

## 二つの2次元ベクトル列間の1:1対応に関するパラメトリックな方法

## — 変形ボトルネック割当て問題

02103250 東京農工大学 \* 嘉村 友作 KAMURA Yuusaku  
 01401940 東京農工大学 中森 眞理雄 NAKAMORI Mario

## 1. はじめに

$n$  個の元から成る二つの2次元ベクトル列間の1:1対応の中で、対応づけられるベクトルの成分和の最大値を最小にする問題を考える。二つのベクトル列を

$$A_1 = \{a_{11}, a_{12}, \dots, a_{1n}\}, \\ A_2 = \{a_{21}, a_{22}, \dots, a_{2n}\}$$

と表わし、ベクトル  $a_{1j}, a_{2k}$  を、それぞれ、成分で  $(a_{1j}^{(1)}, a_{1j}^{(2)}), (a_{2k}^{(1)}, a_{2k}^{(2)})$  と表わす。

この問題は、辺の重み  $c_{jk}$  を

$$c_{jk} = \max\{a_{1j}^{(1)} + a_{2k}^{(1)}, a_{1j}^{(2)} + a_{2k}^{(2)}\}$$

として、節点数  $2n$ 、辺の数  $n^2$  の線形ボトルネック割当て問題 (Linear Bottleneck Assignment Problem) として解くことができる。線形ボトルネック割当て問題は、これまでに広く研究され、いくつかの多項式時間アルゴリズムが提案されている [1]。現在最も速いアルゴリズムは、 $m$  を辺の数、 $n$  を節点数として、 $O(m\sqrt{n} \log n)$  のアルゴリズムである [2]。

ここでは、ボトルネック割当て問題に変形せずに扱うことによって、スカラーの場合には、和の最大値を最小、かつ、最大値と最小値の差を最小にする1:1対応が存在することを示す。次に、このスカラー列の場合の方法を2次元ベクトル列の場合へ拡張するパラメトリックな方法を提案し、計算実験例を示す。

## 2. スカラーの場合

$a_{1j}, a_{2k}$  の次元が1の場合、すなわちスカラーの場合に対する最適な組み合わせ法は、次の定理から導かれる。今、 $a_{1j}, a_{2k}$  はスカラーなので、それぞれ、 $a_{1j}, a_{2k}$  と表わす。

**定理 1**  $n$  個の元から成る二つの非負スカラー列を、それぞれ、 $A_1 = \{a_{11}, a_{12}, \dots, a_{1n}\}$ 、 $A_2 = \{a_{21}, a_{22}, \dots, a_{2n}\}$  と表わす。ここで、 $0 \leq a_{11} < a_{12} < \dots < a_{1n}$ 、 $0 \leq a_{21} < a_{22} < \dots < a_{2n}$  を満たすとする。このとき、 $A_1$  の元  $a_{1j}$  と  $A_2$  の元  $a_{2k}$  と

の間の1:1対応を表わす置換  $\pi: \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}$  について、

$$\pi(1) = n, \pi(2) = n-1, \dots, \pi(n) = 1$$

を満たす置換  $\pi'$  は

$$a_{1j} + a_{2\pi'(j)}, (j = 1, 2, \dots, n)$$

の中の最大値を最小にし、かつ、最小値を最大にする。

□

定理1から、次のアルゴリズム1が導かれる。

## アルゴリズム1

Step 1:  $A_1$  の元  $a_{1j}$  を昇順に整列する。

Step 2:  $A_2$  の元  $a_{2k}$  を昇順に整列する。

Step 3:  $A_1$  の元  $a_{1j}$  の中で  $m$  番目に小さい元と  $A_2$  の元  $a_{2k}$  の中で  $m$  番目に大きい元とを順に組み合わせる。

□

アルゴリズム1による対応づけによって、 $n$  個の和の中の最大値が最小になり、かつ、最大値と最小値の差が最小になるという二つの目的が満たされる。その計算量は  $O(n \log n)$  である。

ボトルネック割当て問題と同様に、辺の重みが与えられた場合に、辺の重みの最大値と最小値の差を最小にする置換を求める問題は平均化割当て問題 (Balanced Assignment Problem) として知られ、 $O(n^4)$  のアルゴリズムが提案されている [3]。

## 3. 2次元ベクトル列間の対応づけへの拡張

アルゴリズム1の2次元の場合への拡張を考える。そのために、ベクトルの第1成分と第2成分の内分の値を使う。今、 $0 \leq t \leq 1$  を満たすパラメータ  $t$  を導入し、 $a_{1j} = (a_{1j}^{(1)}, a_{1j}^{(2)})$ 、 $a_{2k} = (a_{2k}^{(1)}, a_{2k}^{(2)})$  に対して、

$$u_j(t) = t \cdot a_{1j}^{(1)} + a_{1j}^{(2)}, \\ v_k(t) = t \cdot a_{2k}^{(1)} + a_{2k}^{(2)}$$

とする。

相異なる  $j_1, j_2$  に対する  $a_{1j_1} = (a_{1j_1}^{(1)}, a_{1j_1}^{(2)})$ 、 $a_{1j_2} = (a_{1j_2}^{(1)}, a_{1j_2}^{(2)})$  において、 $a_{1j_1}^{(1)}$  と  $a_{1j_2}^{(1)}$  の大小関係と  $a_{1j_1}^{(2)}$

と  $a_{1j_2}^{(2)}$  の大小関係が異なるとき,  $u_{j_1}$  と  $u_{j_2}$  は交点を持ち, 鉛直方向の位置関係が変わることに注意する.

そこで,  $t$  を 0 から 1 まで変化させたときに得られる鉛直方向の位置関係によって  $u_j(t), v_k(t)$  それぞれを順序づけ, Algorithm 1 の Step 3 と同様の方法で対応づける.

例えば  $n = 4$  の場合に  $u_j(t)$  が図 1 のようになるならば,

$$\begin{aligned} \text{Seq}A_{11} &= (a_{11}, a_{12}, a_{13}, a_{14}), 0 \leq t < t_1; \\ \text{Seq}A_{12} &= (a_{11}, a_{13}, a_{12}, a_{14}), t_1 < t < t_2; \\ \text{Seq}A_{13} &= (a_{13}, a_{11}, a_{12}, a_{14}), t_2 < t < t_3; \\ \text{Seq}A_{14} &= (a_{13}, a_{12}, a_{11}, a_{14}), t_3 < t \leq 1; \end{aligned} \quad (1)$$

の 4 つの順序を得る.

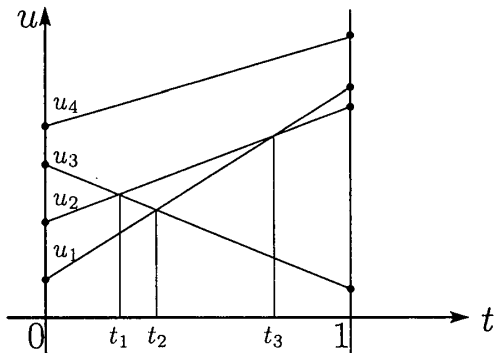


図 1:  $n = 4$  の場合の直線  $u_j(t)$  の例

同様にして,  $t_1 < t'_1 < t_2 < t'_2 < t_3$  を満たす  $t'_1, t'_2$  に対して,  $A_2$  の順序

$$\begin{aligned} \text{Seq}A_{21} &= (a_{21}, a_{22}, a_{23}, a_{24}), 0 \leq t < t'_1; \\ \text{Seq}A_{22} &= (a_{21}, a_{22}, a_{24}, a_{23}), t'_1 < t < t'_2; \\ \text{Seq}A_{23} &= (a_{21}, a_{24}, a_{22}, a_{23}), t'_2 < t \leq 1; \end{aligned} \quad (2)$$

が得られたとする. このとき, まず (3) のように順序どうしを組み合わせ, Algorithm 1 の Step 3 と同様に  $\text{Seq}A_{1j}$  の  $i$  番目の元と  $\text{Seq}A_{2k}$  の  $n - i + 1$  番目の元を対応づける.

$$\begin{aligned} \text{Seq}A_{11} &\text{---} \text{Seq}A_{21}, 0 \leq t < t_1; \\ \text{Seq}A_{12} &\text{---} \text{Seq}A_{21}, t_1 < t < t'_1; \\ \text{Seq}A_{12} &\text{---} \text{Seq}A_{22}, t'_1 < t < t_2; \\ \text{Seq}A_{13} &\text{---} \text{Seq}A_{22}, t_2 < t < t'_2; \\ \text{Seq}A_{13} &\text{---} \text{Seq}A_{23}, t'_2 < t < t_3; \\ \text{Seq}A_{14} &\text{---} \text{Seq}A_{23}, t_3 < t \leq 1. \end{aligned} \quad (3)$$

組合せの中で, 最大値が最小となるものを解とする.

順序  $\text{Seq}A_{1j}, \text{Seq}A_{2k}$  は鉛直方向の平面走査で得られる. また,  $n$  本の線分の交点数は最大  $n(n-1)/2$  個であり, ベクトルの和を求める手間は  $O(n)$  であることから, 総計算量は  $O(n^2) \cdot O(n) = O(n^3)$  である.

#### 4. 計算実験

$A_1, A_2$  それぞれの元の個数  $n$  を 50 とする.  $a_{1j}^{(1)}, a_{1j}^{(2)}, a_{2k}^{(1)}, a_{2k}^{(2)}$  はすべて平均 10, 分散 1 の正規分布に従うとし,  $a_{1j}^{(1)}$  と  $a_{1j}^{(2)}$  の相関係数が  $\rho_1$ ,  $a_{2k}^{(1)}$  と  $a_{2k}^{(2)}$  の相関係数が  $\rho_2$  のデータを用意した.

また, 各相関係数に対し, 50 組の  $A_1$  と  $A_2$  を準備し, 1 組ごとにパラメータ  $t$  を用いる方法を適用して, 最大値に関する厳密解と誤差率, 最大値と最小値の差に関する厳密解との誤差率の二つを求めた. そして, 50 組全体の誤差率の平均を求めた. これを表 1 に示す.

表 1: 計算結果

$\rho_1$	$\rho_2$	最大値の	最大値と最小値の
		平均誤差率	差の平均誤差率
-1.0	-1.0	0.000000000	0.000000000
-1.0	1.0	0.089778494	0.230184026
-0.75	0.75	0.034917730	0.197994244
-0.5	-0.5	0.042582044	1.073767903
-0.5	0.5	0.028167365	0.391141008
0.0	-0.5	0.042677539	0.762184699
0.0	0.0	0.037160605	0.850565009
0.0	0.5	0.028760850	0.440332209
0.5	0.5	0.022625064	0.466443535
0.75	0.75	0.011582654	0.319167014

パラメータ  $t$  を動かして得られる順序の中で, 最大値が最小になる組合せを解とした. その解はじゅうぶん良い精度であると言える. しかしながら, 最大値と最小値の差の最小化も同時に成り立たせることは不可能なようである. この点については, 多目的最適化という視点から検討することが今後の課題である.

#### 参考文献

- [1] R. E. Burkard and E. Çela, Linear Assignment Problem and Extensions, in: D. -Z. Du and P. M. Pardalos (eds.), *Handbook of Combinatorial Optimization, Vol.4*, Kluwer Academic Publisher, 1999.
- [2] H. N. Gabow and R. E. Tarjan, Algorithms for Two Bottleneck Optimization Problems, *J. of Algorithms* 9, pp.411-417, 1988.
- [3] S. Martello, W.R.Pulleyblank, P. Toth and D. de Werra, Balanced Optimization Problems, *Oper. Res. Lett.* 3, pp.275-278, 1984.