11-NA14

量子 i.i.d. 状態のシミュレーション

坂下　達哉（電気通信大学）

概要

量子情報理論において，量子 i.i.d. 状態（n 次のテンソル積で表現される量子状態）に関する極限式は重要である．本研究では，二つの 2×2 の密度行列から成る量子 i.i.d. 状態の仮説検定問題における誤り確率を扱う．これを直接的な方法で計算しようとすると，$2^n \times 2^n$ という非常に大きなサイズの行列の固有値問題を解く必要が生じる．しかし，テンソル積の既約分解という数学的概念にもとづいた計算手法を用いると，最大サイズ $(n+1) \times (n+1)$ までのいくつかの行列の固有値問題を解くことによって同じ計算が可能になる．本稿では，この計算方法に関して本年度取り組んできた並列実装上の工夫について紹介する．

1　研究の目的と意義

量子情報理論を理解する上で，量子 i.i.d. 状態に関する極限式は重要である．これまで，古典情報理論において漸近性を扱うには数値的手法が活用されていたが，量子情報理論においては実現可能とも有効であるとも考えられていなかった．

テンソル積の既約分解という手法は林 [3] により初めて量子 i.i.d. 状態に適用され，長岡 [8] により数値計算アルゴリズムが提案された．これにより，数値計算が現実的な時間で行えるようになった．このアルゴリズムは，長岡の指導のもとで，柿崎 [12]，堂嶋 [16] により研究された．

これらの研究を受けて，我々は，既知の量子仮説検定問題を題材として，大規模で高精度の数値シミュレーションが可能であることを実証した [10, 15]．本研究の目的は，既知の問題を解析し，未解決の問題の極限式予測に活用できるようにすることである．例えば，文献 [10] では，中心極限定理に関連した新しい予想を数値的に検証している．

本年度は，このような新しい予想への適用に備えて，既知の仮説検定問題を題材として並列実装上の工夫を行うことに専念した．以下の副節では，その工夫について述べるための準備として，研究の概要を紹介し，これまで得られた成果について述べる．

1.1　量子仮説検定

本節では量子仮説検定について必要最小限の事柄を述べる．

まず，いくつかの物理的な概念に数学的な定義を与える．量子状態は，トレースが 1 の半正定値行列（密度行列と呼ばれる）ρ

で表される．測定は，半正定値行列の集合 $M = \{M(x)\}_{x\in\mathscr{X}}$ で，$\sum_{x\in\mathscr{X}} M(x) = I$ (単位行列) を満たすもので表される．これを POVM (Positive Operator-Valued Measure) と呼ぶ．ここで，添字 x は測定値を表す．量子状態 ρ のもとで測定 M を行ったとき，測定値 x が得られる確率は

$$P_\rho^M(x) := \mathrm{Tr}[\rho M(x)] \tag{1}$$

で表される．

量子 i.i.d. (independent and identically distributed) 状態は，n 次のテンソル積

$$\rho^{\otimes n} = \underbrace{\rho\otimes\cdots\otimes\rho}_{n}$$

で表される状態のことである．ここで，行列 $A : d\times d$ サイズと $B : d'\times d'$ サイズのテンソル積を $dd'\times dd'$サイズ の行列

$$A\otimes B := \begin{bmatrix} a_{11}B & \cdots & a_{1d}B \\ \vdots & \ddots & \vdots \\ a_{d1}B & \cdots & a_{dd}B \end{bmatrix} \tag{2}$$

と定義する．ただし，n はテンソル次数とよばれる．量子 i.i.d. 状態は，物理的には独立に用意された n 個の状態を意味しており，古典確率論における独立同一分布に対応する．

次に，本研究で扱う量子仮説検定問題について述べる．いま，2×2 サイズの密度行列 ρ, σ で表される量子状態が与えられているとし，実数 a に対して，次の量

$$\beta_n(a) := \mathrm{Tr}[\sigma^{\otimes n}\{\rho^{\otimes n} - e^{na}\sigma^{\otimes n} > 0\}] \tag{3}$$

を考える．ここで，任意のエルミート行列 A に対し，行列 $\{A > 0\}$ を以下のように定める．A の固有値と固有ベクトルから成る正規直交基底をそれぞれ $\{\lambda_i\}$，$\{x_i\}$（縦ベクトル）とおくと，A のスペクトル分解

$$A = \sum_i \lambda_i\, x_i x_i^* \tag{4}$$

が得られる．ここで，x_i は縦ベクトルで x_i^* は横ベクトルであるから，積 $x_i x_i^*$ は A と同じサイズの正方行列である．これに対し，

$$\{A > 0\} := \sum_{\lambda_i > 0} x_i x_i^* \tag{5}$$

と定義する．$\{A > 0\}$ は A と同じサイズのエルミート行列であり，A の正固有値に対応する固有空間の直和への射影を表す．

式 (3) は量子 Neyman-Pearson 検定 $\{\rho^{\otimes n} - e^{na}\sigma^{\otimes n} > 0\}$ の二つの誤り確率のうちの一つであり，$\sigma^{\otimes n}$ が真のときに $\rho^{\otimes n}$ を誤って真と判断する確率を意味する [8]．また，

$$r_n(a) := -\frac{1}{n}\log\beta_n(a), \tag{6}$$

$$r(a) := \lim_{n\to\infty} r_n(a) \tag{7}$$

とすると，量子 Hoeffding の定理 [4, 6] より，任意の $a > -D(\sigma\|\rho)$ に対して，

$$r(a) = \max_{\theta\in\mathbb{R}}\{\theta a - \psi(\theta)\} \tag{8}$$

ここで，$\psi(\theta) = \log\mathrm{Tr}[\rho^\theta\sigma^{1-\theta}]$

が成り立つ．式 (8) の右辺は Newton 法などを用いることで求められ，計算する行列は 2×2 サイズなので計算時間はかからない．

量子情報理論では，式 (7) のような量子 i.i.d. 状態に関する $n\to\infty$ における極限式が至るところに現れる．冒頭でも述べたが，本研究の目的は，このような極限式の予想を行うために数値シミュレーションを活用することである．そのための並列化実装を工夫する上で，既に極限式 (8) が得られている量子 Hoeffding の定理は，プログラムの試作に当たって格好の題材である．そこで，以降では，量子 Hoeffding の定理に的を絞って解説を行う．

1.2 既約成分の計算アルゴリズム

量子 i.i.d. 状態の密度行列 $\rho^{\otimes n}$ のサイズは $2^n \times 2^n$ になるため，この行列についての計算は大変である．そこで，本節ではテンソル積の既約分解という方法を導入する．

任意の 2×2 行列

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \tag{9}$$

に対し適当な基底変換を施すことで，$A^{\otimes n}$ は

$$\bigoplus_{k=0}^{\lfloor n/2 \rfloor} \bigoplus^{m_k} A_k \tag{10}$$

$$= \begin{bmatrix} A_0 & & & & O \\ & A_1 & & & \\ & & A_1 & & \\ & & & \ddots & \\ O & & & & A_{\lfloor \frac{n}{2} \rfloor} \end{bmatrix} \tag{11}$$

という形のブロック対角行列で表される [2]．各 A_k はテンソル積 $A^{\otimes n}$ の既約成分とよばれ，そのサイズは $\dim A_k = n + 1 - 2k$ で，重複度 $m_k := {}_nC_k - {}_nC_{k-1}$ 回だけ現れる．(ここで，便宜的に ${}_nC_{-1} := 0$ とする．) つまり，式 (11) はまず A_0 を $m_0 = 1$ 個だけ対角に並べ，次に A_1 を m_1 個だけ対角に並べるという操作を繰り返して得られる $\sum_k ({}_nC_k - {}_nC_{k-1})(n+1-2k) = 2^n$ サイズの正方行列である．ここで，便宜的に ${}_nC_{-1} := 0$ としている．

$r := n - 2k$ とおくとき，A_k の (i, j) 成分 $\alpha_{k,ij}$ $(i, j \in \{0, 1, \ldots, r\})$ は，まず

$$(a_{11}y + a_{21}z)^{r-j}(a_{12}y + a_{22}z)^j = \sum_{i=0}^{r} \alpha'_{k,ij} y^{r-i} z^i \tag{12}$$

という多項式展開から係数 $\alpha'_{k,ij}$ を求め，これを用いて

$$\alpha_{k,ij} = (\det A)^k \sqrt{\frac{{}_rC_j}{{}_rC_i}} \alpha'_{k,ij} \tag{13}$$

と計算できる．

ここで, 肝要なことは式 (11) の既約分解に用いる基底変換が A の行列成分によらないことである．よって，テンソル積の加算・減算は既約成分ごとに行える．また，後でみるようにトレースをとるので，この基底変換は存在することは重要であるが具体的に求める必要はない．

このように，テンソル積の既約分解を用いることで，指数サイズであるテンソル積の行列成分の計算が多項式オーダーで行える．

1.3 既約分解を用いた誤り確率の計算

本節では前節で述べたテンソル積の既約分解を誤り確率の計算に適用する．

式 (11) の既約分解を用いることによって，誤り確率 $\beta_n(a)$ は次のように表される:

$$\beta_n(a) = \mathrm{Tr}[\sigma^{\otimes n}\{\rho^{\otimes n} - e^{na}\sigma^{\otimes n} > 0\}] \tag{14}$$

$$= \sum_{k=0}^{\lfloor n/2 \rfloor} m_k \beta_{n,k}(a) \tag{15}$$

ここで,

$$\beta_{n,k}(a) := \mathrm{Tr}[\sigma_k\{\rho_k - e^{na}\sigma_k > 0\}] \tag{16}$$

であり，式 (16) の $\{\rho_k\}$ と $\{\sigma_k\}$ はそれぞれ $\rho^{\otimes n}$ と $\sigma^{\otimes n}$ の既約成分である．

図 1 で式 (16),(15) の計算手順を与える．

（ステップ） *1.* 各 k について，以下を行う．

(*a*) 式 (12),(13) を用いて $\{\rho_k\}$ と $\{\sigma_k\}$ を計算する．

(*b*) 行列 $\rho_k - e^{na}\sigma_k$ の固有値 $\{\lambda_{n,k,i}\}_i$ と正規直交化された固有ベクトル $\{v_{n,k,i}\}_i$ を求める．

(*c*) $\beta_{n,k}(a) = \sum_{\lambda_{n,k,i}>0} v^*_{n,k,i}\sigma_k v_{n,k,i}$ を求める．

（ステップ） *2.* $\beta_n(a) = \sum_{k=0}^{\lfloor n/2 \rfloor} m_k \beta_{n,k}(a)$ を求める．

図 1　$\beta_n(a)$ の計算手順

ここで，図 1 の手順のステップ *1* において，各 k に対する処理は独立なので並列化を施している．また，複数の a について $\beta_n(a)$ を求める場合，図 1 のステップ *1.*(*a*) において既約成分 $\{\rho_k\}, \{\sigma_k\}$ は全ての a について共通なので，一度だけ計算すればよいことに留意する．以上の二点についての詳細は 1.5 節で述べる．

2×2 サイズの 2 つの密度行列については，実対称行列に限って良いことが知られているので，実際の計算では ρ, σ は実対称行列のみを用いる．固有値分解にも実対称行列向けのルーチンを用いる．固有値ルーチンを選定する際には，以下に事項に注意を要する．

1. 全固有値, 全固有ベクトルが必要である．
2. 行列 $\rho^{\otimes n}, \sigma^{\otimes n}$ は指数的にばらついた固有値を持つ．したがって，実対称行列 $\rho_k - e^{na}\sigma_k$ には絶対値の小さい正負の固有値があるが，固有値の正負が数値誤差によって変わることがないように正確に求める必要がある．

このような場合，固有値解法には Householder QR 法が適していることがわかっているため [14]，これを用いる．

1.4　計算機環境

使用した計算機環境は以下の通りである．

- 計算機：スーパーコンピュータ 東京大学 HA8000 (AMD Opteron processor 8356)
- コンパイラ：Intel C++ compiler
- C++ テンプレート線形代数ライブラリ：Eigen3[1]
- 多倍長演算ライブラリ：Exflib[11]
- 固有値分解に用いるライブラリ：LAPACK dsyev 関数 (倍精度・QR 法の場合)，Eigen3 の固有値分解関数を改変 (多倍長・QR 法の場合)

工夫した点としては，Eigen3 に多倍長型を追加し，多倍長型を成分に持つ行列を扱えるようにしたことがあげられる．これにより，倍精度に限らず多倍長も MATLAB などと同様に行列演算を記述できる．また，倍精度・多倍長の切り替えも，行列クラス名の置換のみで済む．なお，現在，上述の Eigen3 の固有値分解関数の性能は不十分なため，自前で改良し並列化を施すことも本研究の課題である．

1.5 これまでの並列化方法

本節では，昨年度までの並列化方法について記述する．次節で，この実装の問題点と今回新しく考案した並列化法を述べる．

図 1 の計算手順において，ステップ *1* の異なる k についての処理は互いに独立であるため，自明な並列化が施せることを述べた．

並列化方法としては，各プロセス（スレッド）に複数の k についての処理を担当させる必要がある．その際，各既約成分の格納には膨大なメモリが必要で，データの送受信は各プロセス（スレッド）で計算した $m_k \beta_{n,k}(a)$ と固有値分解の計算時間等の付加的な情報だけである．このような処理には OpenMP よりも MPI が向いているため，本研究では MPI を用いることにする．

各プロセスのメモリ使用量・計算量をできるだけ均一にするため，担当する既約成分は巡回的に割り当て，この割り当てはすべての a に対して固定しておく．また，既約成分 $\{\rho_k\}, \{\sigma_k\}$ は一度だけ計算してメモリに格納しておき使い回すことにする．以下で，一つの a に対して誤り確率 $\beta_n(a)$ の具体的な計算手順を述べる．

まず，各プロセスで既約成分を計算し，多倍長型（または倍精度型）の行列に収める．その際，既約成分のサイズは各々異なるので，行列クラスからなる配列を用いて管理することはできない．そこで，「行列クラスへのポインタ」からなる配列を用いて既約成分を管理する．

次に，各プロセスで $m_k \beta_{n,k}(a)$ を計算し，あらかじめ決めておいた一つのプロセス（ルートプロセスと呼ぶ）に集約した後，ルートプロセスでステップ *2* の総和 $\beta_n(a) = \sum_k m_k \beta_{n,k}(a)$ を計算する．この集約を行うために，`MPI_Gather` 関数を用いることにする．ここで，計算結果の集約と総和の両方を行える `MPI_Reduce` 関数を用いないのは，$m_k \beta_{n,k}(a)$ も考察の対象にするので，これもデータとして出力したいからである．

最後に，ルートプロセスにおいて，誤り確率 $\beta_n(a)$ を用いて誤り指数 $r_n(a) = -\log \beta_n(a)/n$ を計算する．

例として，テンソル次数 $n = 100$ の場合をとりあげる．このとき，既約成分の番号 k は $0, 1, \cdots, \lfloor n/2 \rfloor = 50$ で，対応する既約成分のサイズ $\dim A_k$ は $101, 99, \ldots, 3, 1$ となる．プロセス数が 4 の場合について，$\rho^{\otimes n}$ と $\sigma^{\otimes n}$ の既約成分の格納方式と計算結果の集約方法を図 2，図 3 に示す．ここで，ルートプロセスはプロセス 0 としている．

2 当拠点公募型共同研究として実施した意義

(1) 共同研究を実施した大学名：東京大学

(2) 共同研究分野：超大規模数値計算系応用分野

(3) 当公募型共同研究ならではという事項など：

数値計算・並列化の専門家との協業が行えたことは有意義であった．特に，固有値解法における三重対角化の並列化方法についてアドバイスをいただくことができた．また，スーパーコンピュータ HA8000 は Intel 互換の CPU を採用しているので，手元のパソコンで試したプログラムをそのまま使用できるという利点がある．特に，C++ のテンプレートライブラリ Eigen3 は対応していない CPU もあるが，HA8000 では

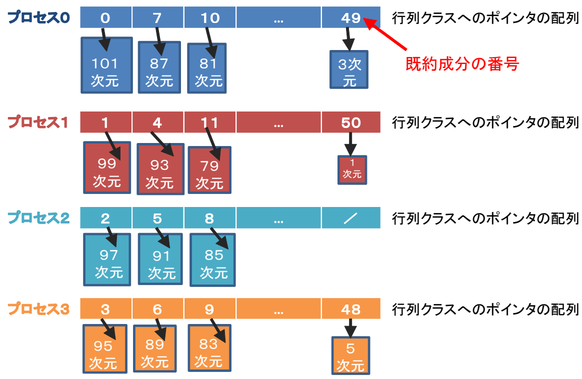

図 2　$\rho^{\otimes n}$ と $\sigma^{\otimes n}$ の既約成分の格納方式

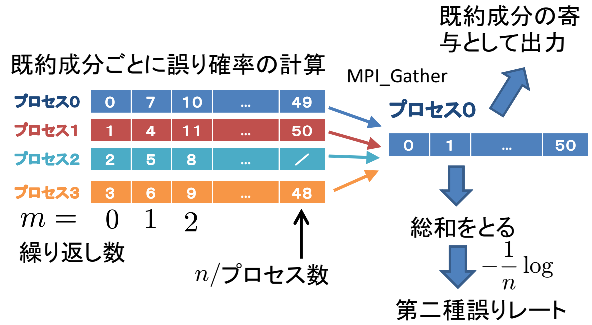

図 3　今までの計算結果の集約方法

問題なく使用できた．

3　研究結果の詳細と当初計画の達成状況

3.1　研究結果の詳細について

本節では，平成 23 年度のプロジェクトにおいて取り組んだ以下の課題について記述する．

(課題 1) 並列化の実装方法についての工夫（中間報告までに実施した課題)

(課題 2) 固有値計算を行う際の三重対角化の OpenMP によるスレッド並列化（最終報告までに実施した課題)

課題 1 は以下に述べる「支配的な既約成分」という知見に基づく．式 (15) により誤り確率 $\beta_n(a)$ は各既約成分 k についての計算値 $m_k\beta_{n,k}(a)$ の総和として求められる．我々は，その総和において実質上，ごく一部の項のみが寄与することを見出した [14]．これらの項に関する既約成分を支配的な既約成分と呼ぶ（図 4 参照)．

支配的な既約成分を考慮すると，以下の点に留意すれば効率の良い計算が行える．

- 支配的な既約成分を精度よく求める．
- その他の既約成分で大きな数値誤差が起こらないようにする．

以上を実現するために，次のように実装および MPI による並列化を変更した．

1. 支配的な既約成分だけを重点的に計算し，不要な既約成分の計算を省略する．そのため，小さな n についてまず計算することで，支配的な既約成分を予想する．
2. これまでは，全ての既約成分についての計算が終わってから集団通信を用いて一箇所に集める方式をとっていたため，プロセス間の待ち時間が生じていた．そこで，非同期通信を用いることにより，計算と通信のオーバーラップを図る．
3. 大きな既約成分ほど数値誤差が大きいことがわかっている．そのため，小さな既約成分から計算し大きな既約成分で数値誤差を検出したら計算を打ち切る（図 5 参照)．

以上について実装は出来上がっているが，現段階において上の実装を施す意義は必ずしも明確ではない．そのため，この実装が生かせる問題設定を提案して，その問題に適用した結果とともに対外発表に望みたい．

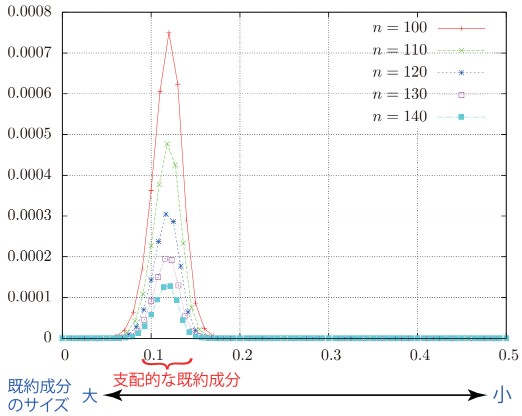

図 4　n を変えたときの既約成分ごとの計算値 ($a = -1.5$)

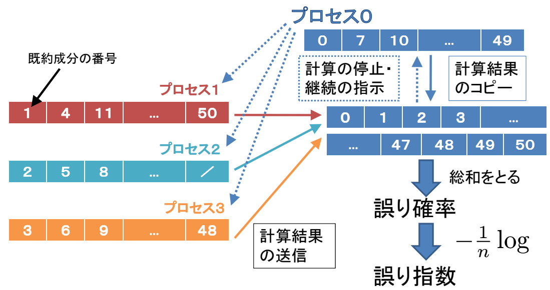

図 5　新しい並列化方法

課題 2 を行う動機について述べる．現時点では n が小さいとき，HA8000 で使用可能な $64 \times 16 = 1024$ コアが生かしきれていない．そこで，多倍長の Householder QR 法を OpenMP により並列化を行う．

Householder QR 法は，Householder 法による三重対角化と三重対角行列の固有値・固有ベクトルを求める QR 法からなる．本研究では，Householder 三重対角化の OpenMP による並列化に取り組んだ．

まず，三重対角化のうち，rank2update の for 文を OpenMP により並列化することから始めた．テンプレートライブラリ Eigen3 のソースコードを解読し三重対角化に必要な部分のみを取り出した．Eigen3 はどのような備え付けの型（float 型，倍精度等）でも受け付けるようになっているが，簡単のためこれを倍精度に書き直した．倍精度の場合には，OpenMP による並列化は成功した．だが，多倍長の場合について OpenMP による並列化を行うと，2 スレッド以上で明らかに誤った計算結果となる．この原因について，現在調査を行っている．

3.2　当初計画の達成状況について

ライブラリの解読と書き換えに多大な時間を要し，課題 2 の実装の完成に至らなかった．課題 2 の実装で起こっている問題を解決するために，テンプレートライブラリの内部構造についてさらに理解を深め，ライブラリの作成者とも連絡をとり解決策を考えたい．

4　今後の展望

5　研究成果リスト

研究成果のリストは以下の通りである．

(1) 学術論文　なし
(2) 国際会議プロシーディングス　なし（発表予定）
(3) 国際会議発表　なし
(4) 国内会議発表　なし
(5) その他（特許，プレス発表，著書等）　なし

参考文献

[1] 線形代数計算用 C++ テンプレートライブラリ Eigen2. http://eigen.tuxfamily.org/index.php?title=Main_Page.

[2] R. Goodman and N. R. Wallach. *Representations and Invariants of the Classical Groups*. Cambridge University Press, 1998.

[3] M. Hayashi. Asymptotics of quantum relative entropy from a representation theoretical viewpoint. *J. Phys. A: Math. Gen.*, 34(16):3413–3419, 2001.

[4] M. Hayashi. Error exponent in asymmetric quantum hypothesis testing and its application to classical-quantum channel coding. *Phys. Rev. A*, 76(6):062301–1,062301–4, 2007.

[5] K. M. R. Audenaert et al. Discriminating states: The quantum Chernoff bound. *Phys. Rev. Lett.*, 98(16):160501–1,160501–4, 2007.

[6] H. Nagaoka. The converse part of the theorem for quantum Hoeffding bound. *arXiv:quant-ph/0611289v1*, 2006.

[7] H. Nagaoka. Some mathematical problems related to quantum hypothesis testing. In *General Theory of Information Transfer and Combinatorics*, Vol. 4123 of *Lecture Notes in Computer Science*, pp. 1100–1103. Springer Berlin / Heidelberg, 2006.

[8] H. Nagaoka and M. Hayashi. An information-spectrum approach to classical and quantum hypothesis testing for simple hypotheses. *IEEE Trans. Inf. Theory*, 53(2):534–549, 2007.

[9] M. Nussbaum and A. Szkola. The Chernoff lower bound for symmetric quantum hypothesis testing. *Ann. Statist.*, 37(2), 2009.

[10] T. Sakashita and H. Nagaoka. A numerical study of hypothesis testing for quantum i.i.d. states. *Asian conference on quantum information science(AQIS)*, pp. 213–214, Aug. 2010.

[11] 藤原. 並列化対応多倍長ライブラリ Exflib. http://www-an.acs.i.kyoto-u.ac.jp/~fujiwara/exflib/.

[12] 柿崎. 量子的 i.i.d. 情報源に関する数値計算によるアプローチ. Master's thesis, 電気通信大学情報システム学研究科, 2006.

[13] 坂下. 量子 i.i.d. 状態のシミュレーションとその理論的考察. スーパーコンピューティングニュース, Vol.13 特集号 1, Mar. 2011.

[14] 坂下, 片桐, 長岡. 量子 i.i.d. 状態の仮説検定に関する数値的手法とその誤差分析. 情報処理学会論文誌コンピューティングシステム (ACS), (35), Oct. 2011.

[15] 坂下, 長岡. 量子 i.i.d. 状態のシミュレーションとその理論的考察. 日本応用数理学会 2010 年度年会 講演予稿集, pp. 245–246, 2010.

[16] 堂嶋, 片桐, 長岡. 量子 i.i.d. 状態における仮説検定の漸近特性に関する数値的アプローチ. 第 29 回情報理論とその応用シンポジウム予稿集 (SITA), pp. 767–770, 2006.