 4, 5, 1, 6)$ を考えると、

$\sigma A = \begin{pmatrix} 234516 \\ 235461 \end{pmatrix}$ となり $\Phi(\sigma A) = 5 + 4 = 9$ である。

一方、 $\tau = (1, 6, 2, 3, 4, 5)$ に対しては、

$\tau A = \begin{pmatrix} 162354 \\ 612345 \end{pmatrix}$ となり $\Phi(\tau A) = 2 + 1 = 3$ である。

効用関数を恒等関数 $\varphi(x) = x$ に変更したとすると、それぞれ $\Phi(A) = 21 + 6 = 28$ ,  $\Phi(\sigma A) = 20 + 16 = 36$ , および $\Phi(\tau A) = 7 + 6 = 13$ となる。

この例では、どちらの効用関数であっても、 $\sigma$ と $\tau$ がそれぞれMaxCAとMinCAの最適解である。しかし、一般には効用関数 $\varphi$ が異なれば、最適解が同じになるとは限らない。

### 3 定数次元の場合の解法

入力として与えられる行列 $A$ の各行に同一要素を含む場合とそうでない場合では、問題の取り扱いが変わる。ここでは、前者を**非重複ケース**のMaxCA/MinCAといい、後者を**一般**のMaxCA/MinCAという。ただし、特に断りがない場合、一般のMaxCA/MinCAを単にMaxCA/MinCAという。

#### 3.1 問題の重み付き有向グラフによる表現

$d$ が定数のとき、最大化および最小化問題のそれぞれに対する動的計画法が構成できる。具体的には、重み付き有向グラフ上の最長/最短経路問題として説明できる。

便宜的に、各行を構成する各全順序集合が、その行に現れる要素よりも小さい要素を持つと仮定し、それらを共有記号\*で表記する。

2つの $d$ 次元ベクトル $u$ と $v$ に対して、すべての $j = 1, 2, \dots, d$ に対して $u_j \geq v_j$ が成り立つとき、 $u \geq v$ と書く。また、 $u \geq v$ かつ $u \neq v$ のとき、 $u > v$ と書く。2つの $d$ 次元ベクトル $u$ と $v$ が与えられたとき、そのマックス和 $u \oplus v$ とは次元ごとの最大要素である。つまり、 $u \oplus v$ の $i$ 番目の要素は $\max\{u_i, v_i\}$ となる。 $A$ のいくつかの列ベクトルからなる部分行列 $B$ が与えられたとき、その指標 $\Lambda(B)$ は、 $B$ の $i$ 行目の最大要素を $i$ 番目にもつ $d$ 次元ベクトルである。言い換えると、 $\Lambda(B)$ は $B$ のすべての列ベクトルのマックス和である。空の部分集合 $\emptyset$ に対しては恣意的に、 $s = \Lambda(\emptyset) = (*, *, \dots, *)$ と定義する。 $V_i$ は $A$ の $i$ 行目に現れる要素の集合とし、 $V' = V_1 \times V_2 \times \dots \times V_d$ を各 $V_i$ の直積とする。 $V = V' \cup \{s\}$ と定義する。明らかに、 $\Lambda(B)$ は $V$ の要素となり、高々 $n^d + 1$ 通りの指標が存在する。図6は $d = 2, n = 3$ を仮定した場合に生成されるグラフの頂点集合である。重み付き有向グラフ $G(A) = (V, E)$ を次のようにして構成する。各列ベクトル $c \in A$ と各ベクトル $v \in V$ に対して、 $v$ から $v' = v \oplus c$ への有向辺 $e \in E$ を作る（ただし、 $v' \neq v$ ）。ある $j \in \{1, 2, \dots, d\}$ に対して $v'_j > v_j$ のとき、 $v'_j$ を $e$ に関する新規エントリーという。 $e$ の重み $w(e)$ を $e$ に関する新規エントリーの効用の和で定義する。

**補題 1.**  $G(A)$ は有向非巡回グラフとなり、あるパスが与えられたとき、そのパスが表す列ベクトルの部分集合に、同一の列ベクトルが二度現れることはない。

[証明] 省略。

#### 3.2 非重複ケースに対するアルゴリズム

**定理 1.** 非重複ケースのMaxCAおよびMinCAはそれぞれ、 $G(A)$ の $s$ から $\Lambda(A)$ への重み付き最長/最短パスの長さとも一致する。したがって、非重複ケースのMaxCAおよびMinCAはそれぞれ $O(n^{d+1})$ 時間で解ける。

[証明] 補題1より、あるパスが行列の列ベクトルの並びに一致する。また、グラフ $G(A)$ の頂点の個数は $O(n^d)$ 、辺の本数は $O(n^{d+1})$ である。非巡回グラフであるので、下記のように、最長パスおよび最短パスが $O(n^{d+1})$ 時間で求まる。まず、 $G(A)$ を図8のようにトポロジカルソートし、頂点のインデックスをトポロジカル順序に付けなおす。トポロジカルソートにかかる計算時間は $O(|V| + |E|) = O(n^{d+1})$ である[4]。トポロジカル順序に基づいて、各頂点 $v_i$ までの最長パスの長さ $d(v_i)$ を動的計画法で求める。 $v_i \in V$ に隣接する $v_j$ より後方の各頂点 $u$ に対して、 $d(v_i) + w((v_i, u)) > d(u)$ ならば、

$d(u) = d(v_i) + w((v_i, u))$ とする。各頂点 $v_i$ から出る有向辺の本数は高々 $n$ である。ゆえに、各頂点での隣接する頂点への最長パスの更新も高々 $n$ 回となり、全体では $O(|V|) \times O(n) = O(n^{d+1})$ 時間かかる。よって、計算量は合計で $O(n^{d+1})$ となる。

頂点をトポロジカル順序の順番で処理していくため、ある頂点を訪れるまでにその頂点へのどのパスも探索が終了し、アルゴリズムの正当性については保証される。

最短パスについては、条件式の大小符号を入れ替えることで同様に計算可能である。

最適解の順列 $\sigma$ は最適パスから対応する列ベクトルの並びを構成することができる。ネットワーク上の辺が元のインスタンスの $d$ 次元ベクトルと対応するので、線形時間で構成できる。ただし、最短パスの長さは $n$ よりも短くなる可能性がある。そのため、その場合は、あらかじめ決められた方法（たとえば、安定な順序）で残りの列ベクトルの順序を定めることで $\sigma$ を構成できる。□

#### 3.2 一般のケースに対するアルゴリズム

一般のケースでは、各行に同一の要素を持つ可能性があるために、非重複ケースのように対応するグラフを適切に構成することが出来ない。そこで、**遅延評価法(deferred evaluation)**という手法を使い、辺の重みを適当に調整し、対応するグラフが構成できる。したがって、同様に $O(n^{d+1})$ 時間で非巡回有向グラフが構成でき、定理1と同様に動的計画法のアルゴリズムを適用することで問題を解くことができる。

**定理 2.** 一般のMaxCAは $O(n^{d+1})$ 時間で解ける。

その手法を用いた具体的なグラフの構成（辺の重みの調整）と、パスから最適解を構成する方法の詳細については[6]を参照されたい。

また、繊細な工夫を施すことで、一般のMinCAも同様に解くことができる。

**定理 3.** 一般のMinCAは $O(n^{d+1})$ 時間で解ける。

### 4 おわりに

本論文では、多次元ソーティング問題の新しい定式化として、行列の列の置換で、各行における非減少列を用いた最適化を提案し、 $d$ が定数である場合の多項式時間アルゴリズムを与えた。本報告で述べたものは、この問題に対する最初の研究である。

### 参考文献

- [1] Y. Azar, L. Gamzu, X. Yin, Multiple Intents Re-ranking, 41st STOC:669-678, 2009.
- [2] E. Balas, The Prize Collecting Traveling Salesman Problem and Its Applications, Chapter 14 of The Traveling Salesman Problem and Its Variations (G. Gutin, A.P. Punnen, ed.), 663-695, Springer Verlag, 2007.
- [3] S. Bspamyatnikh, M. Segal, Enumerating longest increasing subsequences and patience sorting, Information Processing Letters 76: 7-11, 2000.
- [4] T. Cormen, C. Leiserson, R. Rivest, C. Stein, Introduction to Algorithms, 3rd edition, MIT Press, 2009.
- [5] U. Feige, L. Lov'asz, P. Tetali, Approximating Min Sum Set Cover, Algorithmica, 40: 219-234, 2004.
- [6] N. Fujiwara, T. Tokuyama, Arranging Columns of Matrices Optimizing Non-decreasing Subsequences of Rows, working paper.
- [7] S. Iwata, P. Tetali, P. Tripathi, Approximating Minimum Linear Ordering Problems, Proc. RANDOM 2012 :206-217, 2012.
- [8] K. Mehlhorn, Multi-dimensional Searching and Computational Geometry (Data Structures and Algorithms 3), ETACS Monographs on Theoretical Computer Science, Vol.3, Springer Verlag, 1984.
- [9] F. Preparata, M. I. Shamos, Computational Geometry, An Introduction, Springer Verlag, 1985.