Algoritmo de agrupamiento

May 10, 2021

Sea $A = \sum_i^n a_i P_i$ un observable, donde P_i es un producto tensorial N de matrices de Pauli (y/o matrices identidad) (a P_i se le llama **cadena de Pauli**). Sea un conjunto de bases admisibles $B = \{\mathcal{B}_1, \ldots, \mathcal{B}_m\}$ del espacio $\otimes^N \mathcal{H}$, donde $\mathcal{H}$ es el espacio de Hilbert de un qubit. Se quiere distribuir los P_i en el menor número de grupos de modo que todas las P_i de un mismo grupo sean diagonalizables simultáneamente en una de las bases admisibles.

1 Construcción de bases admisibles a partir de un conjunto de medidas admisibles

Dado $L \leqslant N$, una **medida (parcial)** $\mathcal{E} = \{\mathcal{O}_j\}$ de tamaño L es un conjunto completo de observables compatibles en el espacio $\otimes^L \mathcal{H}$. Es decir, al medir todos los observables $\mathcal{O}_j$ de la medida $\mathcal{E}$ sobre un estado $|\Phi\rangle$ de $\otimes^L \mathcal{H}$, determinaremos $|\Phi\rangle$ unívocamente. Nosotros sólo consideraremos medidas $\mathcal{E}$ donde los observables son del tipo

$$\mathcal{O}_j = |\Psi_j\rangle\langle\Psi_j|\,, \tag{1}$$

de modo que lo que haremos al medir con $\mathcal{E}$ será obtener los coeficientes de $|\Phi\rangle$ en la base $\mathcal{B} = \{|\Psi_j\rangle\}$ de $\otimes^L \mathcal{H}$. Es decir, si $|\Phi\rangle = \sum_j c_j |\Psi_j\rangle$, entonces determinaremos $|c_j|$ midiendo $\mathcal{O}_j$, pues

$$|c_j|^2 = \mathrm{tr}(|\Phi\rangle\langle\Phi|\, \mathcal{O}_j). \tag{2}$$

Así, si un operador P sobre $\otimes^L \mathcal{H}$ es diagonalizable en $\{|\Psi_j\rangle\}$ (en cuyo caso diremos que P es **compatible** con $\mathcal{E}$), de modo que $P = \sum_j d_j |\Psi_j\rangle\langle\Psi_j|$, podremos calcular el valor esperado de P sobre $|\Phi\rangle$ midiendo $\mathcal{E} = \{\mathcal{O}_j\}$:

$$\langle P\rangle_{|\Phi\rangle\langle\Phi|} = \mathrm{tr}(P\,|\Phi\rangle\langle\Phi|) = \sum_j d_j \mathrm{tr}(\mathcal{O}_j\,|\Phi\rangle\langle\Phi|). \tag{3}$$

Consideremos ahora $\mathcal{E}$ y $\mathcal{E}'$ medidas parciales de tamaño L y L', con observables como en (1), entonces $\mathcal{E}\otimes\mathcal{E}' := \{\mathcal{O}_j \otimes \mathcal{O}_{j'}\} = \{|\Psi_j\rangle|\Psi_{j'}\rangle\langle\Psi_j|\langle\Psi_{j'}|\}$ es una medida de tamaño $L+L'$ con observables como en (1), cuya base asociada es $\mathcal{B}\otimes\mathcal{B}' = \{|\Psi_j\rangle|\Psi_{j'}\rangle\}$. Sea P'' es un observable de $\otimes^{N+N'}\mathcal{H}$ que se puede escribir como $P'' = P \otimes P'$, donde P es un observable de $\otimes^L \mathcal{H}$ y P' un observable de $\otimes^{L'}\mathcal{H}$. Entonces, si P es diagonizable en la base asociada a $\mathcal{E}$ (P compatible con $\mathcal{E}$), es decir $P = \sum_j d_j |\Psi_j\rangle\langle\Psi_j|$, y P' es diagonizable en la base asociada a $\mathcal{E}'$ (P' compatible con $\mathcal{E}'$), es decir $P' = \sum_{j'} d_{j'} |\Psi_{j'}\rangle\langle\Psi_{j'}|$, se tiene que podemos computar el valor esperado de P'' sobre un estado $|\Phi\rangle$ de $\otimes^{L+L'}\mathcal{H}$ midiendo $\mathcal{E}''$:

$$\langle P''\rangle_{|\Psi\rangle\langle\Psi|} = \mathrm{tr}(P''\,|\Phi\rangle\langle\Phi|) = \sum_{j,j'} d_j d_{j'} \mathrm{tr}(\mathcal{O}_j \otimes \mathcal{O}_{j'}\,|\Phi\rangle\langle\Phi|). \tag{4}$$

En consecuencia, si un observable P se puede expresar como producto tensorial de observables que son compatibles con ciertas medidas, entonces P será compatible con el producto tensorial de las medidas. En otras palabras, podemos construir una medida compatible con P encontrado medidas parciales para cada uno de los factores tensoriales de P.

Nosotros obtendremos un programa que, dado un chip cuántico, nos diga qué medidas parciales $\mathcal{E}$ podremos realizar. Entonces, las bases admisibles serán las asociadas a las **medidas totales** M (medidas sobre todos los qubits del sistema) que obtengamos al multiplicar tensorialmente estas medidas parciales. El problema pues será elegir qué combinaciones de medidas parciales constituyen el menor conjunto de medidas totales, de modo que todas las cadenas de qubits de A sean compatibles con alguna de estas medidas totales.

2 Código de agrupamiento

2.1 Algoritmo alternativo a LDFC para el agrupamiento TPB

Esto es un poco una tontería, pero creo que para hacer el agrupamiento TPB se puede usar un código algo más rápido, que no necesita recorrer todos los vértices de manera no trivial (el LDFC cada vez que pasa por un vértice tiene que chequear cosas). Además, para agrupar por TPB no hace falta ordenar los vértices del grafo de Pauli de ninguna manera.

Construiremos el grafo de Pauli de una manera distinta, poniendo el lado entre P_i y P_j si P_i y P_j son diagonalizables en alguna base TPB.

```
Data: Un observable A = ∑_i^n a_i P_i
Inicializa un grafo con G=(V,E) con n vértices;
for i=1,...,n-1 do
    for j=i+1,...,n do
        if Todos los factores de P_i y P_j conmutan then
            Añadir el vértice (i, j);
        end
    end
end
```

Algorithm 1: Construcción de grafo de Pauli alternativo

Entonces, construimos los grupos aunando los vértices de las componentes conexas del grafo (seguramente para esto haya un buen código ya implementado):

```
Data: Un grafo de Pauli alternativo
Inicializa los colores C;
for i=1,...,n do
    if i no ha sido agrupado then
        A cada i y a cada j adyacente a i, le asignamos el color más bajo no
          asignado;
    end
end
```

Algorithm 2: Algoritmo alternativo a LDFC

2.2 Algoritmo de agrupamiento general adaptado a la arquitectura del sistema

Para adaptar el código de los japoneses a la arquitectura del chip, basta meter un 'if' en su segundo algoritmo.

Data: Observable $A = \sum_i^n a_i P_i$, un conjunto medidas parciales admisibles $E = \{\mathcal{E}_1, \ldots, \mathcal{E}_l\}$, un conjunto de pares de qubits con buena conectividad BC.

Hagamos un grafo de pauli G y ordenemos los índices de mayor a menor grado *# o de menor a mayor grado si usamos el grafo de Pauli alternativo*;

Inicialicemos las medidas M_i. *#Las construiremos localmente, para conseguir que M_1 sea compatible con v_1 y con el mayor número de otras cadenas posible, con M_2 haremos lo mismo, pero pasando de las cadenas que son compatibles con M_1, …*;

for *i=1,…,n* **do**
    if *v_i no está asignado a una medida* **then**
        for *j=i+1,…,n* **do**
            if *v_j no está asignado a una medida* **then**
                Algoritmo 4 *Si v_i y v_J son compatibles con la medida M_i asigna M_i a v_j y actualiza M_i*
            end
        end
    end
end

Devuelve los grupos y las medidas/bases asignadas a cada grupo.

Algorithm 3: Algoritmo de agrupamiento general

El siguiente algoritmo, se inicializa con dos cadenas de Pauli (v_i, v_j), un conjunto de medidas parciales admisibles E, un conjunto de pares de qubits con buena conectividad BC, una asignación actual de la medida M_i. Lo primero, chequea si v_j es compatible con M_i. En la primera iteración en j, M_i estará vacía, así que v_j será compatible con M_i. Entonces, el algoritmo busca un conjunto de medidas parciales que conformen una medida total que sea compatible con v_i y v_j. Para ello, lo primero que hace es detectar lo qubits en los que los factores de v_i y v_j coinciden. En esos qubits podemos usar medidas TPB (a estos no se les asigna medida, se los usa como comodines, pues es en estos qubits en los que buscamos medidas alternativas en futuras iteraciónes en j) *(además, estos qubits no pueden dar lugar a medidas parciales que hagan compatibles a otros qubits, con lo que la única forma de medir simultáneamente en ellos es con TPB)*. Seguidamente, se comprueba si otras medidas parciales (que no sean necesariamente TPB, por lo que entran en juego las medidas entrelazadas) pueden hacer compatibles el resto de qubits; en caso afirmativo, actualizamos M_i con esas medidas parciales y a v_j le asignamos la medida M_i. Si no devolvemos fallo.

En una iteración j en la que no se parta de una M_i vacía, es posible que v_j no sea compatible con M_i y que obtengamos fallo. Si no, entramos en el algoritmo de asignación y creamos un vector U con los qubits a los que M_i no les ha asignado ninguna medida (por construcción, en estos qubits coinciden todos los factores de las cadenas asignadas a M_i). A U le quitamos los qubits en los que los factores de v_i y v_j coincidan (y, por construcción, en estos qubits los factores de v_j coincidirán con todos los factores de las cadenas asignadas a M_i; en particular, en estos qubits podremos medir simultáneamente con TPB). Los qubits que permanecen en U seguirán satisfaciendo que todos los factores de las cadenas asignadas a M_i coinciden, por lo que para hallar medidas parciales compatibles con v_j y con todas las cadenas que están asignadas a M_i, basta fijarse en v_j y v_i. Si se consiguen hallar tales medidas, actualizamos M_i con ellas y a v_J le asignamos la medida M_i. Si no, devolvemos fallo.

Nosotros hemos de meter un if, para chequear si los qubits sobre los que actúa cada medida parcial están bien conectados.

Data: Dos vértices de un grafo de Pauli (v_i, v_j), un conjunto de medidas parciales admisibles E, un conjunto de pares de qubits con buena conectividad BC, una asignación actual de la medida M_i

Devuelve **fallo** si v_i no es compatible con alguna de las medidas parciales asignadas a M_i;

Sea U el vector de los índices de los qubits que no tienen una medida parcial asignada;

Quitemos a U los índices de los qubits donde v_i y v_j coincidan;

```
while U ≠ ∅ do
    for ℰ ∈ E do
        *** Aquí quizá sea importante elegir correctamente el orden de las bases;
        for p ∈ permutaciones en subconjuntos de U de tamaño igual al de ℰ do
            Aquí es donde imponemos que la medida sólo involucre a los qubits bien
             conectados;
            if si los qubits sobre los que actúa p están en BC then
                if v_i y v_j son compatibles con ℰ sobre los qubits de p then
                    Actualizamos M_i con ℰ;
                    Quitamos los qubits de p de U;
                    Volvemos al while;
                end
            end
        end
    end
    Devuelve fallo
end
```

Devuelve la asignación actualizada de la medida M_i y asigna M_i a v_j.

Algorithm 4: Algoritmo de asignación de medidas teniendo en cuenta la conectividad

3 Ideas/problemas a atacar

- **Encontrar medidas entrelazadas que puedan hacer compatibles subcadenas de dos qubits**. Los autores ya proponen algunas útiles: la de Bell para medir a la vez $\sigma_X \otimes \sigma_X$, $\sigma_Y \otimes \sigma_Y$ y $\sigma_Z \otimes \sigma_Z$, la Ω^X para medir a la vez $\sigma_X \otimes \sigma_X$, $\sigma_Y \otimes \sigma_Z$ Y $\sigma_Z \otimes \sigma_Y, \ldots$ Pero, por ejemplo, $\sigma_X \otimes \sigma_Z$ y $\sigma_Y \otimes \sigma_X$ pueden medirse simultáneamente y los autores no proponen una medida para ello (ver la tabla de compatibilidades, donde S significa que sí son compatibles y N que no). Además, puede que las casillas que ya tienen medida asignada puedan tener otras medidas asignadas (como la de $\sigma_X \otimes \sigma_X$ y $\sigma_Y \otimes \sigma_Y$), lo que puede dar pie a formar menos grupos.

	$\sigma_X \otimes \sigma_X$	$\sigma_Y \otimes \sigma_Y$	$\sigma_Z \otimes \sigma_Z$	$\sigma_X \otimes \sigma_Y$	$\sigma_X \otimes \sigma_Z$	$\sigma_Y \otimes \sigma_X$	$\sigma_Y \otimes \sigma_Z$	$\sigma_Z \otimes \sigma_X$	$\sigma_Z \otimes \sigma_Y$
$\sigma_X \otimes \sigma_X$	-	S(Muchas,Bell)	S(Bell)	N	N	N	S(Ω^X)	N	S(Ω^X)
$\sigma_Y \otimes \sigma_Y$	S	-	S(Bell)	N	S(Ω^Y)	N	N	S(Ω^Y)	N
$\sigma_Z \otimes \sigma_Z$	S	S	-	S(Ω^Z)	N	S(Ω^Z)	N	N	N
$\sigma_X \otimes \sigma_Y$	N	N	S	-	N	S(Ω^Z)	S(χ)	S(χ)	N
$\sigma_X \otimes \sigma_Z$	N	S	X	N	-	S	N	S(Ω^Y)	S
$\sigma_Y \otimes \sigma_X$	N	N	S	S	S	-	N	N	S
$\sigma_Y \otimes \sigma_Z$	S	N	N	S	N	N	-	S(χ)	S(Ω^X)
$\sigma_Z \otimes \sigma_X$	N	S	N	S	S	N	S	-	N
$\sigma_Z \otimes \sigma_Y$	S	N	N	N	S	S	S	N	-

- **Dado que vamos a tener en cuenta la conectividad del chip, la correspondencia de qubits del observable y qubits del chip que elijamos es relevante.** Por ejemplo, si tenemos un observable $A = \sigma_X \otimes \sigma_Z \otimes \sigma_Y + \sigma_Y \otimes \sigma_Z \otimes \sigma_X$ y un chip de 3 qubits con los qubits 1 y 2 bien conectados y 2 y 3 bien conectados, no podremos aprovechar las medidas entrelazadas que podemos hacer entre los qubits 1 y 2, pues $\sigma_X \otimes \sigma_Z$ y $\sigma_Y \otimes \sigma_Z$ no conmutan (ver la tabla), y lo mismo ocurre con los qubits 2 y 3. De este modo, el algoritmo arrojará 2 grupos. Ahora, si reescribimos el observable, usando otra correspondencia entre qubits de A y qubits del chip, como $A = \sigma_Z \otimes \sigma_Y \otimes \sigma_X + \sigma_Z \otimes \sigma_X \otimes \sigma_Y$, entonces podremos agrupar las 2 cadenas, puesto que usando la conectividad de los qubits 2 y 3, podremos medir $\sigma_Y \otimes \sigma_X$ y $\sigma_X \otimes \sigma_Y$ simultáneamente (ver tabla).

- **Aunque el algoritmo de agrupamiento sea lento, quizá podamos obtener resultados interesantes.** Esto se debe a que si, dado una tamaño fijo de las cadenas N, conseguimos distribuir en un número suficientemente pequeño de grupos las 4^N cadenas posibles ya habremos obtenido un **agrupamiento universal para las cadenas de tamaño** N. Así, aunque nuestro algoritmo sea lento, si es capaz de agrupar en un tiempo no demasiado extremo (digamos no más de una semana) un observable que tenga las 4^N cadenas posibles, entonces habremos obtenido un resultado general, que se podrá mejorar para casos particulares.

- Quizá el algoritmo de asignación que los autores proponen se pueda mejorar. Hay detalles que dejan un poco en el aire, como ***.

- Aunque no resolvamos estas cuestiones, podemos proponerlas como problemas a resolver.