

通信路容量を計算する新しいアルゴリズムの研究

A Study of New Algorithm to Calculate the Channel Capacity

佐武 拓斗*

Takuto Sabu

中川 健治†

Kenji Nakagawa

渡部 康平†

Kohei Watabe

Abstract— The channel capacity of a discrete memoryless channel is defined as the maximum of the mutual information between input and output sources. We have proposed a new algorithm to calculate the channel capacity, which we call the projection algorithm. However, by extensive simulations, we found that, in some placements of the row vectors of the channel matrix, our method yields incorrect values of the channel capacities. In this paper, we investigate the conditions of the placement of the vectors that yields incorrect values for developing a correct algorithm.

Keywords— channel capacity, projection algorithm, smallest enclosing circle

1 はじめに

通信は通信路を介してやり取りされ、情報を符号化して送信する。よい符号化とは通信速度が速く、雑音に強いものである。良い符号化を作る際に、通信路容量が重要になる。通信路容量とは任意に小さい誤りで伝送可能なデータの最大量であり、通信速度の限界値を表している。通信路符号化定理より、通信路容量 C を超えない通信速度であれば、誤り率をほぼ 0 で情報を伝送できる符号化が存在するため、この通信路容量を求めることで良い符号化を求める目標値となる。本論文では、この通信路容量を求める新しいアルゴリズムを提案する。提案するアルゴリズムを「射影アルゴリズム」と呼び、幾何学問題である最小包含円問題と通信路容量問題の構造がどちらも $\min \max$ 問題の解であることに注目し、この構造を利用し求める手法である [1]。しかしながら、この手法は未完成である。この射影アルゴリズムで様々な問題を解き、通信路容量及び、通信路容量を達成する出力分布が正しく得られる割合について求め結果、全ての問題を求めることができなかった。そこで、求められない問題の原因を突き止めるために、点配置からその原因を得られないかと考えた。本論文では、求められなかった問題の点配置の特徴について調査した。その結果を示す。

* 〒 940-2188 新潟県長岡市上富岡町 1603-1 長岡技術科学大学 工学研究科電気電子情報工学専攻, Department of Electrical, Electronics and Information Engineering, Nagaoka University of Technology, Nagaoka, Niigata, 940-2188, Japan. E-mail: s133125@stn.nagaokaut.ac.jp

† 〒 940-2188 新潟県長岡市上富岡町 1603-1 長岡技術科学大学, Nagaoka University of Technology, Nagaoka, Niigata, 940-2188, Japan. E-mail: nakagawa@nagaokaut.ac.jp

1.1 通信路行列と通信路容量の定式化

X を入力情報源とし、 Y を出力情報源とする離散的無記憶通信路 $X \rightarrow Y$ を考える。入力アルファベットを $\{x_1, \dots, x_m\}$ とし、出力アルファベットを $\{y_1, \dots, y_n\}$ とする。 x_i を送信したときに y_j が受信される条件付確率 P_{ij} を

$$P_{ij} = P(Y = y_j | X = x_i), i = 1, \dots, m, j = 1, \dots, n$$

とし、行ベクトル P_i を

$$P_i = (P_{i1}, \dots, P_{in}), i = 1, \dots, m$$

とする。通信路行列 Φ は

$$\Phi = \begin{pmatrix} P_1 \\ \vdots \\ P_m \end{pmatrix} = \begin{pmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & & \vdots \\ P_{m1} & \cdots & P_{mn} \end{pmatrix} \quad (1)$$

と定義される。入力分布全体を $\bar{\Delta}^m = \{\lambda = (\lambda_1, \dots, \lambda_m) | \lambda_i \geq 0, i = 1, \dots, m, \sum_{i=1}^m \lambda_i = 1\}$, 出力分布全体を $\bar{\Delta}^n = \{Q = (Q_1, \dots, Q_n) | Q_j \geq 0, j = 1, \dots, n, \sum_{j=1}^n Q_j = 1\}$ と定義する。

$\lambda \in \bar{\Delta}^m$ を入力分布とするときの出力分布 $Q \in \bar{\Delta}^n$ が $Q = \lambda \Phi$, すなわち、 $Q_j = \sum_{i=1}^m \lambda_i P_{ij}, j = 1, \dots, n$ と定義され、相互情報量が $I(\lambda, \Phi) = \sum_{i=1}^m \sum_{j=1}^n \lambda_i P_{ij} \log P_{ij}/Q_j$ と定義される。このとき、通信路容量 C が

$$C = \max_{\lambda \in \bar{\Delta}^m} I(\lambda, \Phi) \quad (2)$$

と定義される。

$Q = (Q_j), Q' = (Q'_j) \in \bar{\Delta}^n$ に対して KL 情報量 $D(Q||Q')$ が $D(Q||Q') \equiv \sum_{j=1}^n Q_j \log Q_j/Q'_j$ と定義される [4]。 C は KL 情報量によって次のように表すこともできる [4]。

$$C = \min_{Q \in \bar{\Delta}^n} \max_{1 \leq i \leq m} D(P_i||Q) \quad (3)$$

(3) 式の $D(P_i||Q)$ をユークリッド距離に置き換えると、幾何学問題である最小包含円問題の問題となる。最小包含円問題を

$$\min_{Q \in \mathbb{R}^n} \max_{1 \leq i \leq m} d(P_i, Q) \quad (4)$$

と定義することができる。 $d(P_i, Q)$ は点 P_i と Q とのユークリッド距離を表す。(3) と (4) の構造が同じであるので、最小包含円から通信路容量を解くことができる。

2 最小包含円問題

n 次元ユークリッド空間 \mathbb{R}^n 内の有限個の点 P_1, \dots, P_m を考える. これらの点を内部または境界上に含む球のうち半径が最小のものを最小包含円といい, これを $\Gamma(P_1, \dots, P_m)$ と表す. 最小包含円 $\Gamma(P_1, \dots, P_m)$ は (4) によって定式化される. (4) を達成する $Q = Q^*$ が $\Gamma(P_1, \dots, P_m)$ の中心であり, $d^* = \max_{1 \leq i \leq m} d(P_i, Q^*)$ がその半径である.

2.1 P_1, \dots, P_m からの等距離点

P_1, \dots, P_m に対して, Q^0 が P_1, \dots, P_m からの等距離点とは, $d(P_1, Q^0) = \dots = d(P_m, Q^0)$ を満たすことである. [2] でも述べている通り, この等距離点を求めることが最小包含円を求める基本となっている.

2.2 重心座標

等距離点の位置を表すのに重心座標を利用する. \mathbb{R}^n において一般の位置にある m 個の点 P_1, \dots, P_m に関する重心座標とは

$$\begin{cases} \vec{OQ} = \lambda_1 \vec{OP_1} + \dots + \lambda_m \vec{OP_m} \\ \sum_{i=1}^m \lambda_i = 1 \end{cases} \quad (5)$$

を満たす実数の組 $\lambda = (\lambda_1, \dots, \lambda_m)$ のことである. これ以降, 簡単のため

$$\begin{cases} \vec{OQ} = \lambda_1 P_1 + \dots + \lambda_m P_m \\ \sum_{i=1}^m \lambda_i = 1 \end{cases} \quad (6)$$

と表す. (6) 式を用いることで, 位置を把握することができる. この重心座標が最小となる点を除外することで, 最小包含円を構成する上で必要となる点を求めていく [2].

2.3 高次元化による重心座標

最小包含円を求めるには, 次元を上げてから重心座標を用いて, 最小包含円を構成する上で必要としない点を除外していく [2]. 通信路容量を求めるには, この方法を置き換える必要がある. 置き換える前に基礎的部分について説明する.

点 $P_i = (P_{i1}, \dots, P_{in}) \in \mathbb{R}^n, i = 1, \dots, m$ とおき, 等距離点 $Q^0 = (Q_1^0, \dots, Q_n^0) \in \mathbb{R}^n$ とおく. また,

$$\Phi = \begin{pmatrix} P_1 \\ \vdots \\ P_m \end{pmatrix} = \begin{pmatrix} P_{11} & \dots & P_{1n} \\ \vdots & & \vdots \\ P_{m1} & \dots & P_{mn} \end{pmatrix} \in \mathbb{R}^{m \times n} \quad (7)$$

とおく. m 個の点 P_1, \dots, P_m に対して次元を上げた点を $\tilde{P}_1, \dots, \tilde{P}_m$ とおき,

$$\begin{aligned} P_1(P_{11}, \dots, P_{1n}) &\rightarrow \tilde{P}_1(P_{11}, \dots, P_{1n}, \epsilon, 0, \dots, 0) \\ P_1(P_{21}, \dots, P_{2n}) &\rightarrow \tilde{P}_1(P_{21}, \dots, P_{2n}, 0, \epsilon, \dots, 0) \\ &\vdots \\ P_m(P_{m1}, \dots, P_{mn}) &\rightarrow \tilde{P}_m(P_{m1}, \dots, P_{mn}, 0, 0, \dots, \epsilon) \end{aligned} \quad (8)$$

と定義する. $\epsilon \in \mathbb{R}, \epsilon \neq 0$ で, $|\epsilon|$ は十分小さいとする. さらに,

$$\tilde{\Phi} = \begin{pmatrix} \tilde{P}_1 \\ \vdots \\ \tilde{P}_m \end{pmatrix} \in \mathbb{R}^{m \times (n+m)} \quad (9)$$

とおく. 次元を上げて, (8) 式のように各次元ごとに微小 ϵ 加えたことにより, 等距離点 \tilde{Q}^0 が存在することになる [2]. 等距離点 \tilde{Q}^0 を求めたあと, この等距離点に対する重心座標を求めることで, 除外する点が判明する. 通信路容量を求めるためには, 次元の上げ方を変更させなければならない. そのことについて, 次節で説明する.

3 通信路容量の解法

3.1 通信路行列の次元を上げる方法

通信路容量を求める場合, 通信路行列 Φ を最小包含円の各点とみなす. そして入力確率分布 λ を重心座標とみなし, 同様にして次元を上げる. しかし, 次元の上げ方が最小包含円を求める場合とは違う. 次元を n , 点の個数を m と置いた場合, 通信路行列の列が次元 n , 行が m に相当する. 通信路行列の行ベクトルを点とみなすのである. 通信路行列の行ベクトル P_1, \dots, P_m に対してそれらの次元を上げた行ベクトル $\tilde{P}_1, \dots, \tilde{P}_m$ を以下のように定義する.

$$\tilde{P}_i = \left(\frac{P_{i1}}{2m+1}, \dots, \frac{P_{in}}{2m+1}, \overbrace{\frac{1}{2m+1}, \dots, \frac{1}{2m+1}}^{2i-2 \text{ 個}}, \frac{1+\epsilon}{2m+1}, \frac{1-\epsilon}{2m+1}, \overbrace{\frac{1}{2m+1}, \dots, \frac{1}{2m+1}}^{2m-2i \text{ 個}} \right) \quad (10)$$

通信路行列の各行のベクトルの総和は 1 となるため, (10) 式となる.

$m \times 2m$ 行列 R を定義する.

$$R = \frac{1}{2m} \begin{pmatrix} 1+\epsilon & 1-\epsilon & 1 & 1 & \dots & 1 & 1 \\ 1 & 1 & 1+\epsilon & 1-\epsilon & \dots & 1 & 1 \\ \vdots & \vdots & & \ddots & \ddots & \vdots & \vdots \\ 1 & 1 & 1 & 1 & \dots & 1+\epsilon & 1-\epsilon \end{pmatrix} \quad (11)$$

R の第 i 行を $R_i, i = 1, \dots, m$ とおくと (10) は

$$\tilde{P}_i = \left(\frac{1}{2m+1} P_i, \frac{2m}{2m+1} R_i \right), i = 1, \dots, m \quad (12)$$

と書ける.

最小包含円の中心座標を Q とおくと, 次元を上げた場合

の中心座標を \tilde{Q} とする. $Q = (Q_1, \dots, Q_n) \in \Delta^n, U = (U_1, \dots, U_{2m}) \in \Delta^{2m}$ とすると $\tilde{Q} \in \Delta^{n+2m}$ は

$$\tilde{Q} = \left(\frac{1}{2m+1} Q, \frac{2m}{2m+1} U \right), i = 1, \dots, m \quad (13)$$

と定義される. $\tilde{P}_1, \dots, \tilde{P}_m$ からの等距離点, すなわち KL 情報量が等しい点 \tilde{Q} を計算する. \tilde{Q} は

$$D(\tilde{P}_i || \tilde{Q}) = D(\tilde{P}_1 || \tilde{Q}), i = 2, \dots, m \quad (14)$$

を満たす点である. 計算により

$$D(\tilde{P}_i || \tilde{Q}) = \frac{-1}{2m+1} \left\{ \sum_{j=1}^n P_{ij} \log Q_j + \epsilon \log \frac{U_{2i-1}}{2i} + H(P_i) + \sum_{j'=1}^2 m \log(2m U_{j'} + \bar{H}(\epsilon)) \right\} \quad (15)$$

となる. $H(P_i) = -\sum_{j=1}^n P_{ij} \log P_{ij}$, $\bar{H}(\epsilon) = -(1 + \epsilon) \log(1 + \epsilon) - (1 - \epsilon) \log(1 - \epsilon)$ とおき, (14),(15) より,

$$\begin{aligned} \sum_{j=1}^n (P_{ij} - P_{1j}) \log Q_j + \epsilon \log \frac{U_{2i-1}}{U_{2i}} - \epsilon \log \frac{U_1}{U_2} \\ = -H(P_i) + H(P_1), i = 2, \dots, m \end{aligned} \quad (16)$$

となる. 次元を上げた点 \tilde{P}_i に対する重心座標を $\tilde{\lambda} = (\tilde{\lambda}_1, \dots, \tilde{\lambda}_m)$ とする. これにより, (14) は (17) となる.

$$\begin{aligned} \sum_{j=1}^n (P_{ij} - P_{1j}) \log \sum_{i'}^m \tilde{\lambda}_{i'} P_{i'j} + \epsilon \log \frac{1 + \epsilon \tilde{\lambda}_i}{1 - \epsilon \tilde{\lambda}_i} - \\ \epsilon \log \frac{1 + \epsilon \tilde{\lambda}_1}{1 - \epsilon \tilde{\lambda}_1} = -H(P_i) + H(P_1), i = 2, \dots, m \end{aligned} \quad (17)$$

$\tilde{\lambda}$ は $\sum_{i=1}^m \tilde{\lambda}_i = 1$ という定義があるため, この定義と (17) によって除外すべき点が判明する.

4 射影アルゴリズム

4.1 アルゴリズムの手順

射影アルゴリズム手順を説明するために図 1 にフローチャートを示す.

射影アルゴリズムの手順はまず, P_1, \dots, P_m の各次元を上げて $\tilde{P}_1, \dots, \tilde{P}_m$ とする. ずらす量は微小 ϵ である. 次に $\tilde{P}_1, \dots, \tilde{P}_m$ からの等距離点の重心座標 $\tilde{\lambda}$ を (17) と $\sum_{i=1}^m \tilde{\lambda}_i = 1$ から求める. 求めた重心座標が全て $\tilde{\lambda}_i^0 \geq 0$ であればアルゴリズムは終了する. しかし, そうでなければ, $\tilde{\lambda}_i^0$ の中で負の値で, 最小のものに対応する点を除外する [3]. そして, 点を除外したら, 次元を一つ下げる. これを繰り返すのが射影アルゴリズムである.

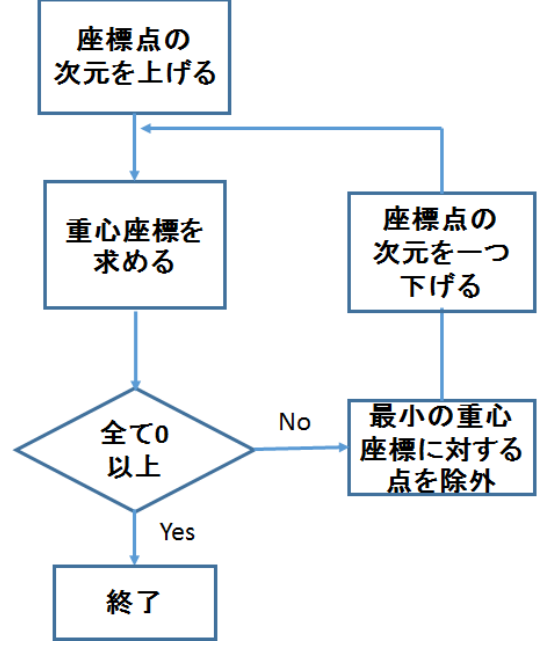


図 1: 射影アルゴリズム

4.2 妥当性の評価実験

射影アルゴリズムを用いてシミュレーションを行った. この射影アルゴリズムの妥当性の評価を行うために, m と n の値を任意に変更させ, $\sum_{i=1}^m P_i = 1$ を満たす実数を通信路行列に対してランダムに生成した. その場合に正しい答えが導出されるかどうかの確認を行った. パターン数は 1 万とし, 合計 20 万パターンもの問題を解いた. ϵ の値は 0.01 とした. その結果を表 1 に示す. 表

表 1: 射影アルゴリズムの正答率

n \ m	2	3	10	20
3	99.97%	99.95%	99.95%	99.99%
4	99.92%	99.92%	99.95%	99.96%
5	99.91%	99.89%	99.91%	99.96%
8	99.90%	99.83%	99.90%	99.96%
10	99.89%	99.58%	99.85%	99.92%

1 の結果から最小値でも 99.58% の正答率となった. n の値によっては, m が増加するごとに正答率が減少しているように見える. しかし, 微小な値なためその法則が当てはまらない可能性もある. 今回試したパターンでは 100% の正答率にならなかった. その原因を調査した結果, 本来除外されてはならない行が除外されることにより, 正しい解が得られないことが判明した.

4.3 失敗の調査

射影アルゴリズムでは100%の正答率にならなかった。失敗する問題には何かしらの規則や特徴があると考えている。考えられることとして座標点の配置に特徴があるのではないかと予想をたてた。点配置の調査を行うに当たって、 m, n になるべく小さいものの座標点を調べることにした。理由としては、 m, n の値が小さい場合はアルゴリズムの基礎部分であり、その m, n が大きい場合に適用できると考えたからである。フローチャートを確認すると、点は除外されると、 $m-1, m-2, \dots, 6, 5, 4$ 点と点が少なくなる。つまり、点の個数が少ない問題になる。だから、 m, n が小さい値が基礎であり、そこを完璧にすることで、 m, n を大きくした場合、正答率があがると考えた。

点配置に特徴がないか確認してみた。以下に失敗した一例を示す。

例題 1

$$\Phi = \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{pmatrix} = \begin{pmatrix} 0.212212 & 0.760935 & 0.0268531 \\ 0.00293794 & 0.824814 & 0.172248 \\ 0.587484 & 0.304272 & 0.108244 \\ 0.194057 & 0.00646235 & 0.799481 \\ 0.399297 & 0.402594 & 0.198109 \end{pmatrix} \quad (18)$$

この問題は3次元の問題であるが、分かりやすくするために2次元に変換し、平面状に最小包含円を描いた。プロットした図を図2に示す。

この図2で P_2 が包含されてない点である。本来ならば、 P_1, P_2, P_4 から構成される最小包含円が描かれるはずだが、 P_2 がはずれてしまった。 P_2 が外れるのには、 P_3, P_5 の点から影響を受けているのではないかと予想し、それぞれの点を外し、再度問題を解いてみた。その結果 P_5 を外した P_1, P_2, P_3, P_4 の4点であれば、正しい最小包含円が描かれることが確認できた。よって P_5 から影響を受けていることが判明した。また、 P_3 は影響を受けていないことが確認できた。そこで、 P_1, P_2, P_4 を固定し、1点をランダムに動かした場合に、どのエリアだと包含しないのか確認してみた。

実験は(19)を用いる。これは(18)から P_3 と P_5 を除いた問題である。

$$P = \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{pmatrix} = \begin{pmatrix} 0.212212 & 0.760935 & 0.0268531 \\ 0.00293794 & 0.824814 & 0.172248 \\ 0.194057 & 0.00646235 & 0.799481 \\ * & * & * \end{pmatrix} \quad (19)$$

実験方法として、3点を固定し、1点を変化させて、包含しない問題のデータを取る。*で示した行がランダムに変化させる点である。各座標を固定し、0.01刻みで変化

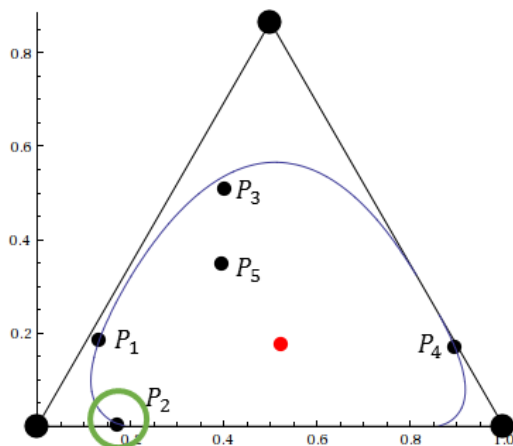


図 2: (18) のプロット

させてプロットさせた。プロットした図を、図3に示す。

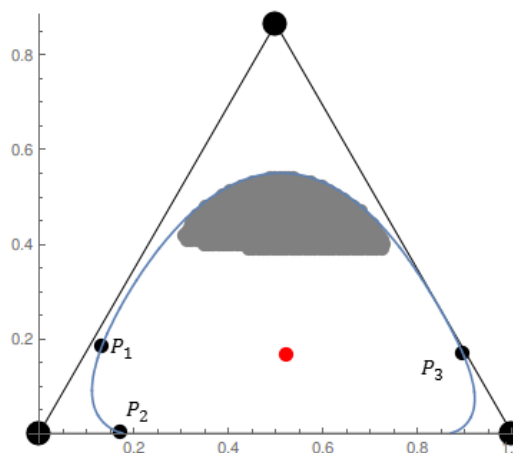


図 3: (19) のプロット

図3で示した灰色のエリアが包含しなかったエリアであり、つまり失敗したエリアである。そして、 P_1, P_2, P_3 を通る円は正しく求めた場合の最小包含円である。この最小包含円の円周が成功と失敗を分けていることが図3から分かる。失敗するエリアに対して、アルゴリズムの条件を変更して再度解いてみた。4.1のアルゴリズムの手順の条件を変更した。変更した条件は重心座標に対する点を除外する箇所である。変更前は「負のものがあれば、最小のもの、すなわち負で絶対値が最大に対応する点を除外する」であった。これを「負のものがあれば、二番目に小さいもの、すなわち負で絶対値が二番目に大きいものに対応する点を除外する」に変更した。その結果、失敗するエリアの問題は全て解くことができた。(20)の問題も失敗する問題であり、この問題に対しても同様に条件を変更して解いた。その結果、失敗するエリアの問題を解く事ができた。このことから、失敗するエリアの

区別することができれば、条件を変更することで全ての問題を解く事ができると考える。

$$P = \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{pmatrix} = \begin{pmatrix} 0.037159 & 0.585589 & 0.377252 \\ 0.755976 & 0.058562 & 0.185462 \\ 0.10507 & 0.736022 & 0.158908 \\ * & * & * \end{pmatrix} \quad (20)$$

4.4 5点の問題

点を5点に増やした場合についても調査してみた。失敗した一例を以下に示す。

$$P = \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{pmatrix} = \begin{pmatrix} 0.0279726 & 0.655278 & 0.316749 \\ 0.765067 & 0.0634631 & 0.17147 \\ 0.808356 & 0.164927 & 0.0267167 \\ 0.237277 & 0.351035 & 0.411687 \\ 0.415503 & 0.218014 & 0.366483 \end{pmatrix} \quad (21)$$

プロットした図を図4に示す。

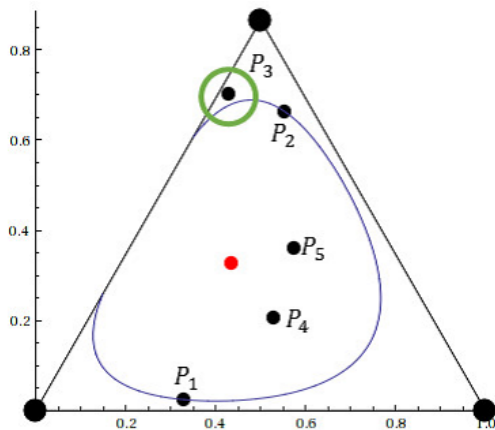


図4: (21)のプロット

図4の P_3 が包含されていない点である。本来ならば、 P_1, P_2, P_3 によって構成される最小包含円が描かれる。4点と同じように調査したところ、 P_4, P_5 がそれぞれ影響を受けていることが分かった。より詳しく調べるために、 P_1, P_2, P_3, P_4 を固定し、1点をランダムに動かした場合に、どのエリアだと失敗するのか確認してみた。実験に使った問題は P_5 を除いた(22)及び、 P_4 を除いた(23)である。

$$P = \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{pmatrix} = \begin{pmatrix} 0.0279726 & 0.655278 & 0.316749 \\ 0.765067 & 0.0634631 & 0.17147 \\ 0.808356 & 0.164927 & 0.0267167 \\ 0.237277 & 0.351035 & 0.411687 \\ * & * & * \end{pmatrix} \quad (22)$$

$$P = \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{pmatrix} = \begin{pmatrix} 0.0279726 & 0.655278 & 0.316749 \\ 0.765067 & 0.0634631 & 0.17147 \\ 0.808356 & 0.164927 & 0.0267167 \\ * & * & * \\ 0.415503 & 0.218014 & 0.366483 \end{pmatrix} \quad (23)$$

実験方法として、4点を固定し、1点を変化させて、包含しない問題のデータを取る。*で示した行がランダムに変化させる点である。各座標を固定し、0.01刻みで変化させてプロットさせた。プロットした図を、図5と図6に示す。

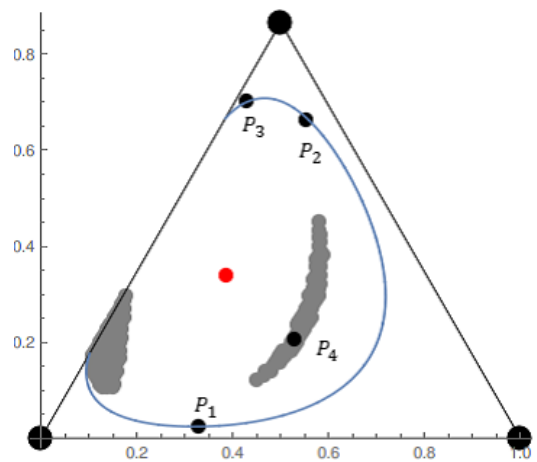


図5: (22)のプロット

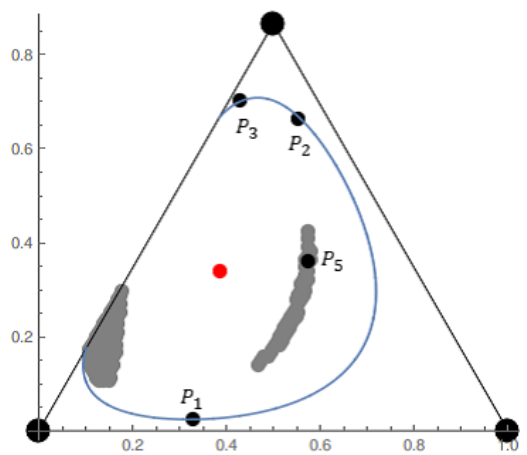


図6: (23)のプロット

図5と図6は P_4 と P_5 の位置が異なるだけで、同じ図であった。しかしながら、失敗したプロットの点の数に差がでた。(22)は159個、(23)は138個と差は19個違った。 P_4 の位置の違いにより、失敗する点の数も変化すると考える。

次に、失敗するエリアを重心座標の条件を変更して解いた。変更した条件は先ほどと同じ、「負のものがあれ

ば、二番目に小さいもの、すなわち負で絶対値が二番目に大きいものに対応する点を除外する」である。その結果、失敗するエリアの部分は解く事ができなかった。そこで、さらに条件を変更させた。フローチャートの一回目のループでは、条件を変更せずにそのままにし、二回目のループで、重心座標の除外条件を変更した。理由としては、5点問題は一回目のループで点が一つ除外される。つまり、4点の問題になる。4点の問題は、重心座標が二番目に小さいものに対応する点を除外することで失敗するエリアも解くことができた。だから、二回目のループを4点と同様に解く事で、成功すると考えた。実行した結果を図7と図8に示す。

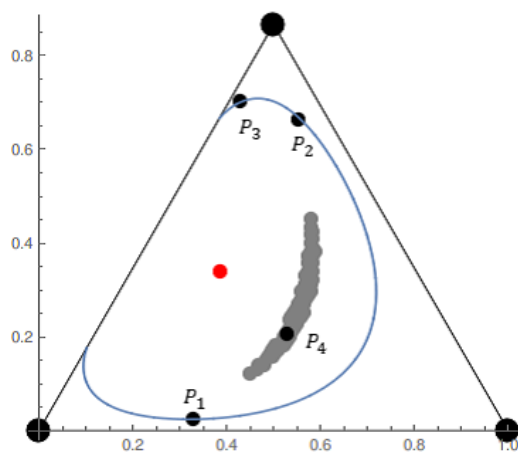


図 7: (22) のプロット

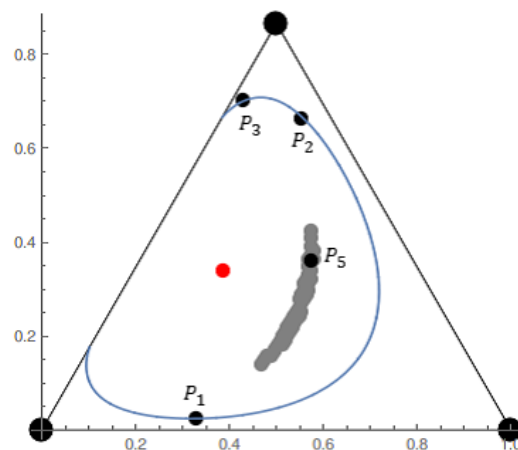


図 8: (23) のプロット

図7は(22)を、図8は(23)を再度解いた問題である。どちらも同じ図となったが、先ほどと同様に、失敗したプロットの点の数に差がでた。(22)は69個、(23)は48個と差は19個であった。数は減少したが、差は先ほどと同様であった。つまり、この差についてより詳しく調べることで、それぞれの問題の違いを明らかにできると

考える。

灰色が失敗するエリアである。この条件では、失敗するエリアがあることが判明した。そこで、失敗するエリアを包含するために、条件をさらに変更した。フローチャートの一回目のループで条件を変更し、重心座標が二番目に小さいものに対応する点を除外し、二回目のループ以降で元の条件に戻した。予想としては、図7の失敗するエリアが包含し、成功したエリアが包含失敗すると考えた。実際に実験を行った結果、失敗するエリアは全て解くことができた。予想とは違った答えとなった。(20)の問題は条件を変更することで、すべての問題を解く事ができた。また、(23)も失敗するエリアを全て解く事ができた。点の個数が増えると、ループの回数によって、条件を変更させなければならない場合がある。成功するエリアと失敗するエリアの違いを区別することができれば、条件を変更することは容易だと考える。そのために重要なのが重心座標の値だと考える。この値から条件を変更させる判別式を見つけることができれば、このアルゴリズムで全ての問題を解く事ができると考える。

5 今後の課題

本論文では、射影アルゴリズムの妥当性の評価及び、失敗する問題の点配置について述べた。そして、包含するエリアと包含しないエリアを判明することができた。それを判別する判別式を見つけることができれば、全ての問題を解くことが可能だといえる。今後は、この判別式をみつけるために重心座標に特徴がないか調査を行う。

参考文献

- [1] 中川健治, 渡部康平, 佐武拓斗, “通信路容量を達成する出力分布の射影アルゴリズムによる探索について”, 第38回情報理論とその応用シンポジウム (SITA 2015), 岡山, 2015.
- [2] 中川健治, 渡部康平, 佐武拓斗, “通信路容量を達成する出力分布の射影アルゴリズムによる探索について”, 第9回シャノン理論ワークショップ (STW 2015), 和歌山, 2015.
- [3] Kenji Nakagawa, Kohei Watabe, Takuto Sabu, “On the Algorithm for the Output Distribution that Achieves the Channel Capacity”, arXiv:1601.01394v2[cs.IT], January 2016.
- [4] Imre Csiszar, Janos Koerner, “Information Theory: Coding Theorems for Discrete Memoryless Systems”, Academic Press, Orlando, 1982.