Roman GIELERAK, Marek SAWERWAIN
UNIWERSYTET ZIELONOGÓRSKI, INSTYTUT STEROWANIA I SYSTEMÓW INFORMATYCZNYCH

Sorting of amount of Entanglement in Quantum States functions implemented for Quantum Computing Simulator

Prof. dr hab. Roman GIELERAK

Wszystkie stopnie i tytuły naukowe uzyskał na Uniwersytecie Wrocławskim w okresie od 1976 (mgr) do 1999 (profesura). Brał udział w licznych projektach badawczych we współpracy z wieloma ośrodkami zagranicznymi (ETH Zurych; ZIBJ Dubna; MGU Moskwa; Bochum-Bonn-Bielefeld Uniwersytety; Instytut Galileusza, Paryż; Uniwersytet w Lisbonie). Autor licznych prac i opracowań monograficznych z podstaw teorii kwantowej i jej zastosowań.

e-mail: R.Gielerak@issi.uz.zgora.pl

Mgr inż. Marek SAWERWAIN

Ukończył studia na Wydziale Elektrotechniki, Informatyki i Telekomunikacji Uniwersytetu Zielonogórskiego. Aktualnie pracuje jako asystent w Instytucie Sterowania i Systemów Informatycznych. Zajmuje się kwantowymi językami programowania, oraz operacyjną semantyką dla procesów kwantowych. Pracuje również nad symulatorem kwantowego modelu obliczeniowego dla systemów jedno oraz wielo-procesorowych oraz wersji sieciowej.

e-mail: M.Sawerwain@issi.uz.zgora.pl

Abstract

A sorting of quantum states routines with respect to the amount of entanglement included is being introduced on the Quantum Computer Simulator and intensively tested. In particular a sorting routine basing on the lexicographic ordering of the corresponding Schmidt coefficients is being formulated and tested. The corresponding algorithm relies on the partial order relations and the final nonlinear sorting is given in terms of MergeSort data.

Keywords: sorting of entangled quantum states, quantum computing simulator.

Funkcje sortowania stanów kwantowych ze względu na poziom splątania zaimplementowane w symulatorze obliczeń kwantowych

Streszczenie

W artykule przedstawiono algorytmy sortowania stanów kwantowych ze względu na poziom splątania. Algorytmy te zostały zaimplementowane w ramach budowanego symulatora obliczeń kwantowych. Przedstawione zostały dwa algorytmy, pierwszy oparty jest o porządek leksykograficzny względem współczynników Schmidta oraz drugi względem porządku liniowego uzyskanego dzięki entropii von Neumanna. Zaprezentowane zostały także wyniki dotyczące złożoności obliczeniowych przedstawionych algorytmów.

Słowa kluczowe: sortowanie splątanych stanów kwantowych, symulator obliczeń kwantowych.

1. Introduction

One the most important feature of quantum information processing is the phenomena of entanglement of the corresponding quantum states. The quantum teleportation protocols, application to quantum cryptographic systems are the well known examples of this. It is known that in general answering the question whether a given quantum state ρ is entangled or not is NP-hard problem [13].

However, the following problem frequently arises. As an example let us recall that in teleportation protocols the efficiency of teleportation depends on the amount of entanglement in the states used. Let

$$\Sigma = \{\rho_1, ..., \rho_n\}. \tag{1}$$

be a given list of quantum states on a Hilbert space

$$h = H_A \otimes H_B. \tag{2}$$

of states of composite systems A and B. The problem is to sort the set $\sum$ with respect to entanglement amount contained in states ρ_i.

Depending on the notion of what and how the quantification measure for calculating entanglement is defined linear and semi-linear sorting algorithms are being formulated, then adopted as a suitable sorting routine on a software being under intensive developments at University of Zielona Góra and called Quantum Computing Simulator [7, 8, 9]. The adapted routines are then finally intensively tested with the intention of checking the computational complexity of the algorithms introduced.

2. Sorting based on the linear order caused by von Neumann entropy

It is longstanding problem to construct a computable function on quantum states quantifying the amount of entanglement contained, separating the separable states from entangled one and obeying some additional natural requirements, monotonicity with respect to LOCC operations including [10, 11, 12]. Although some functions obeying all these demands as above have been presented in the literature [10, 11, 12] the real problem with them is that they are hardly to be calculable in an efficient way.

Therefore instead of entanglement quantifying functions as above we will be satisfied himself by a functions which are monotone under LOCC operations called entanglement monotone functions. In the case of pure states there exists essentially an unique function of this sort (modulo some technical assumptions [1, and ref. therein]) and is connected to the von Neumann entropy of the corresponding reduced density matrices obtained by partial tracing operations.

$$E(\rho) = -\sum_k \lambda_k \log \lambda_k \,. \tag{3}$$

The algorithm of linear sorting of a given set of entangled states on $H_A \otimes H_B$ is depicted in Fig. 1.

3. Sorting based on the partial order caused by LOCC's

Let

$$\rho_1, \rho_2 \in E(H_1 \otimes H_2). \tag{4}$$

be two density matrices. The operational definition of comparing the amount of entanglement contained in the states ρ_1, resp. ρ_2 is the following. The state ρ_1 is more entangled then the state ρ_2 (symbolically):

$$\rho_1 \overset{LOCC}{\succ} \rho_2 \tag{5}$$

iff there exists a local operations complemented by classical communications and such that after applying them to the state ρ_1 we can produce the state ρ_2. In other words the local operations complemented with the information exchange in a classical

channel can only decrease the amount of the existing entanglement. Otherwise the quantum states ρ_1 and ρ_2 are not comparable in the sense as above.

LIN-ENTROPY-Sorting Algorithm

Input:

a set Σ of quantum pure states on $H_A \otimes H_B$

$$\Sigma = \{|\psi_1\rangle,...,|\psi_n\rangle\}$$

Output:

a set Σ^{sort} sorted with respect to the amount of entanglement given by equation (3)

$$\Sigma^{sort} = \{|\psi_{i_1}\rangle,...,|\psi_{i_n}\rangle\}$$

and

$$E\left(\rho^A_{|\psi_{i1}\rangle}\right) \le ... \le E\left(\rho^A_{|\psi_{iN}\rangle}\right)$$

Step 1:

// compute the reduced density matrices by taking partial traces

for i=1:N,

$$\rho^A_{|\psi_i\rangle} = Tr_{H_A}|\psi_i\rangle\langle\psi_i|$$

Step 2:

// compute the spectra of the reduced density matrices

for i=1:N,

$$[\sigma_i, V_i] = EigenSystem(\rho^A_{|\psi_i\rangle})$$

Step 3:

// compute the von Neumann's entropies

for i=1:N,

$$En(i) = E(\rho^A_{|\psi_i\rangle}) = -\sum_k \lambda_k^i \log \lambda_k^i$$

Step 4:

// sort by classical sort-algorithm the set of values from step 3

CLASSICAL-SORT: $\{En(1),..., En(n)\}$

Step 5:

// display the sorted list of states

$$\Sigma^{sort} = \{|\psi_{i_1}\rangle,...,|\psi_{i_n}\rangle\}.$$

Rys. 1. Pseudokod algorytmu liniowego sortowania stanów kwantowych przy wykorzystaniu entropii von Neumann'a

Fig. 1. Pseudo-code for linear sorting algorithm of quantum states using von Neumann's entropy

It is easy to see that the relation (5) defines a partial order on the space of quantum states in the sense that this relation is transitive and irreflexive.

In order to proceed further we have to explain what we mean by sorting the space equipped with partial order instead of the well known case of linear order which is the classical problem of informatics and is very well known explored topic [6].

Let $P = (X,<)$ be finite partially ordered space (poset). A chain $C \subseteq P$ is a subset of mutually comparable elements. The height of an element x is the maximum cardinality of a chain whose elements are all dominated by x. The set of elements of height 0 is called the set of minimal elements. An anti-chain A in P is a subset of mutually incomparable elements. The width $w(P)$ of poset P is defined to be the maximum cardinality of an anti-chain of P.

A decomposition CC of a poset P into chains is a family $CC=[C_1,C_2,\ldots,C_p]$ of disjoint chains such that their union is equal to P. The size of a decomposition is the number of chains in it. A decomposition of size $w(p)$ is called a minimum chain decomposition. Dilworth's theorem guarantees the existence of chain decomposition of P of size $w(P)$.

The sorting problem for a given poset P is to completely determine the partial order $<$ in X. In the absence of a bound on the width the query complexity of such sorting problem is exactly

$$T(n) = \binom{n}{2}, \qquad (6)$$

where n=card(P). In the case of sorting problem for a linearly ordered set P we have w=1.

In the paper [5] two algorithms for sorting posets have been described, both of which have query complexity $O(\ w*n \log n/w)$. The total complexity of sorting posets first algorithm as formulated in [5] depends on the subroutine for computing a chain decomposition (the complexity of which is not analyzed in [5].

The first algorithm of [5] is in some sense the generalization (of the well known in the linear case) of MergeSort classical algorithms and it is this algorithms which we adopt to our problem at hands. Adaptations of an another algorithms are under preparations and will be presented elsewhere.

The query oracle that will be used in our quantum algorithm is based on the following theorem by Nielsen [1, 2].

Theorem 1 [Nielsen]

Let $\rho_1 = |\psi_1\rangle\langle\psi_1|$, $\rho_2 = |\psi_2\rangle\langle\psi_2|$ *be a pair of pure states of the same Schmidt rank k. Than the state* $|\psi_1\rangle$ *is more entangled in LOCC sense iff for any i=1:k*

$$\lambda_1^2(1) + \ldots \lambda_i^2(1) \le \lambda_1^2(2) + \ldots \lambda_i^2(2)$$

where $\{\lambda_1(\alpha),\ldots,\lambda_k(\alpha)\} = \Lambda(\alpha)$ *is the set of the Schmidt's coefficients of the vector* $|\psi_\alpha\rangle, \alpha = 1,2$ *ordered in descending order. If ,say ,the Schmidt rank of* $|\psi_1\rangle$ *is bigger than this of* $|\psi_2\rangle$ *then the state* $|\psi_1\rangle$ *is more entangled then* $|\psi_2\rangle$.

The query oracle used in the following is:

2N-Query Oracle Algorithm

Input: two vector states on $H = H_1 \otimes H_2$, $V = \{\rho_1, \rho_2\}$

Output: $(\rho_1 \prec \rho_2)\, or\, (\rho_2 \prec \rho_1)\, or\, non-comparable$

Step 1: for i=1:2

{r_i, $\{\lambda_1(i),\ldots,\lambda_{r_i}(i)\}$}=SchmidtDecomp($V_i$)

sort the set $\{\lambda_1(i),\ldots,\lambda_{r_i}(i)\}$ in descending order

Step2:

if $r_1 > r_2$ return $(\rho_1 \prec \rho_2)$

if $r_2 > r_1$ return $(\rho_2 \prec \rho_1)$

Step3:

if $r_1 = r_2$

if $\underset{j=1:r_1}{\forall}\left(\sum_{i=1}^{j}\lambda_i^2(1) \le \sum_{i=1}^{j}\lambda_i^2(2)\right)$ then return $(\rho_1 \prec \rho_2)$

if $\underset{j=1:r_1}{\forall}\left(\sum_{i=1}^{j}\lambda_i^2(2) \le \sum_{i=1}^{j}\lambda_i^2(1)\right)$ then return $(\rho_2 \prec \rho_1)$

return non-comparable

As the final output of our algorithm will be presented in the form of CHAINMERGE –like data structure we describe this point now.

Let $CC=[C_1,\ldots,C_P]$ be a chain decomposition of a poset $P=(X,<)$. CHAINMERGE (P,CC) stores, for each element $x \in P$, a p indices as follows: Let C_i be the chain containing the element x. The data structure stores the internal index of $x \in C_i$, and, for all $1 \le j \le p, j \ne i$, the index of the largest element of the chain C_j that is dominated by x. The performance of this type of the data structure is very good as given CHAINMERGE(P, CC) data, the relation $<$ in P of any pair of elements can be found in constant time.

Now the corresponding algorithm runs as follows

LOCC-V-MERGESORT Algorithm

Input:

$V = \left[|\psi_i\rangle \right], i = 1,\ldots,n$

// a list of vector states on the space $H = H_A \otimes H_B$

Output (1):

The partitioning of V:

V=[V_1,…,V_p] where

$V_{(i)} \in \mathbf{V}$, SchmidtRank(V_i) = r_i = const; $r_1 < r_2 < \ldots < r_p$, and $U_i V(i) = \mathbf{V}$

// partitioning with respect to increasing Schmidt's rank

Output(2):

for i=1:r

ChainMerge(V_i)

// data

Step 1:

for i=1:N

S(i)=SchmidtDecomposition(V(i))

// S(i)=[r(i), $\left(\lambda_1(i),\ldots,\lambda_{r_i}(i)\right)$]

// Schmidt decomposition data, in particular r(i)=SchmidtRank of $|\psi_i\rangle$

Step 2:

r*=max(r(1), …,r(N))

for α =1:r*

V(α)=[ ]

for i=1:N

if r(i)=α then V(α)=[V(α), $|\psi_\alpha\rangle$]

end for

Step 3:

for α =1:r* do Steps 4 – 8

Step 4:

Choice randomly $|\psi_\cdot\rangle \in V(\alpha)$ and

$R^\alpha = (|\psi_\cdot\rangle, \{\})$;

$R^\alpha(1) = P'; R^\alpha(2) = \{\}$

U(α)=V(α)\{ $|\psi_\cdot\rangle$ }

Step 5: while U(α)≠Ø do

Step 5.1: choice $|\psi_i\rangle \in U(\alpha)$

Step 5.2: U(α)=U(α)\{ $|\psi\rangle$ }

Step 5.3: construct a chain decomposition C(α)={ $C_1^\alpha, C_2^\alpha \ldots, C_q^\alpha$,} of R^α

Step 6: for i=1:q

Step 6.1: Do binary search on C_i^α using 2n-query-oracle to find smallest element (if any) that dominates $|\psi_i\rangle$

Step 6.2: Do binary search on C_i^α using 2n-query-oracle to find largest element (if any) that is dominated by $|\psi_i\rangle$

Step 7: Infer all results R^α of **6.1** and **6.2** into R^α :

$R^\alpha(1) = R^\alpha(1) \cup |\psi\rangle$

$R^\alpha(2) = R^\alpha(2) \cup R^\alpha$

Step 8:

Step 8.1: Find a chain decomposition C^α of R^α

Step 8.2: Construct ChainMerge(R^α , C^α) data.

Step 9:

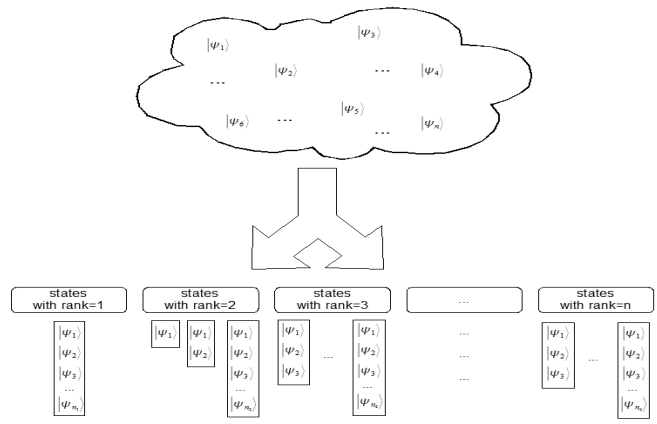
return:

V=[V_1,V_2,…,V_r]

ChainMerge(**V**)=[ChainMerge(V_i),i=1:r]

STOP.

In general case of mixed states the effectively computable function checking whether for a given pair $\{\rho_1, \rho_2\}$ the relation $\rho_1 \prec \rho_2$ holds seems be not available at present [11, 12]. It is long standing problem to construct a computable function on states quantifying the amount of entanglement contained, separating the separable states from entangled one and obeying some additional natural requirements, monotonicity with respect to LOCC operations including [10, 11, 12]. Although some functions obeying all these demands as above have been presented in the literature [1, 10] the real problem with them is that they are hardly to be calculable in an efficient way.



Rys. 2. Ogólna idea sortowania stanów kwantowych, opartych o porządek leksykograficzny wykorzystująca dekompozycję Schmidta. Wynikiem jest struktura odwzorowująca częściowy porządek, gdzie w zbiorze stanów o tym samym rzędzie Schmidta mogą wystąpić ciągi liniowe elementów, które są nieporównywalne

Fig. 2. General idea of sorting quantum states where the lexicographic order is used. In result the obtained structure represents the partial order where in the sets of quantum states with the same Schmidt rank may contain a linear chains which are non-comparable

4. Result of sorting qubits and qudits systems

Let's consider family of the general Bell entangled states for qudits. In case of qubit (qudit with freedom level equal to two) these states have following form

$$B^0 = a|00\rangle + b|11\rangle \text{ and } B^1 = a|01\rangle + b|10\rangle \tag{7}$$

in both cases of course

$$|a|^2 + |b|^2 = 1 \tag{8}$$

If the number a and b are the same we called such states the maximally entangled Bell states. Generalized maximally entangled *d*-level Bell states for two qudits can be expressed by the following equation

$$\left|\psi_{pq}^d\right\rangle = \frac{1}{\sqrt{d}} \sum_{j=0}^{d-1} e^{2\pi i j p/d} |j\rangle |(j+q) \bmod d\rangle, \tag{9}$$

where small letter "*i*" represents the imaginary unit. It is possible to express equation (9) in terms of qudit gates

$$\left|\psi_{pq}^d\right\rangle = (I_d \otimes X_d)^q (H_d \otimes I_d)(Z_d \otimes I_d)^p CNOT_d |00\rangle \tag{10}$$

where symbol "*I*" represents the identify matrix for qudits with d-level, and *H* represents the Hadamard gate and *Z*, *X* are generalized Pauli's operators.

A simple function written in Python which uses the QCS module to generate entangled states is depicted in the Fig. 3. We use this function to make entangled states for earlier created a quantum register.

However these states have always the same amount of entanglement. Therefore function from Fig. 3 must be equipped with some additional unitary gate to modify of entanglement amount. In qubit cases the additional rotation gate after Hadamard gate can be used. In general any random unitary gate may be used to generate Bell states with uniform distribution of amount of entanglement.

```
def make_psi(_r,p,q):
        _r.Reset()
        for i in range(0,q):
                _r.NotN(1)
        _r.HadN(0)
        for i in range(0,p):
                _r.PauliZ(0)
        _r.CNot(0,1)
```

Rys. 3. Funkcja tworząca dowolny splątany stan Bella dla danego rejestru
Fig. 3. The function written in Python preparing the entangled Bell states for given register

Using function from Fig. 3 and computational procedure to calculate of von Neumann entropy it is possible to make a simple benchmark. Additionally, to obtain comparable result we make simple test as a script in Python language for quantum register built only from qubits. The test contains the following computation steps: first we generate n quantum registers, for every register the von Neumann entropy is calculated, and after these steps we sort the obtained list using the classical method called sorting by selection. In Fig. 4 presents the time of work of this simple test.

As we see when we double the number of registers to sort we double the time, because of the use only the selection sort which has theoretical complexity equal to $O(n^2)$.

Number of registers	Linear sorting of quantum states using von Neumann's Entropy (results in secs)
10	0.0008762
100	0.0048304
1000	0.1407390
2000	0.4907356
4000	1.8053686
10000	10.643833

Rys. 4. Czas pracy testu sortującego losowo wygenerowane rejestry kwantowe o różnym poziomie splątania w sensie entropii von Neumann'a
Fig. 4. The time of sorting test which uses the randomly generated quantum registers with a different amount of entanglement in sense of von Neumann's entropy

5. Conclusions

It is clear that still some very important questions have to be answered, the question of how to formulate an effectively calculable query Oracle for comparing the amount of entanglement in the case of mixed states seems to be one of the most important among them. The formulated in the literature procedures like entanglement distillation protocol and similar one are hardly to be used as an effectively computable tool for this purposes.

As in the classical case there do exist several different versions of quantum states sorting process. The mentioned algorithms sorting the space of quantum states together with theirs computational complexity analysis will be presented in our forthcoming publications.

6. References

[1] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information (Cambridge University Press, Cam-bridge, U. K., 2000).
[2] M. A. Nielsen, Conditions for a class of entanglement transformations, Phys. Rev. Lett. 83, 436, 1999.
[3] N. Linden, S. Popescu, J. A. Smolin, Phys. Rev. Lett. 97, 100502 (2006).
[4] C. Daskalakis, R. M. Karp, E. Mossel, S. Riesenfeld ,E. Verbin, Sorting and Selection In Posets, available at http://arxiv.org/abs/0707.1532.
[5] U. Faigle, Gy. Turan, Sorting and Recognition Problems for ordered Sets, SIAM J. Comput. 17(1), pp.: 100-113, 1988.
[6] D. Knuth, The Art of Computer Programming: Sorting and Searching, Massachusetts, Addison-Wesley, 1998.
[7] M. Sawerwain, Parallel algorithm for simulation of circuit and one-way quantum computation models, LNCS: Parallel processing and applied math., 2008, Vol. 4967, pp. 530-539.
[8] M. Sawerwain, R. Gielerak, Natural quantum operational semantics with predicates, Int. J. Appl. Math. Comput. Sci., 2008, Vol. 18, No. 3, pp. 341–359.
[9] R. Gielerak, P. Ratajczak, M. Sawerwain, Nowe funkcjonalności Zielonogórskiego Symulatora Obliczeń Kwantowych, Przegląd Telekomunikacyjny i Wiadomości Telokomunikacyjne, 2008, No 6, pp.: 780-783, (in polish).
[10] K. Zyczkowski, I. Bengtsson, An Introduction to Quantum Entenglament: A Geometric Approach, available at ar-Xiv:quant-phys/0606228v1.
[11] R. Horodecki, P. Horodecki, M. Horodecki, K. Horodecki, Quantum Entanglement, 2007, available at arXiv: quant-phys:/0702225.
[12] O. Guhne, G. Toth, Entanglement detection, 2008, available at arXiv:quant-phys:/0811280.
[13] L. Gurvits, Classical deterministic complexity of Edmonds' problem and Quantum Entanglement, Proc. of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA, also available at quant-ph/0303055.

Artykuł recenzowany