

Depth-First Grover Search Algorithm on Hybrid Quantum-Classical Computer

Haoxiang Guo
yurodidon@gmail.com

Abstract

We demonstrated the detailed construction of the hybrid quantum-classical computer. Based on this architecture, the useful concept of amplitude interception is illustrated. It is then embedded into a combination of Depth-First Search and Grover’s algorithm to generate a novel approach, the Depth-First Grover Search(DFGS), to handle multi-solution searching problems on unstructured databases with an unknown number of solutions. Our new algorithm attains an average complexity of $\mathcal{O}(m\sqrt{N})$ which performs as efficient as a normal Grover Search, and a $\mathcal{O}(\sqrt{p}N)$ complexity with a manually determined constant p for the case with all elements are solutions, where a normal Grover Search will degenerate to $\mathcal{O}(N\sqrt{N})$. The DFGS algorithm is more robust and stable in comparison.

Keywords— Quantum Computation, Hybrid Quantum-Classical Computer, Grover’s Algorithm, Unstructured Database Search

1 Introduction

Since the concept of Quantum Computing was proposed[Ben80; Fey85], bunches of quantum algorithms were designed to achieve quantum supremacy[Pre12]. And these algorithms can be separated into two large groups: **(1)** Algorithms based on Quantum Fourier Transformation[Cop02; Sho94; Sho97] and **(2)** Algorithms based on oracles[DJ92; BV97; Gro96], in which Grover’s algorithm is able to solve the single-solution unstructured searching problems with a complexity of $\mathcal{O}(\sqrt{N})$, and was demonstrated to be asymptotic optimal in [Ben+97].

In recent years, a new concept of Hybrid Quantum-Classical(HQC) Computation was presented[Llo00] and received rising degrees of attention, the concept of HQC is applied to several areas in computer science[End+21; Ott+17; Liu+21; Ber+18]. By attaching quantum components with a classical computer, two parts complement each other, leading HQC to have both merits of them, e.g., quantum parallelism[NC10], data storing, and efficient arithmetic operations.

Whereas a few articles talk about the detailed structure of HQC, in this paper, we devoted Section 2 to investigating the configuration of HQC. Furthermore, we confronted the issue of the inefficiency of Grover’s algorithm when applying it to multi-solution search problems (which will encounter repetitions and worsen up to $\mathcal{O}(N\sqrt{N})$), and proposed the concept of amplitude interception and a novel robust algorithm, the Depth-first Grover Search based on an HQC computer in Section 3.

2 Hybrid Quantum-Classical Structure

We aim to design an architecture for HQC with two basic functions:

- The ability to operate quantum parts and classical parts individually.
- The ability to intercommunicate between the quantum part and the classical part.

Both of these functions can be implemented by two components: `INITIALIZER` and `ENCODER`. `INITIALIZER` component initializes auxiliary qubits for follow-up quantum computing, e.g.,

for handling searching problems, it is usually composed with HX gates to get $|-\rangle$ for phase kickback[Cle+98]. Besides the INITIALIZER, the ENCODER component enables the information transformation from classical to quantum parts. An example is shown in Figure 1, which directly copies the digits to qubits.

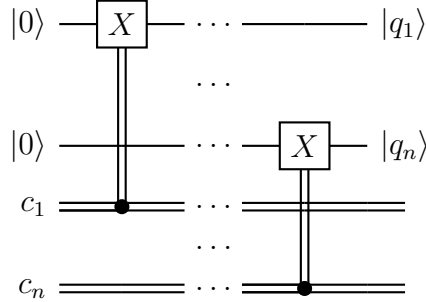


Figure 1: An example of ENCODER, composed with n control-not gate

The bridge from the classical part to the quantum part is also proven to have various forms. In [Suc+18], the quantum part is considered to be arranged in the cloud and intercommunicated with Internet technology. Recently, IBM constructed 127-qubits quantum computers with remote accessibility[IBM21], which implies that future quantum computing will have a trend of online-offline separation. The general architecture of HQC is demonstrated in Figure 2.

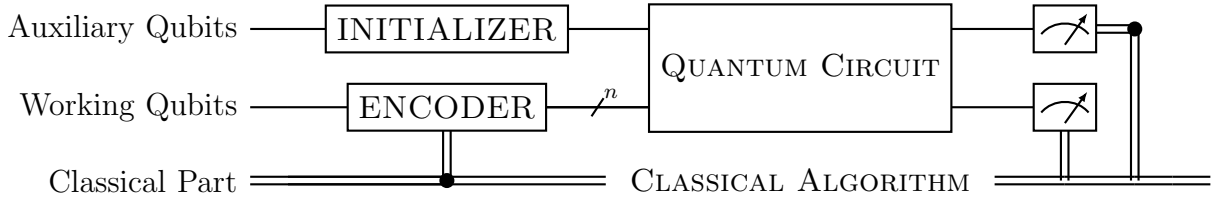


Figure 2: The detailed structure of the hybrid quantum-classical computer. In many cases, the auxiliary qubits are not necessary to be measured.

A crucial feature of this architecture is it allows the implementation of discrete loops[Bli94], we refer to discrete since we always destroy superpositions after each measurement. Since an amount of information is inevitably lost during the measurements, so there exists an irremovable gap between the conditional loops we commonly regard and the discrete loops we have. Or in other words, we are still unable to implement conditional loops upon this architecture.

In HQC, classical parts can either work as the main parts for performing algorithms or auxiliary parts for data storing and updating. A sort of classical algorithm can be improved by replacing some steps(e.g., extreme searching[DH96; AK99; KOW+08]) with quantum algorithms to obtain better performance[Hei03; AS05]. Furthermore, some other authors proposed variational hybrid quantum-classical algorithms[McC+16; Ple+22] by regarding classical computers as an auxiliary part. As well as in [BSS16] the authors discovered a portion of qubits required in quantum algorithms could be separated and replaced by classical bits with the same efficiency but save quantum resources.

3 Improvement of Grover’s Algorithm

We aim to design a novel algorithm, with classical and quantum cooperation, to solve the multi-solution searching problems with an unknown number of solutions based on hybrid quantum-classical architecture. The algorithm is regarded to be a combination of the Partial Grover Search[GR04; KG05] and the Depth-First Search[Tar71]. Besides this, we first proposed a method to deal with the difficulty of applying Grover Search to multi-solution problems.

3.1 Amplitude Interception

The general Grover search is not suitable for multi-solution searching problems[NC10]. Some improvements were presented in [Boy+96; Buh+99], however, there still exist the following flaws: **(1)** One will be possible to get repetitions, **(2)** For an unknown number of solutions the program has no criterion for terminating. In this section, analogous to amplitude amplification[Bra+02], we introduce the concept of amplitude interception, which can solve both of the issues stated above.

The general idea is to add a new gate to the head of each Grover’s operations, where the gate U_I is defined as follows

$$U_I |x\rangle |y\rangle = \begin{cases} |x\rangle |y \oplus 1\rangle & \text{for } x \in S \\ |x\rangle |y\rangle & \text{for } x \notin S \end{cases} \tag{1}$$

Where S is the set of found solutions. In general, the ket $|y\rangle$ is initialized to $|-\rangle$ for phase kickback. Thus, applying U_I gates has the effect of flipping the phase of founded elements. The new quantum circuit is shown in Figure 3.

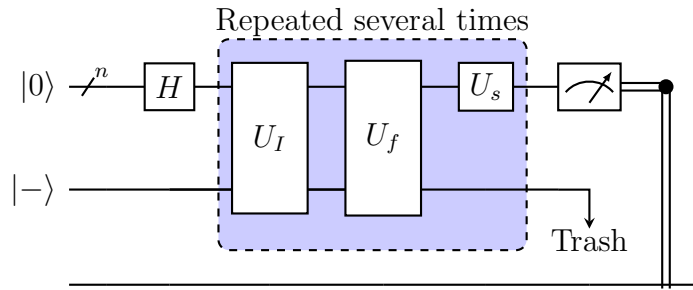


Figure 3: The modified structure of the Grover algorithm, with an additional gate added in front

In the circuit, U_f and U_s are still the oracle and diffuser[Gro96]. The combination $U_I \otimes U_f$ will flip and only flip the undiscovered solutions, so that found solutions will be regarded as the same as other trivial elements, and contributes amplitudes to the real solutions.

Therefore, if we try to apply this amplitude interception version of Grover’s search several times to find all solutions, we cross out the founded solutions before each trial and avoid repetitions. At the same time, with attachment to a classical computer, we can terminate the program immediately after getting wrong results at certain times, where the classical computer plays the role of register and validator.

3.2 Depth-First Grover Search

Suppose now we are doing searches on a database constructed by $N = 2^n$ elements, with an unknown number of solutions. Taking the method above to do multi-solution search problems gives an average complexity of $\mathcal{O}(m\sqrt{N})$ and is expected to degenerate to $\mathcal{O}(N\sqrt{N})$ for the extreme case with all elements are solutions, which is even less efficient than a classical brute-force $\mathcal{O}(N)$ search. Therefore, the method aforementioned is still not optimal.

In this section, we merged Partial Grover Search with Depth-First Search(DFS) and presented the Depth-First Grover Search(DFGS) with the help of amplitude interception to solve the multi-solution searching problems. The outline of DFGS is illustrated in Figure 4.

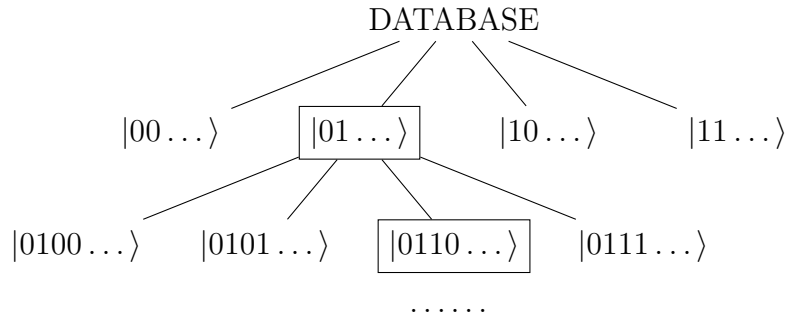


Figure 4: The steps of DFGS with $p = 4$. $p = 4$ means to split the database into 4 pieces for each layer, or in other words, to determine 2 bits of solution on each layer. Where the general idea is depth-first search, but the task of determining the next 2 bits of the solution address is accomplished by a Partial Grover Search

To be specific, we assume one has the resources to do (partial) Grover searches with high precision for p blocks. At each layer, we used Partial Grover Search to determine which intervals among p equal-size splitted sub-intervals contain solutions, and with DFS to recurse through every potential solution interval. Finally, for intervals with sizes less or equal to p , we adopt a normal Grover Search to find the exact address of the solution.

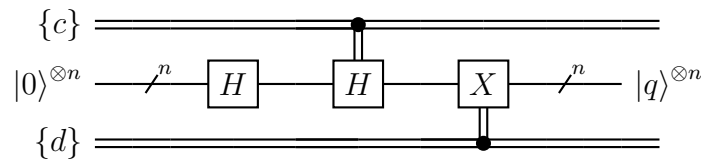


Figure 5: The modified ENCODER component, with two classical arrays. The classical array labeled c plays the role of recording current bits and copying them to qubits when encoding. Another classical array labeled d stores Bool values to cache if corresponding bits are determined. If one is determined, then the Hardmard gate is activated to convert the corresponding qubits from EPR superposition to $|0\rangle$ or $|1\rangle$. This ENCODER is applied to both the normal Grover Search and the partial one.

The amplitude interception is applied to Grover and Partial Grover searches. In the original paper on Partial Grover Search, the algorithm was designed to find the first k bits of the solution index. In our algorithm, we expect the Partial Grover Search to find the *next* $\ell = \log_2 p$ bits. Or

in other words, the action we take is

$$| \underbrace{\dots\dots}_{\text{Determined next } \ell} \underbrace{???}_{\dots} \dots \rangle \xrightarrow{\text{Partial Grover Search}} | \underbrace{\dots\dots}_{\text{Determined next } \ell} \underbrace{abc}_{\dots} \dots \rangle \quad (2)$$

This can be implemented by modifying the ENCODER of the circuit, as shown in Figure 5. We put two auxiliary classical arrays for storing (1) the determined value of bits, (2) the state of the bits(Checked and Unchecked). With this ENCODER, the qubits will be initialized into a superposition of

$$| \underbrace{\dots\dots}_{\text{Determined r bits}} \rangle \otimes \frac{1}{\sqrt{2^{n-r}}} \bigoplus_{i=0}^{n-r} (|0\rangle + |1\rangle) \quad (3)$$

To clarify the process of ENCODER, consider the example given in Figure 4. On the second layer, the first 4 bits were decided, therefore two classical arrays $\{c\}$ and $\{d\}$ will be like $\{c\} = \{0, 1, 1, 0, \dots\}$ and $\{d\} = \{1, 1, 1, 1, \dots\}$, represents the first four bits are determined to be 0110. Under these conditions, ENCODER will give

$$|0110\rangle \otimes \frac{1}{\sqrt{2^{n-4}}} \bigoplus_{i=0}^{n-4} (|0\rangle + |1\rangle) \quad (4)$$

The determined bits are fixed, and the Partial Grover Search will only find the next k bits, which is our goal. So now we have everything we need to write down DFGS, as follows, implemented by recursions[Cor+09].

Algorithm Depth-First Grover Search(DFGS)

```

1:  $c \leftarrow \{0\}_n, d \leftarrow \{0\}_n$ 
2: procedure DFGS( $r, \text{ans}$ )
3:    $S \leftarrow \emptyset, \text{count} \leftarrow 0$ 
4:   Assign values to  $c$  and  $d$  for ENCODER
5:   if  $n - r \leq \ell$  then
6:     address  $\leftarrow$  Grover Search ▷ Normal Grover search to find solutions
7:     ans  $\leftarrow$  ans  $\ll \ell$  | address ▷ Bitwise operations
8:     return  $\{x \mid x \in \text{ans}, f(x) = 1\}$ 
9:   while count  $< p$  do
10:    address  $\leftarrow$  Partial Grover Search ▷ Most possible sub-interval
11:    ans'  $\leftarrow$  ans  $\ll \ell$  | address
12:    Ret  $\leftarrow$  DFGS( $r + \ell, \text{ans}'$ ) ▷ Recursion
13:    if Ret =  $\emptyset$  then ▷ The most possible interval gives no solution
14:      break
15:     $S \leftarrow S \cup \text{Ret}, \text{count} \leftarrow \text{count} + 1$ 
16:   Recover elements in  $c$  and  $d$ 
17:   return  $S$ 
18: end procedure

```

Where bitwise operations were used for simplicity. In line 6, we expect the performance of Grover Search will find all possible solutions in the given interval, so the resultant addresses and

therefore `ans` are sets. In line 8 we use the classical part to do further verification and abandon all incorrect solutions in the answer set (for $f(x) \neq 1$ is an incorrect solution).

With recursion, we anticipate DFGS will have the ability to find all solutions. After looking up one interval, the algorithm will backtrack to the previous layer and continue recursing into the next possible interval (line 12) until the most-possible intervals give no solutions (line 13).

3.3 Analysis on Performance

3.3.1 Time complexity

Some rough analyses tell the complexity of the DFGS algorithm will have an average complexity of $\mathcal{O}(m\sqrt{N})$, and $\mathcal{O}(\sqrt{p}N)$ for every element is a solution.

For each recursion, the program performs either partial Grover searches or normal Grover searches (for intervals with a size less or equal to p). According to [GR04], the total queries required for one partial Grover Search is $\sqrt{L} - \sqrt{L/p}$, where L refers the length of searching interval for recursion depth k , the number of queries is $\sqrt{N/p^k} - \sqrt{N/p^{k+1}}$. The maximum depth of recursion, λ , can be solved out by

$$\frac{N}{p^\lambda} = p \rightarrow \lambda = \frac{\log N}{\log p} - 1 \quad (5)$$

We suppose at depth k there are α_k partial searches performed ($1 \leq \alpha_k \leq p$), then the complexity of DFGS will be

$$\sum_{k=0}^{\lambda-1} \alpha_k \left(\sqrt{\frac{N}{p^k}} - \sqrt{\frac{N}{p^{k+1}}} \right) + \alpha_{\lambda+1} \sqrt{\frac{N}{p^\lambda}} \quad (6)$$

α_k represents the expected number of Partial Grover Searches performed on layer k . For the case with $\alpha_k \equiv 1$, corresponds to a situation with only one solution, DFGS has a strict complexity of $\mathcal{O}(\sqrt{N})$, and for the case with all elements are solutions, $\alpha_k \equiv p^k$, DFGS has an asymptotic complexity of $\mathcal{O}(\sqrt{p}N)$.

To figure out the average complexity, α_k should be numerically equal to the expected number of intervals that involve solutions. We define

$$x_i = \begin{cases} 1 & \text{for interval with no solutions} \\ 0 & \text{for interval with any number of solutions} \end{cases}$$

Where, the index i refers to the i th block on current layer k (where there are p^k blocks in total). Therefore $X = \sum x_i$ is the number of blocks with no solutions. We have the relationship

$$E(X) = E\left(\sum_{i=0}^{p^k} x_i\right) = \sum_{i=0}^{p^k} E(x_i) \quad (7)$$

$E(x_i)$ is relatively easier to figure out, the result is $E(x_i) = (1 - p^{-k})^m$, and so $E(X) = p^k(1 - p^{-k})^m$, then the expectation number of intervals with solutions, $\alpha_k = p^k(1 - (1 - p^{-k})^m)$.

So the summation can be written as

$$\sum_{k=0}^{\lambda-1} p^k (1 - (1 - p^{-k})^m) \left(\sqrt{\frac{N}{p^k}} - \sqrt{\frac{N}{p^{k+1}}} \right) + m \sqrt{\frac{N}{p^\lambda}} \quad (8)$$

Apply Bernoulli's inequality $(1 + x)^r \geq 1 + rx$, we have $(1 - (1 - p^{-k})^m) \leq mp^{-k}$

$$\text{Equation (8)} \leq \sum_{k=0}^{\lambda-1} m \left(\sqrt{\frac{N}{p^k}} - \sqrt{\frac{N}{p^{k+1}}} \right) + m \sqrt{\frac{N}{p^\lambda}} \quad (9)$$

After finishing this summation, we conclude that DFGS has an average complexity of $\mathcal{O}(m\sqrt{N})$. In comparison to a normal Grover Search, the Depth-First Grover Search(DFGS) attains the same average complexity but a better performance for problems with very densely-distributed solutions (it will not worsen to $\mathcal{O}(N\sqrt{N})$). Or in other words, DFGS is more stable for unstructured multi-solution problems.

The remaining question is the determination value of p . Qualitative analysis will suggest that 2 is a reasonable choice of p , the smallest integer greater than 1 (which corresponds to a classical brute-force search, with complexity $\mathcal{O}(N\sqrt{N})$), or in other words to find a single bit on each layer. The selection of $p = 2$ will work outstandingly for an evenly-distributed database (including the case where all elements are solution). On the other hand, for the remarkable case of a densely-distributed database, a larger searching block (i.e., a greater p) is expected to be more efficient in determining intervals of solutions and catching all of them in one draft.

In general, the value of p will closely rely on the predicted configuration of the database and the quantum resources available, so in this paper, we will not give a specific upper bound for the value p but keep it as a manually determined parameter.

3.3.2 Correctness

Complexity aside, the algorithm has an unavoidable possibility of not managing to figure out all solutions. In an extreme case, the program may go astray on the first layer and directly cause the whole algorithm to end up with no solutions. One method to eliminate this risk is to change the criteria of terminating, to make the program terminate until it finds incorrect solutions(or no solutions) for q_k times on layer k . Since for deeper layers, the number of elements is decreasing, q_k is expected to be inversely proportional to k , as follows

$$q_k = \left\lceil \frac{\nu}{k+1} \right\rceil, \quad \alpha'_k = \alpha_k + q_k$$

Again, ν is still a manually determined value but with an upper bound p . The value of it depends on the goal accuracy and the number of elements. The corresponding average complexity for DFGS with new q_k becomes $\mathcal{O}((m + \nu)\sqrt{N})$. Since the coefficients of summation have the following relationship

$$\alpha_k + \left\lceil \frac{\nu}{k+1} \right\rceil \leq \alpha_k + \nu$$

After implementing this new criterion, the algorithm will then have greater opportunities to sort out all solutions.

4 Conclusion

In this paper, we proposed a quantum advantage algorithm DFGS, with quantum-classical computer hybridizing to do search problems for an unknown number of solutions. With the help of amplitude interception, our novel algorithm has the ability to figure out all solutions and attains the same efficiency as the normal Grover search but a more robust and stable performance for extreme cases.

On the implementation level, the main body of DFGS takes an equal number of Grover's diffusers and measurements. An extra ENCODER is required for intercommunication between the classical and quantum parts. For the ENCODER, besides Hardmard gates, several gates controlled by the classical part need to be arranged. Moreover, the new gate U_I added for amplitude interception can be absorbed by adjusting the U_f gate.

Based on the current trends of quantum computation, more and more quantum algorithms will rely on the hybrid quantum-classical computer. On the other hand, when designing classical algorithms, one can also consider embedding quantum algorithms for optimal performance. For example, a heap structure is difficult on determining the exact index of a specific element [FT87]. Applying a quantum search with complexity $\mathcal{O}(\sqrt{N})$ may be a solution.

In conclusion, Well-designed hybrid quantum-classical based algorithms or data structures can be *universal*, which means capable of handling any circumstances and requirements. The Depth-first Grover Search algorithm is illustrated to be *universal* since its capability for any configuration of search problems. Besides it, tons of hybrid algorithms with better performance are waiting to be discovered.

References

- [Tar71] Robert Tarjan. "Depth-first search and linear graph algorithms". In: *12th Annual Symposium on Switching and Automata Theory (swat 1971)*. 1971, pp. 114–121. DOI: 10.1109/SWAT.1971.10.
- [Ben80] Paul Benioff. "The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines". In: *Journal of Statistical Physics* 22.5 (May 1980), pp. 563–591. DOI: 10.1007/bf01011339. URL: <https://doi.org/10.1007/bf01011339>.
- [Fey85] Richard P. Feynman. "Quantum Mechanical Computers". In: *Optics News* 11.2 (Feb. 1985), p. 11. DOI: 10.1364/on.11.2.000011. URL: <https://doi.org/10.1364/on.11.2.000011>.
- [FT87] Michael L. Fredman and Robert Endre Tarjan. "Fibonacci heaps and their uses in improved network optimization algorithms". In: *Journal of the ACM* 34.3 (July 1987), pp. 596–615. DOI: 10.1145/28869.28874. URL: <https://doi.org/10.1145/28869.28874>.
- [DJ92] David Deutsch and Richard Jozsa. "Rapid solution of problems by quantum computation". In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439.1907 (Dec. 1992), pp. 553–558. DOI: 10.1098/rspa.1992.0167. URL: <https://doi.org/10.1098/rspa.1992.0167>.

-
- [Bli94] Johann Blieberger. “Discrete loops and worst case performance”. In: *Computer Languages* 20.3 (Aug. 1994), pp. 193–212. DOI: 10.1016/0096-0551(94)90004-3. URL: [https://doi.org/10.1016/0096-0551\(94\)90004-3](https://doi.org/10.1016/0096-0551(94)90004-3).
- [Sho94] P.W. Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. IEEE Comput. Soc. Press, 1994. DOI: 10.1109/sfcs.1994.365700. URL: <https://doi.org/10.1109/sfcs.1994.365700>.
- [Boy+96] Michel Boyer et al. “Tight bounds on quantum searching”. In: (1996). DOI: 10.1002/(SICI)1521-3978(199806)46:4/5<493::AID-PROP493>3.0.CO;2-P. eprint: [arXiv:quant-ph/9605034](https://arxiv.org/abs/quant-ph/9605034).
- [DH96] Christoph Durr and Peter Hoyer. *A Quantum Algorithm for Finding the Minimum*. 1996. eprint: [arXiv:quant-ph/9607014](https://arxiv.org/abs/quant-ph/9607014).
- [Gro96] Lov K. Grover. “A fast quantum mechanical algorithm for database search”. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC 96*. ACM Press, 1996. DOI: 10.1145/237814.237866. URL: <https://doi.org/10.1145/237814.237866>.
- [Ben+97] Charles H. Bennett et al. “Strengths and Weaknesses of Quantum Computing”. In: *SIAM Journal on Computing* 26.5 (Oct. 1997), pp. 1510–1523. DOI: 10.1137/s0097539796300933. URL: <https://doi.org/10.1137/s0097539796300933>.
- [BV97] Ethan Bernstein and Umesh Vazirani. “Quantum Complexity Theory”. In: *SIAM Journal on Computing* 26.5 (Oct. 1997), pp. 1411–1473. DOI: 10.1137/s0097539796300921. URL: <https://doi.org/10.1137/s0097539796300921>.
- [Sho97] Peter W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. In: *SIAM Journal on Computing* 26.5 (Oct. 1997), pp. 1484–1509. DOI: 10.1137/s0097539795293172. URL: <https://doi.org/10.1137/s0097539795293172>.
- [Cle+98] R. Cleve et al. “Quantum algorithms revisited”. In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 454.1969 (Jan. 1998), pp. 339–354. DOI: 10.1098/rspa.1998.0164. URL: <https://doi.org/10.1098/rspa.1998.0164>.
- [AK99] Ashish Ahuja and Sanjiv Kapoor. *A Quantum Algorithm for finding the Maximum*. 1999. DOI: 10.48550/ARXIV.QUANT-PH/9911082. URL: <https://arxiv.org/abs/quant-ph/9911082>.
- [Buh+99] H. Buhrman et al. *Bounds for Small-Error and Zero-Error Quantum Algorithms*. 1999. DOI: 10.48550/ARXIV.CS/9904019. URL: <https://arxiv.org/abs/cs/9904019>.
- [Llo00] Seth Lloyd. *Hybrid quantum computing*. 2000. DOI: 10.48550/ARXIV.QUANT-PH/0008057. URL: <https://arxiv.org/abs/quant-ph/0008057>.
- [Bra+02] Gilles Brassard et al. *Quantum amplitude amplification and estimation*. 2002. DOI: 10.1090/conm/305/05215. URL: <https://doi.org/10.1090/conm/305/05215>.
- [Cop02] D. Coppersmith. *An approximate Fourier transform useful in quantum factoring*. 2002. DOI: 10.48550/ARXIV.QUANT-PH/0201067. URL: <https://arxiv.org/abs/quant-ph/0201067>.

-
- [Hei03] Mark Heiligman. *Quantum Algorithms for Lowest Weight Paths and Spanning Trees in Complete Graphs*. 2003. eprint: [arXiv:quant-ph/0303131](https://arxiv.org/abs/quant-ph/0303131).
- [GR04] Lov K. Grover and Jaikumar Radhakrishnan. *Is partial quantum search of a database any easier?* 2004. eprint: [arXiv:quant-ph/0407122](https://arxiv.org/abs/quant-ph/0407122).
- [AS05] Andris Ambainis and Robert Spalek. *Quantum Algorithms for Matching and Network Flows*. 2005. eprint: [arXiv:quant-ph/0508205](https://arxiv.org/abs/quant-ph/0508205).
- [KG05] Vladimir E. Korepin and Lov K. Grover. “Simple Algorithm for Partial Quantum Search”. In: (2005). DOI: [10.1007/s11128-005-0004-z](https://doi.org/10.1007/s11128-005-0004-z). eprint: [arXiv:quant-ph/0504157](https://arxiv.org/abs/quant-ph/0504157).
- [KOW+08] LUIS ANTONIO BRASIL KOWADA et al. “A NEW QUANTUM ALGORITHM FOR SOLVING THE MINIMUM SEARCHING PROBLEM”. In: *International Journal of Quantum Information* 06.03 (June 2008), pp. 427–436. DOI: [10.1142/s021974990800361x](https://doi.org/10.1142/s021974990800361x). URL: <https://doi.org/10.1142/s021974990800361x>.
- [Cor+09] Thomas H Cormen et al. *Introduction to Algorithms*. 3rd ed. The MIT Press. London, England: MIT Press, July 2009.
- [NC10] Michael A Nielsen and Isaac L Chuang. *Quantum Computation and Quantum Information*. Cambridge, England: Cambridge University Press, Dec. 2010.
- [Pre12] John Preskill. *Quantum computing and the entanglement frontier*. 2012. DOI: [10.48550/ARXIV.1203.5813](https://arxiv.org/abs/1203.5813). URL: <https://arxiv.org/abs/1203.5813>.
- [BSS16] Sergey Bravyi, Graeme Smith, and John A. Smolin. “Trading Classical and Quantum Computational Resources”. In: *Physical Review X* 6.2 (June 2016). DOI: [10.1103/physrevx.6.021043](https://doi.org/10.1103/physrevx.6.021043). URL: <https://doi.org/10.1103/physrevx.6.021043>.
- [McC+16] Jarrod R McClean et al. “The theory of variational hybrid quantum-classical algorithms”. In: *New Journal of Physics* 18.2 (Feb. 2016), p. 023023. DOI: [10.1088/1367-2630/18/2/023023](https://doi.org/10.1088/1367-2630/18/2/023023). URL: <https://doi.org/10.1088/1367-2630/18/2/023023>.
- [Ott+17] J. S. Otterbach et al. *Unsupervised Machine Learning on a Hybrid Quantum Computer*. 2017. DOI: [10.48550/ARXIV.1712.05771](https://arxiv.org/abs/1712.05771). URL: <https://arxiv.org/abs/1712.05771>.
- [Ber+18] Ville Bergholm et al. *PennyLane: Automatic differentiation of hybrid quantum-classical computations*. 2018. DOI: [10.48550/ARXIV.1811.04968](https://arxiv.org/abs/1811.04968). URL: <https://arxiv.org/abs/1811.04968>.
- [Suc+18] Martin Suchara et al. “Hybrid quantum-classical computing architectures”. In: *Proceedings of the 3rd International Workshop on Post-Moore Era Supercomputing, 2018*. 2018.
- [End+21] Suguru Endo et al. “Hybrid Quantum-Classical Algorithms and Quantum Error Mitigation”. In: *Journal of the Physical Society of Japan* 90.3 (Mar. 2021), p. 032001. DOI: [10.7566/jpsj.90.032001](https://doi.org/10.7566/jpsj.90.032001). URL: <https://doi.org/10.7566/jpsj.90.032001>.
- [IBM21] IBM. *IBM quantum breaks 100-qubit barrier*. Nov. 2021. URL: <https://www.ibm.com/quantum-computing>.
- [Liu+21] Junhua Liu et al. “Hybrid quantum-classical convolutional neural networks”. In: *Science China Physics, Mechanics & Astronomy* 64.9 (Aug. 2021). DOI: [10.1007/s11433-021-1734-3](https://doi.org/10.1007/s11433-021-1734-3). URL: <https://doi.org/10.1007/s11433-021-1734-3>.
- [Ple+22] Kirill Plekhanov et al. “Variational quantum amplitude estimation”. In: *Quantum* 6 (Mar. 2022), p. 670. DOI: [10.22331/q-2022-03-17-670](https://doi.org/10.22331/q-2022-03-17-670). URL: <https://doi.org/10.22331/q-2022-03-17-670>.