

Ein Quantenalgorithmus zur Berechnung des Minimums

Sei x_1, \dots, x_n eine (unsortierte) Tabelle mit Elementen einer geordneten Menge. Der Einfachheit halber nehmen wir an, dass alle Elemente verschieden sind. Der Zugriff auf die Tabelle erfolgt über ein Orakel, also einen Schaltkreis bzw. Quantenschaltkreis, der zu einem Index i das Element x_i ausgibt. Ein klassischer Algorithmus benötigt offensichtlich eine lineare Anzahl von Zugriffen auf die Tabelle, um das Minimum zu finden. Wir wollen hier den Grover-Algorithmus benutzen, um das Minimum auf eine schnellere Weise zu finden. Wir gehen dazu in zwei Schritten vor. Zuerst konstruieren wir einen Algorithmus, der endlos lange läuft. Anschließend analysieren wir die erwartete Anzahl an Iterationen, bis der Algorithmus das Minimum gefunden hat. Durch Anwenden der Markov-Ungleichung können wir ein Abbruchkriterium für die Endlosschleife finden.

Der Algorithmus arbeitet auf die folgende Weise.

1. Wähle zufällig gemäß Gleichverteilung ein $j_0 \in \{1, \dots, n\}$. Sei $\ell := 0$.
2. FOREVER DO
 - Definiere $f_{j_\ell}(i) = 1 \Leftrightarrow x_i < x_{j_\ell}$.
 - Suche mit dem Grover-Algorithmus nach einem i mit $f_{j_\ell}(i) = 1$.
 - Falls $x_i < x_{j_\ell}$, sei $\ell := \ell + 1$; $j_\ell := i$.

Offensichtlich ist die Folge $S = (x_{j_0}, x_{j_1}, \dots)$ streng monoton fallend. Wir schätzen nun für die Elemente x_1, \dots, x_n die Wahrscheinlichkeit ab, dass sie in der Folge S vorkommen. Unter dem Rang eines Elements x_i verstehen wir seine Position in der Eingabe, nachdem diese aufsteigend sortiert wurde. Das gesuchte Minimum hat also den Rang 1.

Lemma: Sei t der Rang des Vergleichselementes x_{j_ℓ} aus der Folge S und sei x_i ein Element mit Rang r . Sei $p(t, r)$ die Wahrscheinlichkeit, dass x_i von dem Algorithmus hinter x_{j_ℓ} in die Folge S aufgenommen wird (nicht notwendigerweise direkt hinter x_{j_ℓ}). Es gilt

$$p(t, r) = \begin{cases} 1/r, & \text{falls } r < t, \\ 0, & \text{falls } r \geq t. \end{cases}$$

Beweis. Die Aussage für den Fall $r \geq t$ ist offensichtlich. Die Aussage für den Fall $r < t$ beweisen wir mit Induktion über t , wobei r als fest angenommen wird. Der Induktionsanfang ist $t = r + 1$. Der Grover-Algorithmus wählt ein i mit $f_{j_\ell}(i) = 1$ zufällig gemäß Gleichverteilung aus allen i mit $f_{j_\ell}(i) = 1$. Da es davon $t - 1$ viele gibt und $t - 1 = r$ ist, ist die Wahrscheinlichkeit, das Element mit Rang r in diesem Schritt zu wählen, $1/r$. Wird dagegen in diesem Schritt ein Element mit einem Rang kleiner als r gewählt, kann das Element mit Rang r später nicht mehr gewählt werden.

Wir nehmen nun an, dass die Aussage $p(k, r) = 1/r$ für alle $k \in \{r + 1, \dots, t\}$ bewiesen ist und betrachten die Situation, dass das Vergleichselement den Rang $t + 1$ hat. Dann wird unter den Elementen mit den Rängen $1, \dots, t$ gemäß Gleichverteilung ein Element ausgewählt, das entweder (i) einen Rang kleiner als r , (ii) den Rang r oder (iii) einen Rang aus $\{r + 1, \dots, t\}$ hat. Im Fall (i) wird das Element mit Rang r (auch später) nicht ausgewählt, im Fall (ii) wird es

in diesem Schritt ausgewählt und im Fall (iii) wird es später mit einer Wahrscheinlichkeit von $p(k, r)$ ausgewählt, wobei k der Rang des gewählten Elements und damit aus $\{r + 1, \dots, t\}$ ist. Also gilt

$$p(t + 1, r) = \frac{1}{t} + \sum_{k=r+1}^t \frac{1}{t} p(k, r) = \frac{1}{t} + \frac{t-r}{t} \cdot \frac{1}{r} = \frac{1}{r}.$$

Das zweite Gleichheitszeichen folgt dabei aus der Induktionsannahme. \square

Wir können nun die erwartete Rechenzeit abschätzen, bis der Algorithmus das Minimum gefunden hat.

Lemma: *Die erwartete Rechenzeit, bis der Algorithmus das Minimum gefunden hat, ist $O(\log n \sqrt{n})$.*

Beweis. Die erwartete Rechenzeit des Grover-Algorithmus bis zu einem erfolgreichen Suchergebnis ist durch $c\sqrt{n}$ für eine geeignete Konstante c beschränkt. Also kann man die Rechenzeit zwischen zwei erfolgreichen Suchen durch $c'\sqrt{n}$ abschätzen. Wir beachten dabei, dass diese Rechenzeit sich über mehrere Iterationen der Schleife erstrecken kann. Für jedes Element mit Rang r beträgt die Wahrscheinlichkeit, dass es gewählt wird, $1/r$; die Rechenzeit zwischen der Suche, in der ein Element mit einem Rang $r > 1$ gefunden wird und der nächsten erfolgreichen Suche, kann, wie gesagt, durch $c'\sqrt{n}$ abgeschätzt werden. Die erwartete Rechenzeit ist daher durch

$$\sum_{r=2}^n \frac{1}{r} c' \sqrt{n} + O(\log n) = c'(H_n - 1)\sqrt{n} + O(\log n) \leq c'' \log n \sqrt{n}$$

nach oben beschränkt, wobei H_n die n -te harmonische Zahl ist, also $H_n = 1 + 1/2 + 1/3 + \dots + 1/n$. Wir benutzen dabei die Abschätzung $H_n \leq \ln n + 1$. Der Term $O(\log n)$ steht für die Initialisierung von j_0 . \square

Die Wahrscheinlichkeit, dass die Rechenzeit $d \cdot c'' \log n \sqrt{n}$ übersteigt, ist dann nach der Markov-Ungleichung durch $1/d$ beschränkt. Wir erhalten also für jede Konstante d einen Algorithmus für die Berechnung des Minimums mit einer Erfolgswahrscheinlichkeit von mindestens $1 - 1/d$ und der Rechenzeit $O(\log n \sqrt{n})$. Mit genaueren Abschätzungen erhält man sogar die Rechenzeit $O(\sqrt{n})$. Wir beachten, dass wir einen Monte-Carlo-Algorithmus erhalten, da nicht klar ist, wie man effizient testen kann, ob das berechnete Element wirklich das Minimum ist.

Quelle: Dürr, C. und Høyer, P. (1996). A quantum algorithm for finding the minimum. *Quantph/9607014*.