

Quantum search with prior knowledge

Xiaoyu HE^{1,2}, Xiaoming SUN^{1,2*} & Jialing ZHANG^{1,2}¹State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China;²School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China

Received 23 August 2023/Revised 22 November 2023/Accepted 8 March 2024/Published online 20 August 2024

Abstract The combination of contextual side information and search is a powerful paradigm in the scope of artificial intelligence. The prior knowledge enables the identification of possible solutions but may be imperfect. Contextual information can arise naturally, for example in game AI where prior knowledge is used to bias move decisions. In this work we investigate the problem of taking quantum advantage of contextual information, especially searching with prior knowledge. We propose a new generalization of Grover's search algorithm that achieves the optimal expected success probability of finding the solution if the number of queries is fixed. Experiments on small-scale quantum circuits verify the advantage of our algorithm. Since contextual information exists widely, our method has wide applications. We take game tree search as an example.

Keywords quantum computing, quantum search, quantum query algorithm, prior knowledge

1 Introduction

Searching is one of the most important paradigms for algorithm design. Many computationally difficult problems, including deciphering some popular encryption schemes [1, 2], Monte Carlo tree search for game [3] and NP-hard problems [4] can be reduced to searching problems. Grover's quantum search algorithm [5, 6] shows a quadratic speedup over the classical algorithm. The problem is represented by an Oracle function, which gives the solution by flipping the sign of the corresponding quantum state, among an unsorted list of size N .

Grover's search algorithm applies Oracle and preparation of the initial state iteratively to implement the rotation on the 2-dimensional Hilbert space spanned by the initial state and target state. It rotates a fixed angle, determined by size N and number of solutions, towards the target every time and it may miss the solution when the fixed angle is not exact. Quantum amplitude amplification [7] generalized Grover's standard algorithm to make the success probability of the search reach 100%. The exact query complexity of this algorithm to find one solution among N items exactly is the same as Grover's algorithm, $\lceil \frac{\pi}{4 \arcsin \sqrt{1/N}} - \frac{1}{2} \rceil \approx O(\sqrt{N})$, which is optimal [8]. When required to output an answer in a smaller limited number of queries, repeating the iteration of Grover's algorithm can reach the maximum success probability as well.

These searching algorithms are also suitable for finding one of M solutions among items of size N . However, the case is complicated when the number of solutions is unknown since too few iterations cannot reach the target state and too many iterations may pass by the target state. Grover's $\pi/3$ -algorithm [9] and a critically damped quantum search algorithm [10] were proposed to handle this fixed point search problem. Then an amplitude amplification improvement [11] was made to achieve a quadratic speed-up of query complexity. An exact algorithm for weight decision has also been proposed recently [12].

Grover's quantum search algorithm and the above generalizations consider all the items to be searched to be of equal status. However, the items are rarely equally important. We note that heuristic methods are commonly used in classical searching algorithms. For practical search problems, some prior knowledge is

* Corresponding author (email: sunxiaoming@ict.ac.cn)

usually known about where the solution is more likely to be located. Then a biased probability distribution can be obtained from the current problem to instruct how to prune branches or enter branches of the search tree randomly to improve the expected success probability of the search algorithm. A famous modern example is Google's go AI: AlphaGo and AlphaZero [13, 14], despite the emphasis of their work being machine learning, the subroutine of searching for the best placement is instructed by knowledge drawn from current patterns and checking whether a placement is good by further recursion. A quantum algorithm has also been proposed recently to solve learning problems [15].

In this work, we focus on how to take advantage of prior knowledge to improve quantum search. The prior knowledge can usually be formalized as a probability distribution $\mathbf{p} = (p_1, p_2, \dots, p_N)$, which indicates the probability of the location of the solution. Let us first consider the classical algorithm in the face of such a situation. Assuming $p_1 \geq p_2 \geq \dots \geq p_N$, then the optimal classical search algorithm with T queries is to query the item from 1 to T , and the maximum expected success is obviously $\sum_{i=1}^T p_i$. What about the quantum algorithm? To what extent can we surpass the classical counterpart? A naive example that shows the advantage of prior knowledge is $\mathbf{p} = (0.25, 0.25, 0.25, 0.25, 0, 0, 0, 0)$ with $N = 8$, the success probabilities of quantum algorithms in one Oracle query with and without this prior knowledge are 100% and about 78.1% respectively, while the classical success probability in one query is 25%. If we consider another example with similar prior knowledge $(0.24, 0.24, 0.24, 0.24, 0.01, 0.01, 0.01, 0.01)$ with $N = 8$, can we still achieve a success probability close to 100% with one Oracle query? How close can we achieve? In this work, we give definite answers to all these questions.

Our result shows that by a variant of the quantum search algorithm with some parametric operations related to the given probability distribution, we can improve the expected success probability to optimal with any limited number of Oracle queries.

Montanaro [16] shows an algorithm that finds the solution with the asymptotically optimal expected number of queries. His quantum algorithm is a Las Vegas algorithm, that repeats many times until finding the solution. The expected number of Oracle queries is proven a constant multiple of the optimal value. Nevertheless, the constant complexity is not unimportant. An algorithm with an asymptotically optimal running time is not good enough for time-sensitive tasks, such as game AI which must output an answer in a limited time the running time directly determines the strength of artificial intelligence. Our result can offer an improvement of the success probability of finding the best choice with some prior knowledge when running time is limited.

On the other hand, we note that on noisy intermediate-scale quantum (NISQ) computers, the fidelity of the circuit decreases rapidly as the number of Oracle queries increases. A three-qubit Grover search experiment on a scalable quantum computing system [17] uses only one Oracle query, it seems that searching one solution among eight items using one Oracle query performs better than using two Oracle queries, although the relationship between their theoretical success probability is opposite. A quantum algorithm solving subset sum problem has also been demonstrated successfully [18]. Our result can offer a trade-off between theoretical success probability and experimental error of the circuit, which results in an algorithm with a suitable number of Oracle queries to reach optimal experiment success probability.

2 Search with prior knowledge

Grover's quantum search algorithm can be described as a rotation in the space spanned by the initial state and the target solution state. Given the probability distribution $\mathbf{p} = (p_1, p_2, \dots, p_N)$ as prior knowledge about where the solution might be located, we show that by replacing the initial state with $|s\rangle = \sum_{i=1}^N \sqrt{q_i} |i\rangle + \sqrt{1 - \sum_{i=1}^N q_i} |0\rangle$, where q_i are some parameters that $q_i \geq 0$ and $\sum_{i=1}^N q_i \leq 1$, we can improve the expected success probability.

We first analyze the success probability of our proposed algorithm with parameters \mathbf{q} . Let $R_s = I - 2|s\rangle\langle s|$, which is the reflection operation about $|s\rangle$. Consider the transformation $(R_s O_x)^T$ applied to $|s\rangle$. For a fixed solution x , let $|x^\perp\rangle = 1/(1 - q_x) \sum_{i \neq x} \sqrt{q_i} |i\rangle$, the state after $(R_s O_x)^T$ applied to $|s\rangle$ is $\sin((2T + 1) \arcsin \sqrt{q_x}) |x\rangle + \cos((2T + 1) \arcsin \sqrt{q_x}) |x^\perp\rangle$. The success probability of getting solution x is $\sin^2((2T + 1) \arcsin \sqrt{q_x})$ if we use measurement with a standard basis. Hence given distribution $\mathbf{p} = (p_1, p_2, \dots, p_N)$ indicating where the solution might be, for any non-negative vector $\mathbf{q} = (q_1, \dots, q_N)$

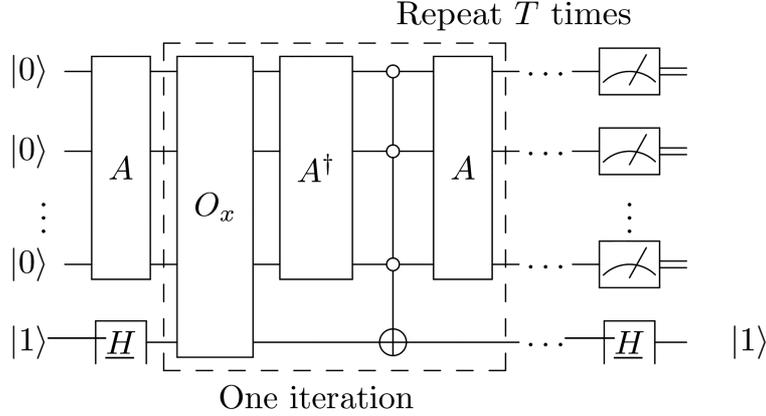


Figure 1 We provide a circuit for our quantum search algorithm in T Oracle queries. Here A is a unitary that prepares the initial state $A|0\rangle = \sum_{i=1}^N \sqrt{q_i}|i\rangle + \sqrt{1 - \sum_{i=1}^N q_i}|0\rangle$. O_x is the Oracle that flip an ancilla qubit, that $O_x|x\rangle|b\rangle = |x\rangle|b \oplus 1\rangle$ and $O_x|x^\perp\rangle|b\rangle = |x\rangle|b \oplus 1\rangle$.

that $\sum_{i=1}^N q_i \leq 1$, the expected success probability of quantum search in T queries can be

$$\text{ESP}_T(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^N p_i \sin^2((2T + 1) \arcsin \sqrt{q_i}). \quad (1)$$

Therefore we can choose \mathbf{q} to optimize the success probability. Note that $\sin^2((2T + 1) \arcsin \sqrt{q_i})$ reaches its maximum value 1 when $q_i = \sin^2 \frac{\pi}{2(2T+1)}$ and it increases monotonously with q_i in $[0, \sin^2 \frac{\pi}{2(2T+1)}]$, so we can add extra constraints that $q_i \leq \sin^2 \frac{\pi}{2(2T+1)}$, without decreasing the maximum value of (1). Let $\mathbf{q}^*(\mathbf{p})$ denote the parameters that maximize (1).

$$\mathbf{q}^*(\mathbf{p}) = \arg \max_{\mathbf{q}} \text{ESP}_T(\mathbf{p}, \mathbf{q}) \quad (2)$$

$$\text{s.t. } \sum_{i=1}^N q_i \leq 1, \quad (3)$$

$$\forall i : 0 \leq q_i \leq \sin^2 \frac{\pi}{2(2T + 1)}. \quad (4)$$

Since $\sin^2((2T + 1) \arcsin \sqrt{q_i})$ is a concave function when $q_i \in [0, \sin^2 \frac{\pi}{2(2T+1)}]$, $\sum_{i=1}^N p_i \sin^2((2T + 1) \arcsin \sqrt{q_i})$ is a concave function about (q_1, \dots, q_N) . Eq. (2) is a concave optimization with linear constraints, which can be solved as a convex optimization problem by Lagrange multiplier method, gradient descent or some other contemporary methods. For any given probability distribution, the corresponding state can be prepared efficiently [19]. We show a quantum algorithm framework that can achieve that value in Figure 1.

We summarize the main theoretical result as Theorem 1.

Theorem 1. Given probability distribution $\mathbf{p} = (p_1, \dots, p_N)$ indicating which item might be the solution, the maximum expected success probability of finding the solution in T Oracle queries is

$$\max_{\mathbf{q}} \sum_{i=1}^N p_i \sin^2((2T + 1) \arcsin \sqrt{q_i}) \quad (5)$$

$$\text{s.t. } \sum_{i=1}^N q_i \leq 1, \quad (6)$$

$$\forall i : 0 \leq q_i. \quad (7)$$

We claim that our algorithm is optimal, which means the maximum expected success probability in T queries can be reached by our algorithm, regardless of whether the algorithm is adapted based

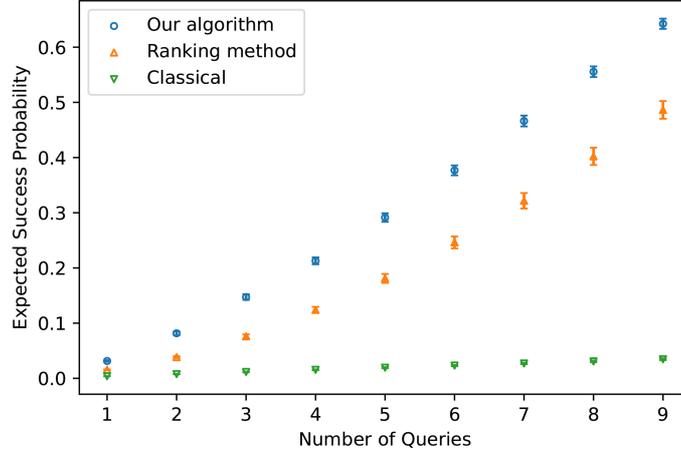


Figure 2 (Color online) Comparison of the expected success probability between classical, ranking method and our algorithm for probability distributions of size 512 which are generated uniformly randomly that generate each p_i uniformly randomly in $[0, 1]$ then normalize \mathbf{p} , with 100 samples of the probability distribution for every number of queries. Here ranking method refers to calling Grover's search algorithm among items with the highest probability.

on intermediate measurements or using variational diffusion operators. The optimality is proved in Appendix A.

When the number of Oracle queries is limited to T , a naive quantum algorithm is to call Grover's algorithm among the top $\lfloor \frac{1}{\sin^2 \frac{\pi}{2(2T+1)}} \rfloor$ items after ranking by probability values. To achieve the same expected success probability, the classical algorithm requires at least $\lfloor \frac{1}{\sin^2 \frac{\pi}{2(2T+1)}} \rfloor \geq T^2$ queries, which means quantum has at least quadratic speed-up over classical when given probability distribution of the solution. Our algorithm performs obviously better than the naive ranking method in that it provides 25% improvement when the number of queries is not close to \sqrt{N} , as shown in Figure 2.

Although the perfect information about complete distribution can help to improve the search success rate optimally by our algorithm, partial information can also help improve search in a simple way that keeps the known distribution and lets the unknown items share the remaining probability. For example we know that the first item is 50% likely to be the answer among 8 items, we may assuming the probability distribution $(0.5, \frac{1}{14}, \frac{1}{14}, \frac{1}{14}, \frac{1}{14}, \frac{1}{14}, \frac{1}{14}, \frac{1}{14})$, and the perform our algorithm as above.

Finally, we will show that $\mathbf{q}^*(\mathbf{p})$ is robust to the perturbation of \mathbf{p} . If we have some imperfect information, an estimation of distribution $\hat{\mathbf{p}}$ that the L_1 distance $d_1(\mathbf{p}, \hat{\mathbf{p}}) = \sum_{i=1}^N |p_i - \hat{p}_i| \leq \epsilon$, then

$$\begin{aligned}
 & \text{ESP}_T(\mathbf{p}, \mathbf{q}^*(\hat{\mathbf{p}})) \\
 &= \text{ESP}_T(\hat{\mathbf{p}}, \mathbf{q}^*(\hat{\mathbf{p}})) + \text{ESP}_T(\mathbf{p} - \hat{\mathbf{p}}, \mathbf{q}^*(\hat{\mathbf{p}})) \\
 &\geq \text{ESP}_T(\hat{\mathbf{p}}, \mathbf{q}^*(\mathbf{p})) + \text{ESP}_T(\mathbf{p} - \hat{\mathbf{p}}, \mathbf{q}^*(\hat{\mathbf{p}})) \\
 &= \text{ESP}_T(\mathbf{p}, \mathbf{q}^*(\mathbf{p})) + \text{ESP}_T(\hat{\mathbf{p}} - \mathbf{p}, \mathbf{q}^*(\mathbf{p})) \\
 &\quad + \text{ESP}_T(\mathbf{p} - \hat{\mathbf{p}}, \mathbf{q}^*(\hat{\mathbf{p}})) \\
 &\geq \text{ESP}_T(\mathbf{p}, \mathbf{q}^*(\mathbf{p})) - \sum_{i=1}^N |p_i - \hat{p}_i| \\
 &\geq \text{ESP}_T(\mathbf{p}, \mathbf{q}^*(\mathbf{p})) - \epsilon.
 \end{aligned}$$

This means that a good estimation of the precise distribution can approach the optimal expected success probability well.

3 Experimental results

We implement our algorithm on a five-qubit quantum computer provided by IBMQ [20]. We consider searching among items of size 8 with one solution. The probability distribution of the solution is a simple biased distribution formed as $(\frac{1}{8} + \sigma, \frac{1}{8} + \sigma, \frac{1}{8} + \sigma, \frac{1}{8} + \sigma, \frac{1}{8} - \sigma, \frac{1}{8} - \sigma, \frac{1}{8} - \sigma, \frac{1}{8} - \sigma)$, for $\sigma \in \{\frac{1}{80}, \frac{2}{80}, \frac{3}{80}, \frac{4}{80}, \frac{5}{80}, \frac{6}{80}, \frac{7}{80}, \frac{8}{80}\}$.

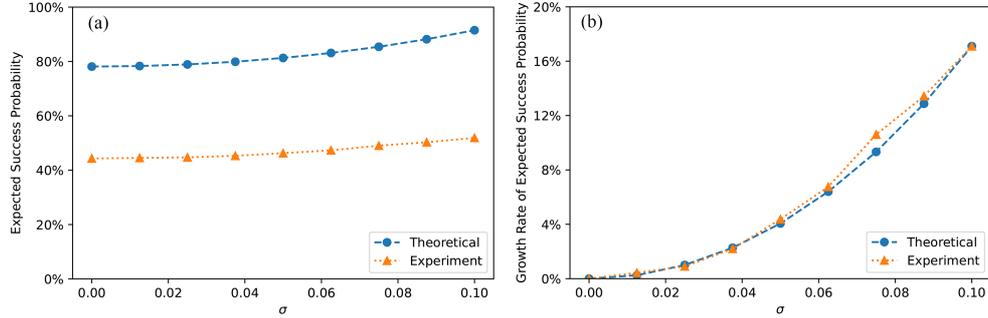


Figure 3 (Color online) Quantum search using a single Oracle query, for finding one solution among 8 items on 3 qubits of IBM’s 5-qubit computing system ibmqx2. (a) Comparison of the theoretical maximum expected success probability and the experiment result of expected success probability. (b) Comparison of the theoretical growth rate and the experimental growth rate of maximum expected success probability than directly calling Grover’s algorithm without using prior knowledge. The growth rate here is the ratio of the expected success probability with prior knowledge to the expected success probability directly calling Grover’s algorithm without using prior knowledge, then minus 1.

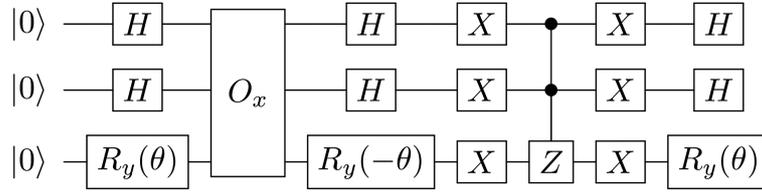


Figure 4 A 3-qubit circuit for a ‘half-half’ distribution in 1 Oracle query. Here θ depends on the deviation of the probability distribution.

Table 1 A table of the eight Oracles for $N = 8$

	000	001	010	011
Circuit	$ q_1\rangle$ — X — X $ q_2\rangle$ — X — X $ q_3\rangle$ — X — Z — X	$ q_1\rangle$ — X — X $ q_2\rangle$ — X — X $ q_3\rangle$ — Z	$ q_1\rangle$ — X — X $ q_2\rangle$ — X — X $ q_3\rangle$ — X — Z — X	$ q_1\rangle$ — X — X $ q_2\rangle$ — X — X $ q_3\rangle$ — Z
	100	101	110	111
Circuit	$ q_1\rangle$ — X — X $ q_2\rangle$ — X — X $ q_3\rangle$ — X — Z — X	$ q_1\rangle$ — X — X $ q_2\rangle$ — X — X $ q_3\rangle$ — Z	$ q_1\rangle$ — X — X $ q_2\rangle$ — X — X $ q_3\rangle$ — X — Z — X	$ q_1\rangle$ — X — X $ q_2\rangle$ — X — X $ q_3\rangle$ — Z

We conclude the experiment results in Figures 3(a) and (b). Data in these figures were obtained on IBMQ [20] hardware with ‘ibmqx2’ backend, for eight biased and one uniform probability distribution, eight different Oracles, 8192 shots for each. The CNOT gate and Readout error on this device is about 5%. The coherent time of qubits is about 50 μ s. We have run our algorithm on theoretical noiseless simulators and actual noisy devices. It shows that the experimental success probability is lower than the theoretical, which is caused by the noise of the device. However, with respect to the growth rate of success probability caused by prior knowledge, the experiment result agrees with the theory well.

The circuit implementation of our algorithm is shown in Figure 4, the corresponding parameters \mathbf{q}^* and θ for different probability distributions are provided in Appendix B. The eight Oracles for different solutions is implemented directly similarly to those in the textbook [21], as shown in Table 1.

4 Application: heuristic tree search

Since contextual information arises naturally in the scope of artificial intelligence [22]. We can hopefully expect that our quantum method has widespread applications such as quantum exploration algorithm for multi-armed bandits [23] with episode context and game tree search with some prior knowledge. We show how to apply our method to game tree search as an example.

When playing a game, an interesting problem is whether and how a player can win. By formalizing

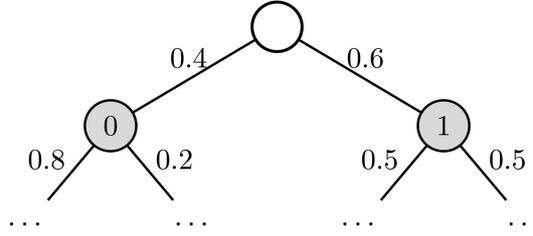


Figure 5 Example of a weighted search tree, where the edge weights indicate the probability that each choice is good.

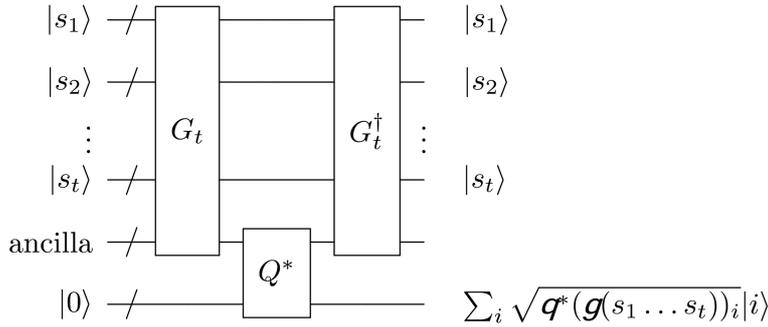


Figure 6 We provide a circuit for generating the optimal initial state corresponding to the distribution $\mathbf{g}(s_1 \cdots s_t)$ for our algorithm. G_t calculates the probability distribution $\mathbf{g}(s_1 \cdots s_t)$ as a classical algorithm and stores it to ancilla qubits. Q^* calculates $\mathbf{q}^*(\mathbf{g}(s_1 \cdots s_t))$ as a classical convex optimization algorithm on ancilla qubits, generates the corresponding quantum state, and then recovers the ancilla qubits. Then ancilla qubits are recovered by reversing the circuit.

a game as a tree whose nodes are positions in the game and edges are actions taken by players, we can determine whether a player can win at the current position by traversing the game tree. Ambainis and Kokainis [24] proposed a quantum algorithm that can evaluate a 2-player game quadratically faster than the classical deterministic algorithm. However, deterministic search does not apply to large-scale problems such as go. For practical game AI, instead of determining the winning strategy, the algorithm only needs to provide a good enough choice. Hence the heuristic method is commonly used so that we have some efficient ways to evaluate the current position and guess the best choice. We will show a nested framework of searching game trees heuristically by calling our algorithm.

We formalize the heuristic method as a weighted search tree (see Figure 5 as an example), such as the Monte Carlo tree after training, whose edge weights suggest the probability of each choice to be good. The classical search strategy at step t is formed as a probability distribution $\mathbf{g}_t(s_1 \cdots s_t)$, where $s_1 \cdots s_t$ is the choice history of players and $\mathbf{g}_t(s_1 \cdots s_t)_i$ is the probability that choice i is good at this position. These strategies can be obtained from domain expert knowledge or by machine learning. Taking the cheese game as an example, when $s_1 \cdots s_t$ represents the first t placement positions that two players alternately played, $\mathbf{g}_t(s_1 \cdots s_t)_i$ is the probability that i might be the best placement. Actually, $\mathbf{g}_t(s_1 \cdots s_t)$ can be seen as an approximated probability distribution of the real optimal solution. When the number of queries is limited to T , the top T nodes with the highest probability are evaluated in the classical algorithm. Directly calling Grover’s algorithm after ranking the probabilities implies a quadratic speed-up over classical. Our algorithm is better than directly calling Grover’s algorithm after ranking, as shown in Figure 2. Since the strategy \mathbf{g}_t can be calculated efficiently by a classical computer, the quantum operator that generates state $\sum_i \sqrt{\mathbf{q}^*(\mathbf{g}(s_1 \cdots s_t))_i} |i\rangle$ can be implemented efficiently, as shown in Figure 6.

Consider the case when Alice and Bob are playing a cheese game. Note that determining whether Alice can win by taking choice i can be implemented by determining whether Bob can win after Alice takes choice i . Hence the “Oracle” determining whether a choice is good can be implemented by calling our search algorithm instead of using the heuristic function directly to improve the estimation accuracy. We can call our algorithm recursively to notea certain depth in the game tree and then use “approximated Oracle” such as a neural network or rule-based evaluation, just like the classic game tree search in practice. This leads to a quantum framework similar to the classical game tree search, which can speed up the game tree search. Our algorithm is optimal at each level of the tree locally, we believe it would perform well globally as well.

5 Conclusion

We consider how to take advantage of contextual information in quantum computation. We provide an optimal quantum search algorithm that achieves the maximum expected success probability with a given number of queries when the probability distribution of the solution is given. In particular, the quantum advantage of our algorithm increases as the distribution of the solution becomes more biased. We also provide a framework showing how to apply our quantum algorithm to the heuristic search algorithm for the game tree. Since a significant advantage of our algorithm has been shown by implementation on state-of-the-art devices with a very limited number of items, more practical problems of larger size can be expected to perform much better in the future.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 62325210, 62272441) and Strategic Priority Research Program of Chinese Academy of Sciences (Grant No. XDB28000000).

References

- 1 Wiener M J. Efficient DES Key Search. Technical Report TR-244, 1994
- 2 Daemen J, Rijmen V. AES proposal: rijndael. 1999. <https://www.cs.cmu.edu/afs/cs/project/pscico-guyb/realworld/www/docs/rijndael.pdf>
- 3 Coulom R. Efficient selectivity and backup operators in Monte-Carlo tree search. In: Proceedings of the International Conference on Computers and Games, 2006. 72–83
- 4 Bennett C H, Bernstein E, Brassard G, et al. Strengths and weaknesses of quantum computing. *SIAM J Comput*, 1997, 26: 1510–1523
- 5 Grover L K. Quantum mechanics helps in searching for a needle in a haystack. *Phys Rev Lett*, 1997, 79: 325–328
- 6 Grover L K. Quantum computers can search rapidly by using almost any transformation. *Phys Rev Lett*, 1998, 80: 4329–4332
- 7 Brassard G, Hoyer P, Mosca M, et al. Quantum amplitude amplification and estimation. 2002. ArXiv:quant-ph/0005055
- 8 Zalka C. Grover's quantum searching algorithm is optimal. *Phys Rev A*, 1999, 60: 2746–2751
- 9 Grover L K. Fixed-point quantum search. *Phys Rev Lett*, 2005, 95: 150501
- 10 Mizel A. Critically damped quantum search. *Phys Rev Lett*, 2009, 102: 150501
- 11 Yoder T J, Low G H, Chuang I L. Fixed-point quantum search with an optimal number of queries. *Phys Rev Lett*, 2014, 113: 210501
- 12 He X Y, Sun X M, Yang G, et al. Exact quantum query complexity of weight decision problems via Chebyshev polynomials. *Sci China Inf Sci*, 2023, 66: 129503
- 13 Silver D, Schrittwieser J, Simonyan K, et al. Mastering the game of go without human knowledge. *Nature*, 2017, 550: 354–359
- 14 Silver D, Hubert T, Schrittwieser J, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 2018, 362: 1140–1144
- 15 Wang Y L, Li G X, Wang X. A hybrid quantum-classical Hamiltonian learning algorithm. *Sci China Inf Sci*, 2023, 66: 129502
- 16 Montanaro A. Quantum search with advice. In: Proceedings of the Conference on Quantum Computation, Communication, and Cryptography, 2010. 77–93
- 17 Figgatt C, Maslov D, Landsman K A, et al. Complete 3-qubit grover search on a programmable quantum computer. *Nat Commun*, 2017, 8: 1–9
- 18 Zheng Q L, Zhu P Y, Xue S C, et al. Quantum algorithm and experimental demonstration for the subset sum problem. *Sci China Inf Sci*, 2022, 65: 182501
- 19 Sun X, Tian G, Yang S, et al. Asymptotically optimal circuit depth for quantum state preparation and general unitary synthesis. *IEEE Trans Comput-Aided Des Integr Circ Syst*, 2023, 42: 3301–3314
- 20 Cross A. The IBM Q experience and QISKit open-source quantum computing software. *Bull Am Phys Soc*, 2018, 2018: 63
- 21 Nielsen M A, Chuang I. *Quantum Computation and Quantum Information*. Cambridge: Cambridge University Press, 2002
- 22 Rosin C D. Multi-armed bandits with episode context. *Ann Math Artif Intell*, 2011, 61: 203–230
- 23 Wang D, You X, Li T, et al. Quantum exploration algorithms for multi-armed bandits. 2020. ArXiv:2007.07049
- 24 Ambainis A, Kokainis M. Quantum algorithm for tree size estimation, with applications to backtracking and 2-player games. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, 2017. 989–1002

Appendix A Proof of the optimality

Theorem A1 (Upper bound). Given a distribution $\mathbf{p} = (p_1, p_2, \dots, p_N)$ indicating where the solution might be, the expected success probability of the quantum algorithm in T Oracle queries is upper bounded by

$$\max_{\mathbf{r}} \sum_{i=1}^N r_i \sin^2((2T+1) \arcsin \sqrt{r_i}) \quad (\text{A1})$$

$$\text{s.t. } \sum_{i=1}^N r_i \leq 1, \quad (\text{A2})$$

$$\forall i: 0 \leq r_i. \quad (\text{A3})$$

Proof. Any quantum algorithm in T queries is formed as a transformation $U_T O_x U_{T-1} \cdots U_1 O_x U_0$ applied to $|0\rangle$ and measurement $\{M_i\}$, $i \in \{1, 2, \dots, N\}$.

Let $\phi_t^x = U_T O_x U_{T-1} \cdots O_x U_t U_{t-1} \cdots U_0 |0\rangle$. Let

$$f(x) := \begin{cases} \sin x, & 0 \leq x \leq \pi/2, \\ 1, & \pi/2 < x. \end{cases} \quad (\text{A4})$$

For unit vectors $|a\rangle, |b\rangle, |c\rangle$, let $\langle a, c \rangle = 2 \arcsin(|a - c|)$ denotes the angle between a and c , then $\langle a - c | a - c \rangle = 4 \sin^2(\langle a, c \rangle / 2)$. Since $\langle a, c \rangle \leq \langle a, b \rangle + \langle b, c \rangle$ by triangle inequality,

$$|a - c| \tag{A5}$$

$$= 2 \sin(\langle a, c \rangle / 2) \tag{A6}$$

$$\leq 2 \sin(\min(\langle a, b \rangle / 2 + \langle b, c \rangle / 2, \pi / 2)) \tag{A7}$$

$$= 2f(\langle a, b \rangle / 2 + \langle b, c \rangle / 2). \tag{A8}$$

Quantum states are unit vectors in Hilbert space. Note that ϕ_t^x is unit vector for all t and $\phi_0^x = (\phi_0^x - \phi_1^x) + (\phi_1^x - \phi_2^x) + \dots + (\phi_{T-1}^x - \phi_T^x) + \phi_T^x$, M_x is the measurement which outputs x . We have

$$\cos(\langle \phi_0^x, \phi_T^x \rangle) = |\langle \phi_0^x | \phi_T^x \rangle| \tag{A9}$$

$$= \langle \phi_0^x | M_x M_x | \phi_T^x \rangle + \langle \phi_0^x | (I - M_x)(I - M_x) | \phi_T^x \rangle \tag{A10}$$

$$\leq |M_x | \phi_0^x \rangle| |M_x | \phi_T^x \rangle| + \sqrt{1 - |M_x | \phi_0^x \rangle|^2} \sqrt{|M_x | \phi_T^x \rangle|^2} \tag{A11}$$

$$= \cos(\arcsin(|M_x \phi_0^x|) - \arcsin(|M_x \phi_T^x|)). \tag{A12}$$

Then

$$\arcsin(|M_x \phi_0^x|) - \arcsin(|M_x \phi_T^x|) \tag{A13}$$

$$\leq \langle \phi_0^x, \phi_T^x \rangle \tag{A14}$$

$$\leq \langle \phi_0^x, \phi_1^x \rangle + \langle \phi_1^x, \phi_2^x \rangle + \dots + \langle \phi_{T-1}^x, \phi_T^x \rangle \tag{A15}$$

$$= 2 \arcsin(|\phi_0^x - \phi_1^x|/2) + \dots + 2 \arcsin(|\phi_{T-1}^x - \phi_T^x|/2). \tag{A16}$$

Hence the the probability of outputting x with input solution x is

$$|M_x \phi_0^x|^2 \tag{A17}$$

$$\leq f^2(\arcsin(|M_x \phi_T^x|) + \sum_{t=1}^T 2 \arcsin(|\phi_{t-1}^x - \phi_t^x|/2)). \tag{A18}$$

On the other hand

$$\langle \phi_t^x - \phi_{t+1}^x | \phi_t^x - \phi_{t+1}^x \rangle \tag{A19}$$

$$= \langle 0 | U_0^\dagger \dots U_t^\dagger (O_x - I) U_{t+1}^\dagger O_x \dots O_x U_T^\dagger U_T O_x U_{T-1} \dots O_x U_{t+1} (O_x - I) U_t \dots U_0 | 0 \rangle \tag{A20}$$

$$= \langle 0 | U_0^\dagger \dots U_t^\dagger (2|x\rangle\langle x|)(2|x\rangle\langle x|) U_t \dots U_0 | 0 \rangle \tag{A21}$$

$$= 4 |\langle x | U_t \dots U_0 | 0 \rangle|^2. \tag{A22}$$

Let $u_{t,x} = |\langle x | U_t \dots U_0 | 0 \rangle|^2$, we have

$$|M_x \phi_0^x|^2 \tag{A23}$$

$$\leq f^2 \left(\arcsin(|M_x \phi_T^x|) + \sum_{t=1}^T 2 \arcsin(|\phi_{t-1}^x - \phi_t^x|/2) \right) \tag{A24}$$

$$= f^2 \left(\arcsin \sqrt{u_{T,x}} + 2 \sum_{t=0}^{T-1} \arcsin \sqrt{u_{0,x}} \right). \tag{A25}$$

So the expected success probability is upper bounded by

$$\max_{\{u_{t,x}\}} \sum_{x=1}^N p_x f^2 \left(\arcsin \sqrt{u_{T,x}} + 2 \sum_{t=0}^{T-1} \arcsin \sqrt{u_{0,x}} \right) \tag{A26}$$

$$\text{s.t. } \forall t : \sum_{x=1}^N u_{t,x} \leq 1, \tag{A27}$$

$$\forall t, x : u_{t,x} \geq 0. \tag{A28}$$

The optimization function (A26) increases monotonously with $u_{t,x}$, so we can replace the ≤ 1 with $= 1$ in constraints (A27) without changing the maximum value. By Lemma A1, the function can reach its maximum when $u_{t,x} = u_{0,x}$, which matches Theorem A1.

Lemma A1. The following optimization problem:

$$\max_{\{u_{t,x}\}} \sum_{x=1}^N p_x f^2 \left(\sum_{t=1}^m \arcsin \sqrt{u_{t,x}} \right) \tag{A29}$$

$$\text{s.t. } \forall t : \sum_{x=1}^N u_{t,x} = 1, \tag{A30}$$

$$\forall t, x : u_{t,x} \geq 0 \tag{A31}$$

can reach the maximum when $\forall x, t : u_{t,x} = u_{0,x}$.

Proof. Let the function reach its maximum value with the sum of the variance

$$\sum_{t=1}^m \sum_{x=1}^N \left(u_{t,x} - \sum_{x=1}^N u_{t,x}/n \right)^2, \quad (\text{A32})$$

minimized at the same time. If there exists x, t that $u_{t,x} \neq u_{0,x}$, assuming $u_{0,1} < u_{1,1}, u_{0,2} > u_{1,2}$ without loss of generality.

First, if $u_{0,1} + u_{1,1} \geq 1$, we can update $u_{0,1}$ and $u_{1,1}$ to $u_{0,1} + \delta$ and $u_{1,1} - \delta$ without decreasing the maximum value when $\delta \leq |u_{0,1} - u_{1,1}|/2$, since $\arcsin \sqrt{u_{0,1} + \delta} + \arcsin \sqrt{u_{1,1} - \delta} \geq \pi/2$.

Second, if $u_{0,1} + u_{1,1} < 1$, we have differentials

$$d \arcsin \sqrt{u_{0,1}} = 1/\sqrt{(1-u_{0,1})u_{0,1}} du_{0,1}, \quad (\text{A33})$$

$$d \arcsin \sqrt{u_{1,1}} = 1/\sqrt{(1-u_{1,1})u_{1,1}} du_{1,1}, \quad (\text{A34})$$

and

$$1/\sqrt{(1-u_{0,1})u_{0,1}} - 1/\sqrt{(1-u_{1,1})u_{1,1}} > 0, \quad (\text{A35})$$

when $u_{0,1} + u_{1,1} < 1, u_{0,1} < u_{1,1}$. So when $\delta \leq |u_{0,1} - u_{1,1}|/2$

$$\arcsin \sqrt{u_{0,1} + \delta} + \arcsin \sqrt{u_{1,1} - \delta} \quad (\text{A36})$$

$$\geq \arcsin \sqrt{u_{0,1}} + \arcsin \sqrt{u_{1,1}}. \quad (\text{A37})$$

Similarly, when $\delta \leq |u_{0,2} - u_{1,2}|/2$

$$\arcsin \sqrt{u_{0,2} - \delta} + \arcsin \sqrt{u_{1,2} + \delta} \quad (\text{A38})$$

$$\geq \arcsin \sqrt{u_{0,2}} + \arcsin \sqrt{u_{1,2}}. \quad (\text{A39})$$

So we can choose suitable δ to update $u_{0,1}$ and $u_{1,1}$ or $u_{0,2}$ and $u_{1,2}$ to their average without decreasing the maximum value.

Then the maximum can be obtained with a smaller summation of variance in (A32), which contradicts the assumption!

Hence the maximum value of the optimization function can be obtained when $\forall x, t : u_{t,x} = u_{0,x}$.

Appendix B How to calculate parameters q^*

Here we show how to find the optimal parameters $q^* = (q_1^*, \dots, q_N^*)$ by Lagrange multiplier method when $T = 1$.

Note that $\sin^2((2T+1)\arcsin\sqrt{q_i})$ is a polynomial of degree $2T+1$ with respect to q_i . When $T = 1$, the optimization function is $\sum_{i=1}^N p_i \sin^2(3\arcsin\sqrt{q_i}) = \sum_{i=1}^N p_i q_i (3-4q_i)^2$. Note that when $n \leq 4$, the success probability can reach 1. When $n > 4$, Eq. (2) always reaches its maximum value when $\sum_{i=1}^N q_i = 1$ since the function increases monotonously with increase of q_i . The Lagrangian function

$$\mathcal{L}(\mathbf{q}, \lambda) = \sum_{i=1}^N p_i q_i (3-4q_i)^2 - \lambda \left(1 - \sum_{i=1}^N q_i \right), \quad (\text{B1})$$

By $\nabla \mathcal{L}(\mathbf{q}, \lambda) = \mathbf{0}$, we get

$$\begin{cases} p_i (48q_i^2 - 48q_i + 9) + \lambda = 0, \\ \sum_{i=1}^N q_i = 1, \end{cases} \quad (\text{B2})$$

which can be simplified to

$$\sum_{i=1}^N \frac{1}{2} - \sqrt{\frac{1}{16} - \frac{\lambda}{48p_i}} = 1, \quad (\text{B3})$$

and

$$q_i = \frac{1}{2} - \sqrt{\frac{1}{16} - \frac{\lambda}{48p_i}}, \quad (\text{B4})$$

Eq. (B3) can be solved efficiently by binary search of λ since the left side is a monotone function of x .