Survey Article

Multi-Solution Search and Solution-Count Estimation with Grover's Algorithm: A Survey

Yuxuan Wang[1]，Jiakai Sun[2,*]

[1] School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, P.R.China
[2] College of Computer Science, Sichuan University, Chengdu 610065, P.R.China
Corresponding author Email: sunjiakai@scu.edu.cn

Abstract

Grover's algorithm delivers a quadratic speed-up for searching an unstructured space, but its behaviour is governed by a single, easily mis-set quantity: the number of marked items M. A single search must apply the Grover iterator close to $\lfloor \pi/4 \times \sqrt{(N/M)} \rfloor$ times, and any deviation drives the success probability away from its peak. The problem becomes acute when the goal is not one solution but the complete set of solutions, because each measurement returns at most one item and an inaccurate count causes searches to stop early, repeat needlessly, or run without end. This survey organises the literature around two tightly coupled questions: how to enumerate many or all solutions efficiently, and how to obtain and maintain a usable estimate of the solution count. We review the geometry of amplitude amplification and the overshoot phenomenon, the principal families of find-all strategies (sampling with replacement, oracle modification, and over-sampling), the spectrum of counting and amplitude-estimation techniques (phase-estimation-based counting, maximum-likelihood and iterative estimators, and signal-processing formulations), and the count-free approaches based on exponential schedules and fixed-point amplification. We compare query and space costs, analyse the cost–accuracy trade-off that links estimation to discovery, and discuss the recent move toward adaptive, quantum–classical hybrid schemes that refine the count during execution. We close with open problems on near-term hardware, robustness, and the interaction between algorithmic and physical error.

Keywords: Quantum search, grover's algorithm, amplitude amplification, quantum counting, amplitude estimation, multi-solution enumeration, fixed-point search, quantum–classical hybrid algorithms.

Article History:
Received: July 12, 2023
Revised: September 16, 2023
Accepted: November 28, 2023
Available Online: December 30, 2023

1. Introduction

Quantum search occupies a peculiar position in the catalogue of quantum algorithms. Unlike Shor's factoring algorithm, which offers a super-polynomial advantage but applies to a narrow problem, Grover's algorithm offers only a quadratic advantage yet applies to an enormous range of tasks: any problem whose candidate solutions can be recognised by an efficiently computable predicate (Grover, 1996). Constraint satisfaction, combinatorial optimisation cast as feasibility testing, database lookup, and brute-force inversion of one-way functions all reduce, at least in principle, to the same primitive—find an input that an oracle marks as good. Because the predicate is treated as a black box, the speed-up is generic, and because it is only quadratic, it is also delicate: constant factors, mis-set parameters, and hardware noise all matter in practice.

The delicacy is concentrated in one quantity. The number of marked items, written M, determines how many times the Grover iterator should be applied. Apply it the right number of times and a measurement returns a marked item with probability exceeding ninety per cent; apply it too few or too many times and the probability falls, sometimes precipitously. When a single solution is sought, a modest misjudgement of M is usually tolerable, and several classical wrappers exist to recover from it. The situation changes when the objective is to enumerate every solution. Now the algorithm must be invoked many times, each invocation is sensitive to the current value of M, and an error in the count

ISSN: © 2023 INATGI (Institute of Advanced Technology and Green Innovation). Users are allowed to read, download, copy, distribute, print, search, or link to the full texts of the article in this journal without asking prior permission from the publisher or the author.
See: https://inatgi.in/index.php/rft/index for more information.

propagates into both correctness—which solutions are found—and cost—how many oracle calls are spent. Enumerating all solutions is the natural requirement in settings where completeness is safety-critical: listing every vulnerable code path, every infeasible configuration in a design, or every low-energy molecular conformation.

Two research threads have grown up around these facts. The first asks how to recover many or all solutions efficiently once the count is known, and includes sampling-with-replacement schemes analysed through the coupon-collector lens, oracle-modification schemes that suppress already-found items, and brute over-sampling that collects a histogram. The second asks how to obtain the count in the first place, and spans quantum counting built on phase estimation, maximum-likelihood and iterative amplitude estimators designed for shallow circuits, and recent formulations grounded in quantum signal processing. A third, smaller thread sidesteps the count entirely using exponential search schedules or fixed-point amplification. These threads are usually presented separately, but they are not independent: the accuracy demanded of the estimator is set by the sensitivity of the search, and the cost charged by the estimator is paid out of the same query budget as the discovery.

The historical arc helps to motivate this organisation. Grover's 1996 construction was quickly followed by a tight analysis of its success probability and by the realisation that the same machinery could be generalised to amplitude amplification and turned, via phase estimation, into a counter. For roughly two decades the dominant paradigm was estimate-then-search: obtain a count, fix the schedule, and run. The advent of noisy intermediate-scale devices changed the cost calculus by making circuit depth, rather than qubit count, the binding constraint, which in turn drove a wave of shallow estimators and a renewed interest in deterministic and fixed-point variants. Most recently, the recognition that the count can be refined online—using the discovery loop's own measurements—has begun to dissolve the boundary between the estimation and discovery phases. Reading the literature along this arc clarifies why methods that look unrelated are in fact responses to the same underlying sensitivity.

This survey treats the two questions together. Section 2 reviews the geometry of amplitude amplification and makes precise the sense in which M is the controlling parameter, including the overshoot or "soufflé" phenomenon. Section 3 surveys the find-all strategies and their query costs. Section 4 surveys solution-count estimation, quantifies the sensitivity of a single search to a mis-estimated count, and develops the cost–accuracy trade-off that couples estimation to discovery. Section 5 covers count-free methods. Section 6 draws cross-cutting comparisons—query-complexity landscape, near-term hardware constraints, the distinction between algorithmic and physical error, and application domains. Section 7 lists open problems, and Section 8 concludes. Table 1 fixes the notation used throughout.

Table 1. Notation used throughout the survey.

Symbol	Meaning
N	Size of the search space (number of candidate elements).
n	Number of qubits encoding the space, $n = \lceil \log_2 N \rceil$.
M	True number of marked items (solutions).
$\hat{M}$	Working estimate of the solution count.
θ	Search angle, $\theta = \arcsin\sqrt{(M/N)}$.
j	Number of applications of the Grover iterator (oracle calls).
ε	Error margin of a count estimate, commonly chosen $\approx \sqrt{N}$.
G	The Grover iterator (oracle reflection followed by diffusion).
p(j)	Probability that a measurement after j iterations returns a marked item.

2. Amplitude Amplification and the Role of the Solution Count

ISSN: © 2023 INATGI (Institute of Advanced Technology and Green Innovation). Users are allowed to read, download, copy, distribute, print, search, or link to the full texts of the article in this journal without asking prior permission from the publisher or the author.
See: https://inatgi.in/index.php/rft/index for more information.

2.1 Geometry of the search

Grover's algorithm begins by placing the n qubits in a uniform superposition over all N basis states, then repeatedly applies the Grover iterator G. Each application is the composition of two reflections: the oracle marks solutions by flipping the sign of their amplitudes, and the diffusion operator reflects all amplitudes about their mean. The textbook account of these operators and their circuit realisations is given by Nielsen and Chuang (2010). The crucial observation, due in its tight form to Boyer, Brassard, Høyer and Tapp (1998), is that the entire evolution is confined to a two-dimensional plane spanned by the equal superposition of marked states and the equal superposition of unmarked states. Within that plane the state is a unit vector, and each application of G rotates it by a fixed angle 2θ, where $\theta = \arcsin\sqrt{(M/N)}$.

Because the dynamics are a pure rotation, the probability of measuring a marked item after j iterations has the closed form $p(j) = \sin^2((2j + 1)\theta)$. This single expression underlies almost everything that follows. The probability is maximised when the rotated angle $(2j + 1)\theta$ is closest to $\pi/2$, which occurs at the integer j nearest to $(\pi/4)\sqrt{(N/M)} - 1/2$; rounding gives the familiar optimal count $j^* = \lfloor \pi/4 \times \sqrt{(N/M)} \rfloor$. At j^* the success probability is at least $1 - M/N$, comfortably above ninety per cent whenever M is small relative to N. The number of oracle queries spent is exactly j^*, which is $\Theta(\sqrt{(N/M)})$ and embodies the quadratic speed-up over the $\Theta(N/M)$ expected queries of classical sampling.

2.2 Overshoot and the "soufflé" problem

Since p(j) is periodic, continuing past j^* does not help; it hurts. The success probability rises to its peak, then falls back toward zero, then rises again, oscillating with a period set by θ. Brassard memorably likened this to a soufflé that collapses if left in the oven too long. The practical consequence is that one cannot simply "run Grover for a while" and expect a good outcome—the stopping time must be matched to θ, and therefore to M. Figure 1 plots p(j) for a fixed space of size $N = 2^{12}$ and three different solution counts. Larger M means a larger θ, a faster rotation, an earlier first peak, and a shorter window of high probability; the open markers indicate j^* for each curve.

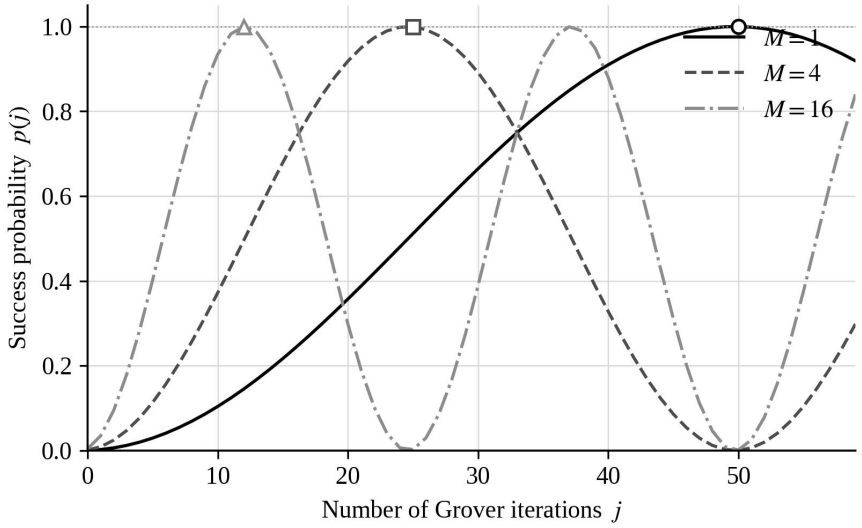

Fig. 1. Success probability $p(j) = \sin^2((2j+1)\theta)$ versus the number of Grover iterations for a space of size $N = 2^{12}$ and three solution counts. Larger M shortens the period and the high-probability window; open markers show the optimal iteration count j^* for each case.

Three lessons follow from the figure. First, the location of the first peak depends on M, so the algorithm needs the count before it can choose its stopping time. Second, the curves for different M cross repeatedly, so a stopping time tuned for one count can land near a trough for another—overshooting is not a benign error. Third, the high-probability plateau is broad for small M and narrow for large M, which means tolerance to mis-estimation degrades exactly as the

ISSN: © 2023 INATGI (Institute of Advanced Technology and Green Innovation). Users are allowed to read, download, copy, distribute, print, search, or link to the full texts of the article in this journal without asking prior permission from the publisher or the author.
See: https://inatgi.in/index.php/rft/index for more information.

search becomes harder. A family of modified-phase variants replaces the $\pm\pi$ reflections with tunable rotations to reach a chosen marked state with certainty rather than high probability (Long, 2001), and more recent work pushes deterministic search to the limit allowed by a restricted oracle (Roy, Jiang and Schuster, 2022) and analyses its behaviour under phase noise (Leng, Yang and Wang, 2023). These refinements remove the residual failure probability at the peak, but they do not remove the dependence on knowing where the peak lies.

A concrete instance fixes intuition. Take $N = 2^{20} \approx 10^6$ and $M = 4$. The search angle is $\theta = \arcsin\sqrt{(4/2^{20})} \approx 1.95\times10^{-3}$ radians, the optimal count is $j^* = \lfloor \pi/4 \times \sqrt{(2^{20}/4)} \rfloor = \lfloor \pi/4 \times 512 \rfloor = 402$, and the peak success probability is essentially one. Now suppose the count were mis-set to $\hat{M} = 1$. The schedule would call for $\hat{\jmath} = \lfloor \pi/4 \times 1024 \rfloor = 804$ iterations, almost exactly twice the optimum; the state would sweep past the peak and back down, and the achieved probability $\sin^2((2\cdot804+1)\theta)$ would fall to a small value. A factor-of-four error in the count thus halves or doubles the iteration count and can convert a near-certain success into a likely failure—while consuming roughly double the queries. This is the quantitative content of "the count is the controlling parameter."

2.3 From one solution to many

Amplitude amplification, the generalisation of Grover's algorithm formalised by Brassard, Høyer, Mosca and Tapp (2002), replaces the uniform-superposition preparation with an arbitrary state-preparation unitary and the marked-state predicate with an arbitrary Boolean test. The same rotation geometry holds, with θ now defined by the initial overlap with the good subspace. This abstraction matters for the multi-solution problem because it lets one reason uniformly about searches whose marked fraction changes as solutions are discovered and removed. When several solutions exist, a single optimally tuned search returns one of them, each equally likely; obtaining the rest requires repetition, and repetition is where the count re-enters as a budgeting and stopping question rather than merely a tuning one.

3. Strategies for Enumerating Many or All Solutions

Suppose every one of the M solutions must be reported. Because a tuned search yields a single, uniformly random solution, the enumeration problem is a covering problem layered on top of the search problem. The literature offers three broad strategies, which differ in how they handle duplicates and in what they assume about M.

3.1 Sampling with replacement and the coupon-collector cost

The simplest scheme runs an independent, optimally tuned search and records whatever solution it returns, repeating until all M distinct solutions have appeared. Because each run draws uniformly from the M solutions with replacement, this is precisely the coupon collector's problem (Motwani and Raghavan, 1995): collecting all M coupons takes about $M \times H_M \approx M \ln M$ draws on average, where H_M is the M-th harmonic number. Each draw is a search costing $\Theta(\sqrt{(N/M)})$ queries, so the total query cost is on the order of $\sqrt{(N/M)} \times M \ln M = \Theta(\sqrt{(NM)} \log M)$. The logarithmic factor is the price of blind resampling: the last few coupons are slow to arrive because most draws rediscover items already held.

This approach has the virtue of simplicity—the oracle is never modified—but two weaknesses. The log factor inflates cost, and, more seriously, the scheme needs a termination rule. Knowing when "all M" have been seen presupposes knowing M; without it, one must guess, and a wrong guess either stops early or loops forever. The dependence on the count is thus inherited directly from the single-solution case and amplified by repetition.

3.2 Oracle modification: marking out discovered solutions

A more efficient strategy edits the oracle after each discovery so that an already-found solution is no longer recognised as marked. Each search then draws from a strictly shrinking pool, every search returns a fresh solution, and duplicates vanish entirely. The number of searches needed is therefore close to M rather than M log M. Summing the per-search

ISSN: © 2023 INATGI (Institute of Advanced Technology and Green Innovation). Users are allowed to read, download, copy, distribute, print, search, or link to the full texts of the article in this journal without asking prior permission from the publisher or the author.
See: https://inatgi.in/index.php/rft/index for more information.

cost as the pool shrinks from M down to one solution gives a total of roughly $\Sigma (\pi/4)\sqrt{(N/m)}$ over m from M to 1, which is well approximated by the integral $(\pi/2)\sqrt{(NM)}$; the total query cost is $\Theta(\sqrt{(NM)})$, shedding the logarithmic factor.

The cost of this efficiency is that the oracle grows: each removed solution adds a clause or an ancilla-mediated condition, so the per-query gate cost can increase with the number of solutions removed. In most analyses this overhead is treated as part of the oracle complexity f(N) and folded into the gate count rather than the query count, leaving the query advantage intact. Oracle modification is the de facto baseline for find-all comparisons because it is asymptotically optimal among count-aware schemes, but it still relies on a count: the per-search tuning uses the current estimate, and the stopping rule—when the pool is believed empty—uses it too.

Figure 2 contrasts the total oracle-call cost of the three strategies as a function of M for a fixed space $N = 2^{20}$. The oracle-modified curve is the lowest non-trivial cost across the range; the coupon-collector curve sits above it by the logarithmic factor; and the over-sampling curve, discussed next, dominates for small M because its cost is essentially independent of M.

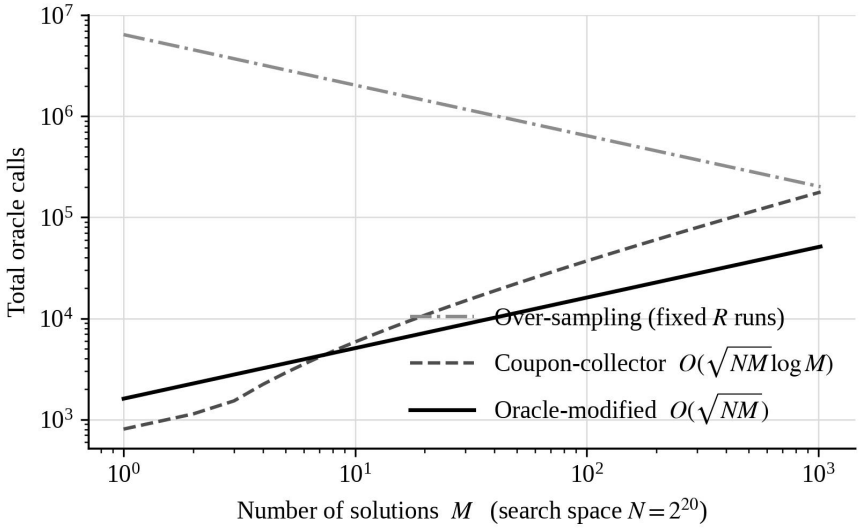

Fig. 2. Total oracle-call cost of three enumeration strategies versus the number of solutions M for a fixed space $N = 2^{20}$ (log–log axes). Oracle modification attains $O(\sqrt{(NM)})$; the coupon-collector scheme adds a logarithmic factor; fixed-budget over-sampling is wasteful unless M is large.

3.3 Over-sampling and histogram methods

A third approach, common in early demonstration code, fixes a large number of repetitions in advance, runs the tuned search that many times, and reads off the most frequent measurement outcomes from the resulting histogram. With enough repetitions the M genuine solutions dominate the histogram and can be separated from noise. The method is robust and trivially parallel, but it is profligate: the repetition count is chosen conservatively and is typically orders of magnitude larger than necessary, so for small M the total cost dwarfs that of the count-aware schemes, as the steep upper curve in Figure 2 shows. It also offers no principled stopping rule and no theoretical optimality guarantee. Over-sampling is best understood as a baseline that trades query efficiency for implementation simplicity and resilience to a poor count estimate.

3.4 Summary of find-all strategies

Table 2 summarises the three strategies along the axes that matter for practice: whether the oracle is rewritten during execution, the dominant query cost, the dependence on a count estimate, and the chief failure mode. The recurring theme is that efficiency and count-dependence trade off against each other: the cheaper a scheme is in queries, the more

ISSN: © 2023 INATGI (Institute of Advanced Technology and Green Innovation). Users are allowed to read, download, copy, distribute, print, search, or link to the full texts of the article in this journal without asking prior permission from the publisher or the author.
See: https://inatgi.in/index.php/rft/index for more information.

tightly it leans on an accurate M, both for per-search tuning and for deciding when to stop.

Table 2. Comparison of strategies for enumerating all M solutions.

Strategy	Oracle edited?	Total query cost	Needs M?	Main failure mode
Sampling with replacement	No	$O(\sqrt{(NM)} \log M)$	Yes (stop rule)	Slow tail; loops if M wrong
Oracle modification	Yes	$O(\sqrt{(NM)})$	Yes (tuning + stop)	Early stop / over-run if $\hat{M} \neq M$
Fixed-budget over-sampling	No	$R \cdot O(\sqrt{(N/M)})$	Weakly	Wasteful; no optimality bound

3.5 Verification overhead, exclusion mechanics, and optimality

Two practical points deserve emphasis. First, every measured candidate must be checked against the oracle before it is accepted, and in the oracle-modification scheme each acceptance triggers an edit to the marking circuit. The edit is commonly implemented by conjugating the oracle with a comparison against the set of found solutions, or by appending an ancilla-controlled phase that unmarks them; either way the per-query gate cost can drift upward as the found set grows. Because this growth is in gate count rather than query count, it is usually charged to the oracle complexity f(N) and does not alter the $\Theta(\sqrt{(NM)})$ query order, but on depth-limited hardware it is not negligible and can favour the resampling scheme, whose oracle never changes, despite its worse query order. Second, the resampling scheme can avoid editing the oracle altogether by filtering duplicates classically—accepting a measured solution only if it is new—at the cost of paying the coupon-collector tail. The choice among schemes is therefore partly a hardware question: query-optimal oracle modification is preferable when oracle edits are cheap, and edit-free resampling when they are not.

On optimality, the quadratic speed-up is known to be the best possible for unstructured search: any quantum algorithm that finds a marked item among N with bounded error must make $\Omega(\sqrt{(N/M)})$ oracle calls, and Grover's algorithm meets this bound to within a small constant—its query count comes within a few per cent of the proven lower limit (Boyer, Brassard, Høyer and Tapp, 1998). The find-all cost $\Theta(\sqrt{(NM)})$ is correspondingly tight up to the duplicate-handling factor, so the room for improvement in this regime lies not in asymptotics but in constants, in robustness to a poor count, and in circuit depth—the very dimensions the rest of this survey emphasises.

4. Estimating the Number of Solutions

Every count-aware strategy in Section 3 presupposes a value of M. In realistic settings the true count is unknown and must be estimated, and the estimate carries an error. This section first quantifies how much that error costs a single search, then reviews the principal estimation techniques, and finally develops the trade-off that couples estimation cost to discovery cost.

4.1 How sensitive is a search to a mis-estimated count?

Let $\hat{M}$ be the estimate used to choose the iteration count and let M be the truth. The search applies the iterator $\hat{\jmath} = \lfloor \pi/4 \times \sqrt{(N/\hat{M})} \rfloor$ times, but the rotation angle is governed by the true $\theta = \arcsin\sqrt{(M/N)}$. The achieved success probability is therefore $\sin^2((2\hat{\jmath} + 1)\theta)$ rather than the peak value. Figure 3 plots this achieved probability against the ratio $\hat{M}/M$ for two true counts in a space of size $N = 2^{16}$. The picture is informative and slightly asymmetric. Underestimating the count (ratio below one) makes the chosen iteration count too large, pushing the state past the peak and down the far side of the sine; the probability falls steeply. Overestimating (ratio above one) makes the iteration count too small,

ISSN: © 2023 INATGI (Institute of Advanced Technology and Green Innovation). Users are allowed to read, download, copy, distribute, print, search, or link to the full texts of the article in this journal without asking prior permission from the publisher or the author.
See: https://inatgi.in/index.php/rft/index for more information.

leaving the state short of the peak; the probability also falls, but more gently. The staircase pattern is a discretisation artefact of the floor in ĵ, and is itself a reminder that for small counts a change of a single iteration can move the success probability appreciably.

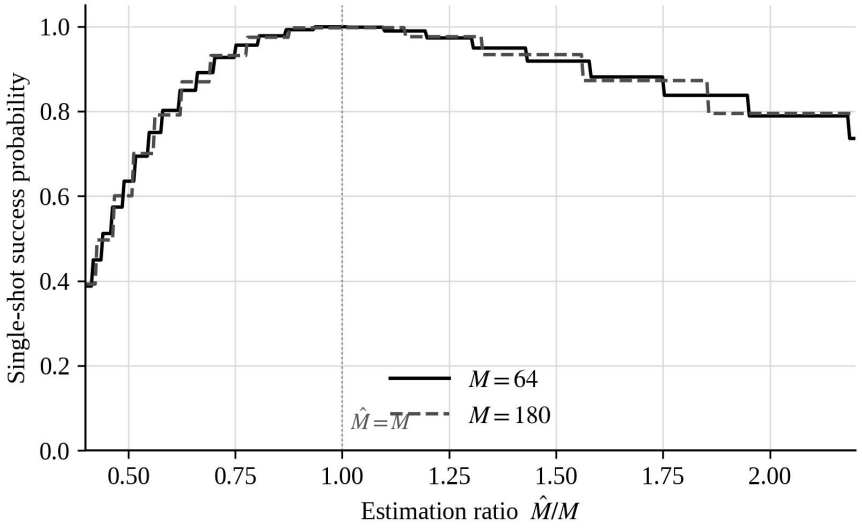

Fig. 3. Single-search success probability as a function of the estimation ratio M̂/M, for N = 2^{16} and two true counts. Underestimation (ratio < 1) degrades the probability more sharply than overestimation; the staircase reflects the integer iteration count.

The figure also explains a subtlety reported in adaptive schemes: in the early phase of an enumeration, when many solutions remain, the relative error |M̂ − M|/M is small even if the absolute error is not, so the success probability stays high and the search appears healthy. It is only later, as the pool shrinks and the relative error grows, that mis-estimation begins to bite. Any estimator or correction mechanism is therefore most needed precisely when the fewest observations remain to inform it—a tension that motivates the hybrid methods of Section 5.3.

4.2 Quantum counting via phase estimation

The canonical estimator is quantum counting, introduced by Brassard, Høyer and Tapp (1998). Its insight is that the Grover iterator G, viewed as a rotation by 2θ, has eigenvalues e^{±i2θ}; estimating that eigenphase by quantum phase estimation yields θ and hence M = N sin²θ. Phase estimation with m counting qubits prepares a register of controlled powers of G and applies an inverse quantum Fourier transform, returning θ to a precision that improves geometrically in m. The number of oracle calls is on the order of 2^m, and the resulting error margin in the count scales roughly as N / 2^m. Choosing m so that 2^m ≈ √N yields an error margin ε ≈ √N at a cost of Θ(√N) oracle calls—comfortably below the Θ(√(NM)) cost of the subsequent enumeration.

Quantum counting is accurate and well understood, but it is demanding for near-term hardware: it requires the counting register, the controlled applications of high powers of G, and a Fourier transform, giving deep circuits with many two-qubit gates. This cost has driven a decade of work on estimators that retain the quadratic scaling while shedding the controlled operations and the Fourier transform.

4.3 Estimation without phase estimation

Suzuki, Uno, Raymond, Tanaka, Onodera and Yamamoto (2020) proposed estimating the amplitude—and hence the count—by maximum likelihood over the outcomes of several circuits that apply different, uncontrolled powers of the Grover iterator. Because no controlled-G operations or Fourier transform are needed, the circuits are markedly shallower, and the scheme still approaches the optimal quadratic scaling. The same group later analysed maximum-

ISSN: © 2023 INATGI (Institute of Advanced Technology and Green Innovation). Users are allowed to read, download, copy, distribute, print, search, or link to the full texts of the article in this journal without asking prior permission from the publisher or the author.
See: https://inatgi.in/index.php/rft/index for more information.

likelihood estimation on noisy hardware, showing how to incorporate a depolarising-style noise model into the likelihood so that the estimate remains usable when circuits are imperfect (Tanaka, Suzuki, Uno, Raymond, Onodera and Yamamoto, 2021).

Grinko, Gacon, Zoufal and Woerner (2021) introduced iterative quantum amplitude estimation, which dispenses with phase estimation in favour of an adaptive sequence of Grover-amplified measurements that progressively narrows a confidence interval for θ. The method comes with rigorous confidence guarantees and uses only the bare iterator, making it attractive on devices where ancilla counts and circuit depth are at a premium. These maximum-likelihood and iterative estimators are now the default choice for amplitude estimation on noisy intermediate-scale hardware, and they apply directly to counting by identifying the amplitude with $\sqrt{(M/N)}$.

In practice these shallow estimators share a common recipe that is worth stating explicitly, because it recurs in the adaptive hybrids of Section 5.3. One chooses a small set of amplification depths, runs each depth for a number of shots, records the empirical fraction of marked outcomes at each depth, and fits the single parameter θ that best explains all the data under the model $p = \sin^2((2j+1)\theta)$. Spreading the depths geometrically concentrates information near the steepest part of the sine, which is what gives these methods their near-optimal scaling; keeping each individual circuit shallow is what makes them runnable on noisy hardware. The estimate of M then follows immediately as $N \sin^2\theta$. The same likelihood machinery, evaluated incrementally as new shots arrive, is exactly what an online scheme uses to track a count that is itself changing as solutions are removed.

4.4 Approximate counting and signal-processing formulations

A complementary line of work targets the looser goal of counting to within a multiplicative factor. Aaronson and Rall (2020) gave a strikingly simple approximate-counting algorithm that uses only Grover iterations and elementary classical post-processing, yet matches the optimal query scaling for multiplicative-factor estimation. Venkateswaran and O'Donnell (2021) then asked how much of that algorithm's adaptivity is essential and proved that nonadaptive Grover iterations suffice to reach the tight $O(\sqrt{(N/\varepsilon)})$ query complexity, clarifying the role of adaptivity in counting. Most recently, Rall and Fuller (2023) recast amplitude estimation in the language of quantum signal processing, unifying several estimators and exposing the design freedom in the choice of amplifying polynomial.

Table 3 places the principal estimators side by side. The trend over time is unmistakable: from the deep, ancilla-heavy phase-estimation circuit of quantum counting toward shallow, controlled-operation-free circuits whose cost is borne mostly in classical post-processing and in a modest number of repeated short experiments—exactly the profile suited to current devices.

Table 3. Representative solution-count / amplitude-estimation methods.

Method	Query scaling	Circuit features	Reference
Quantum counting (QPE)	$O(\sqrt{N})$ for $\varepsilon \approx \sqrt{N}$	Counting register, controlled G, QFT; deep	Brassard, Høyer & Tapp (1998)
Maximum-likelihood AE	near-optimal	No controlled G, no QFT; shallow	Suzuki et al. (2020)
MLAE on noisy hardware	near-optimal	Noise model folded into likelihood	Tanaka et al. (2021)
Iterative QAE	rigorous CI	Bare iterator; adaptive intervals	Grinko et al. (2021)
Approximate counting	$O(\sqrt{(N/\varepsilon)})$	Grover iterations + classical post-proc.	Aaronson & Rall (2020)
Nonadaptive counting	$O(\sqrt{(N/\varepsilon)})$, tight	Nonadaptive Grover iterations	Venkateswaran & O'Donnell (2021)

ISSN: © 2023 INATGI (Institute of Advanced Technology and Green Innovation). Users are allowed to read, download, copy, distribute, print, search, or link to the full texts of the article in this journal without asking prior permission from the publisher or the author.
See: https://inatgi.in/index.php/rft/index for more information.

QSP-based AE	near-optimal	Signal-processing polynomial design	Rall & Fuller (2023)

4.5 The estimation–discovery cost trade-off

Estimation is not free, and its cost is drawn from the same query budget as discovery, so the two must be balanced. For phase-estimation counting the oracle cost grows as the error margin ε shrinks, roughly as N/ε, while the discovery that follows costs about $(\pi/2)\sqrt{(NM)}$ regardless of ε (once ε is small enough not to wreck the search). Setting the two equal identifies a natural operating point. Figure 4 plots both costs against ε for $N = 2^{20}$ and M = 256. To the right of the crossover the estimator is cheap relative to discovery and adds only a lower-order term; to the left, the estimator dominates and erodes the quantum advantage. Driving ε to a constant pushes the estimation cost to Θ(N), the cost of classical exhaustive counting, which defeats the purpose entirely.

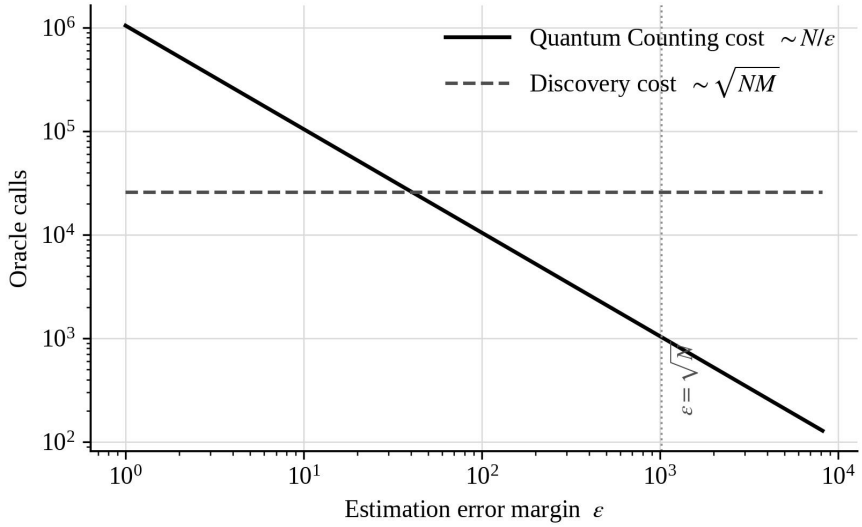

Fig. 4. Estimation cost (≈ N/ε) and discovery cost (≈ √(NM)) versus the error margin ε, for $N = 2^{20}$ and M = 256 (log–log axes). Reducing ε far below √N makes estimation the dominant cost and erodes the quantum advantage.

The trade-off explains why ε ≈ √N is the conventional choice: it makes the estimator a negligible fraction of the total while keeping the count accurate enough that the search starts near its peak. It also exposes the weakness of the one-shot estimate-then-search pipeline. The single ε must be conservative enough to serve the entire enumeration, including its late, count-sensitive phase; but buying that accuracy up front is wasteful in the early phase, where, as Section 4.1 showed, a loose estimate is perfectly adequate. This mismatch is the opening that adaptive methods exploit.

5. Searching Without a Prior Count

A separate body of work removes the assumption that M is known in advance, either by adapting the schedule on the fly or by using amplification dynamics that are insensitive to the count.

5.1 Exponential-schedule search

The earliest count-free method, due to Boyer, Brassard, Høyer and Tapp (1998), runs Grover with a randomly chosen number of iterations drawn from a window whose size grows geometrically between rounds. Starting from a small window, the algorithm samples a random iteration count, measures, and tests the outcome; if it fails, the window is enlarged by a constant factor and the process repeats. Because the window eventually brackets the unknown optimum, a good iteration count is sampled with constant probability, and the expected total cost remains Θ(√(N/M)) up to

ISSN: © 2023 INATGI (Institute of Advanced Technology and Green Innovation). Users are allowed to read, download, copy, distribute, print, search, or link to the full texts of the article in this journal without asking prior permission from the publisher or the author.
See: https://inatgi.in/index.php/rft/index for more information.

constants even though M is never estimated. The exponential schedule is the standard answer when a single solution must be found with no count; extending it to enumerate all solutions reintroduces the coupon-collector overhead and a termination problem, so it is rarely used unmodified for find-all tasks.

5.2 Fixed-point amplification

Fixed-point search attacks the overshoot problem directly. Standard amplification overshoots because it keeps rotating; a fixed-point scheme instead drives the state monotonically toward the target so that, once near it, further iterations do no harm. Grover (2005) gave the first fixed-point construction using $\pi/3$ phase rotations, guaranteeing convergence regardless of the precise marked fraction. The catch, made precise by Yoder, Low and Chuang (2014), is that the original fixed-point scheme converges only linearly and thereby sacrifices the quadratic speed-up. Their contribution was a fixed-point algorithm, built on Chebyshev-polynomial phase sequences, that preserves the optimal quadratic query complexity while guaranteeing a chosen success probability over the broadest possible range of marked fractions. In effect it needs only a reliable lower bound on M/N rather than its exact value.

Fixed-point amplification is attractive when only a crude bound on the count is available and a single solution suffices. For enumeration, however, the marked fraction changes after every discovery, so the bound must be maintained as the pool shrinks; the method controls the tuning sensitivity of Section 4.1 but not the budgeting and stopping questions that dominate the find-all setting. It is therefore complementary to, rather than a replacement for, count estimation.

5.3 Adaptive quantum–classical hybrids

The most active recent direction blends a quantum search loop with a classical estimator that refines the count from observed outcomes as the enumeration proceeds. The classical layer maintains a belief over candidate counts—often as a set of hypotheses with associated probabilities—and updates that belief by Bayesian conditioning on the running record of successes and failures, periodically resetting the working estimate $\hat{M}$ to the most probable value. Such schemes inherit the shallow-circuit advantages of the iterative and maximum-likelihood estimators (Suzuki et al., 2020; Grinko et al., 2021) but apply them online, spending no separate up-front estimation budget and instead extracting count information from the very measurements that the discovery loop already performs.

The appeal of this design is that it resolves the early-versus-late mismatch identified in Section 4.5: it spends little effort while the relative error is small and the search is healthy, and it concentrates correction in the late phase, where mis-estimation causes early stopping or unbounded looping. The reported effect is a sharp reduction in missed solutions—from several per cent of the total down to near zero—while the dominant query cost remains $\Theta(\sqrt{(NM)})$, because the classical updates are arranged to contribute only a lower-order overhead. The principal open questions for these hybrids concern the stability of the online estimate: aggressive updates can chase statistical noise and oscillate, while conservative updates converge slowly, and the right balance appears to depend on N, on the prior, and on the schedule of updates.

6. Cross-Cutting Analysis

6.1 The query-complexity landscape

Gathering the costs from the preceding sections gives a compact picture. A single tuned search costs $\Theta(\sqrt{(N/M)})$. Enumerating all solutions costs $\Theta(\sqrt{(NM)})$ with oracle modification and $\Theta(\sqrt{(NM)}\log M)$ with blind resampling. Quantum counting to an error margin $\sqrt{N}$ adds $\Theta(\sqrt{N})$, a lower-order term. Adaptive hybrids preserve the $\Theta(\sqrt{(NM)})$ discovery cost while folding estimation into it. The standing assumption $M \leq \sqrt{N}$ is what keeps enumeration sub-classical: if M approaches N, the $\sqrt{(NM)}$ cost approaches N and the quantum advantage evaporates, since one might as well scan classically. Within that regime the leading costs of the count-aware methods agree to constants, so the practical differences among them lie in those constants, in gate cost per query, and in robustness to a poor count—not

ISSN: © 2023 INATGI (Institute of Advanced Technology and Green Innovation). Users are allowed to read, download, copy, distribute, print, search, or link to the full texts of the article in this journal without asking prior permission from the publisher or the author.
See: https://inatgi.in/index.php/rft/index for more information.

in asymptotic query order.

6.2 Space and near-term feasibility

The qubit budget of the search itself is modest: $n = \lceil \log_2 N \rceil$ qubits hold the superposition, and the diffusion and oracle reflections add only a logarithmic number of gates per iteration beyond the oracle's own cost. The classical state—discovered solutions and, for hybrids, the hypothesis set—grows only linearly in M and in the estimation margin ε, so it is not the bottleneck. The true near-term obstacle is circuit depth: the optimal iteration count grows as √(N/M), and each iteration invokes the oracle, so even moderate N implies thousands of sequential oracle calls without error correction. This is why the estimation literature has migrated away from phase-estimation counting toward shallow, controlled-operation-free estimators (Suzuki et al., 2020; Grinko et al., 2021; Rall and Fuller, 2023), and why fixed-point methods that reach the target with certainty rather than probability (Long, 2001; Yoder, Low and Chuang, 2014; Roy, Jiang and Schuster, 2022) are of more than theoretical interest: removing the residual failure probability reduces the number of repetitions, and repetitions are expensive when each is deep.

6.3 Algorithmic error versus physical error

It is worth separating two error sources that are easily conflated. Physical error—decoherence, gate infidelity, readout noise—corrupts the quantum state and is the province of error mitigation and error correction. Algorithmic error—here, the gap between $\hat{M}$ and M—arises even with perfect hardware, because estimation algorithms return an estimate with a finite margin by design, and because that margin is deliberately kept loose to control cost (Section 4.5). The two interact: noisy hardware degrades the very measurements an estimator relies on, which is why noise-aware likelihoods (Tanaka et al., 2021) matter, and why analyses of deterministic search under phase noise (Leng, Yang and Wang, 2023) are relevant. But the count problem would persist on a fault-tolerant machine, and the adaptive correction of $\hat{M}$ is a classical post-processing remedy for a classical-statistical defect, distinct from quantum error correction.

6.4 Application domains

The find-all-with-uncertain-count setting recurs wherever completeness matters and the number of targets is not known a priori. In software and hardware security, every code path or configuration matching a vulnerability signature must be enumerated, and missing even one can leave a system exposed; the count of vulnerable sites is exactly what one does not know in advance. In design verification for structures or circuits, all unsafe configurations must be surfaced. In computational chemistry, the low-energy conformations of a molecule form a solution set whose size is unknown and whose completeness affects conclusions about stability and binding. In databases, answering a complex multi-attribute predicate over a large table is a find-all query whose result cardinality is itself as hard to predict as the query is to run, so a fixed pre-estimate is unreliable. In all of these, the predicate is a natural oracle, completeness is the objective, and the count is genuinely uncertain—precisely the conditions under which the adaptive and fixed-point techniques surveyed here add value over a one-shot estimate-then-search pipeline.

6.5 Benchmarking and reproducibility

Comparing methods fairly is harder than it looks. Reported results depend on the choice of N and M, on whether the oracle cost f(N) is held fixed across methods, on the termination threshold used to declare a search complete, and on how the initial count estimate is generated. A scheme can appear to dominate simply because its evaluation used a more favourable count range or a more forgiving stopping rule. Sound practice is to fix the search-space sizes and solution counts in advance, sweep M across the sub-classical regime up to roughly √N, draw the initial estimate from the same estimator for every method under test, repeat each configuration enough times to report variance rather than a single trial, and state the stopping criterion explicitly—for example, a maximum run of consecutive failed searches. Because full quantum simulation becomes infeasible well before interesting N, many studies validate a fast custom simulator against a circuit-level simulator on small instances and then extrapolate; the validation step and the simulator

ISSN: © 2023 INATGI (Institute of Advanced Technology and Green Innovation). Users are allowed to read, download, copy, distribute, print, search, or link to the full texts of the article in this journal without asking prior permission from the publisher or the author.
See: https://inatgi.in/index.php/rft/index for more information.

should be published so that results can be reproduced. Reporting both the missed-solution count and the oracle-call count, with their distributions, lets readers see the accuracy–efficiency trade-off directly rather than through a single summary number.

7. Open Problems and Future Directions

Several questions remain open. The first concerns stability of online estimation: adaptive hybrids need an update rule that converges quickly without chasing statistical fluctuations, and a principled theory relating the update schedule, the prior, and the variance of the running estimate is still lacking. The second concerns optimal allocation of a fixed query budget between estimation and discovery when the budget is bounded—Figure 4 identifies an asymptotic operating point, but the finite-budget, finite-M version is a genuine optimisation problem that has received little formal treatment. The third concerns the integration of fixed-point amplification with enumeration: fixed-point methods control tuning sensitivity but assume a maintained lower bound on the marked fraction, and combining them with an online count estimate that supplies and updates that bound is largely unexplored.

A fourth direction is hardware co-design. Because depth is the binding near-term constraint, estimators and search loops that minimise sequential oracle calls—through nonadaptive schedules (Venkateswaran and O'Donnell, 2021), parallel batches, or signal-processing constructions (Rall and Fuller, 2023)—deserve empirical comparison on real devices, not only asymptotic analysis. A fifth is robustness certification: applications that motivate find-all search are often safety-critical, yet most evaluations report average missed-solution counts rather than tail guarantees, and a framework that bounds the probability of missing any solution would be valuable. Finally, the boundary case $M \approx \sqrt{N}$, where the quantum advantage thins, merits sharper analysis, since real instances do not respect the convenient assumption $M \leq \sqrt{N}$ and may demand a graceful hand-off to classical enumeration.

8. Conclusion

Grover's algorithm reduces an enormous class of search problems to a single tunable rotation, but the tuning depends on the number of solutions, and that dependence is the thread connecting the otherwise disparate literatures of multi-solution enumeration and solution-count estimation. We have argued that the two should be read together. The find-all strategies—resampling, oracle modification, and over-sampling—trade query efficiency against reliance on an accurate count, with oracle modification attaining the optimal $\Theta(\sqrt{(NM)})$ at the price of the tightest count-dependence. The estimation techniques have evolved from deep phase-estimation counting toward shallow, controlled-operation-free estimators better matched to near-term hardware, while the estimation–discovery trade-off pins the conventional $\varepsilon \approx \sqrt{N}$ operating point and exposes the inefficiency of a single up-front estimate. Count-free methods—exponential schedules and fixed-point amplification—address parts of the problem but not the budgeting and stopping questions central to enumeration. The most promising current direction interleaves a quantum search loop with an online, Bayesian count estimate, spending estimation effort where it is actually needed and reducing missed solutions toward zero at no asymptotic cost. The open problems—estimator stability, budget allocation, fixed-point integration, depth-aware design, and tail-risk certification—suggest that the interaction between knowing how many solutions exist and finding them all will remain a productive area as quantum hardware matures.

References

Aaronson, S., & Rall, P. (2020). Quantum approximate counting, simplified. In Symposium on Simplicity in Algorithms (SOSA), pp. 24–32. SIAM. https://doi.org/10.1137/1.9781611976014.5

Boyer, M., Brassard, G., Høyer, P., & Tapp, A. (1998). Tight bounds on quantum searching. Fortschritte der Physik, 46(4–5), 493–505. https://doi.org/10.1002/(SICI)1521-3978(199806)46:4/5<493::AID-PROP493>3.0.CO;2-P

Brassard, G., Høyer, P., Mosca, M., & Tapp, A. (2002). Quantum amplitude amplification and estimation. In Quantum

ISSN: © 2023 INATGI (Institute of Advanced Technology and Green Innovation). Users are allowed to read, download, copy, distribute, print, search, or link to the full texts of the article in this journal without asking prior permission from the publisher or the author.
See: https://inatgi.in/index.php/rft/index for more information.

Computation and Information, Contemporary Mathematics, vol. 305, pp. 53–74. American Mathematical Society. https://doi.org/10.1090/conm/305/05215

Brassard, G., Høyer, P., & Tapp, A. (1998). Quantum counting. In Automata, Languages and Programming (ICALP), Lecture Notes in Computer Science, vol. 1443, pp. 820–831. Springer. https://doi.org/10.1007/BFb0055105

Grinko, D., Gacon, J., Zoufal, C., & Woerner, S. (2021). Iterative quantum amplitude estimation. npj Quantum Information, 7, 52. https://doi.org/10.1038/s41534-021-00379-1

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC), pp. 212–219. https://doi.org/10.1145/237814.237866

Grover, L. K. (2005). Fixed-point quantum search. Physical Review Letters, 95, 150501. https://doi.org/10.1103/PhysRevLett.95.150501

Leng, J., Yang, F., & Wang, X.-B. (2023). Improving D2p Grover's algorithm to reach performance upper bound under phase noise. Physical Review Research, 5, 023202. https://doi.org/10.1103/PhysRevResearch.5.023202

Long, G.-L. (2001). Grover algorithm with zero theoretical failure rate. Physical Review A, 64, 022307. https://doi.org/10.1103/PhysRevA.64.022307

Motwani, R., & Raghavan, P. (1995). Randomized Algorithms. Cambridge University Press. (Coupon collector's problem, pp. 57–63.)

Nielsen, M. A., & Chuang, I. L. (2010). Quantum Computation and Quantum Information (10th Anniversary ed.). Cambridge University Press.

Rall, P., & Fuller, B. (2023). Amplitude estimation from quantum signal processing. Quantum, 7, 937. https://doi.org/10.22331/q-2023-03-02-937

Roy, T., Jiang, L., & Schuster, D. I. (2022). Deterministic Grover search with a restricted oracle. Physical Review Research, 4, L022013. https://doi.org/10.1103/PhysRevResearch.4.L022013

Suzuki, Y., Uno, S., Raymond, R., Tanaka, T., Onodera, T., & Yamamoto, N. (2020). Amplitude estimation without phase estimation. Quantum Information Processing, 19, 75. https://doi.org/10.1007/s11128-019-2565-2

Tanaka, T., Suzuki, Y., Uno, S., Raymond, R., Onodera, T., & Yamamoto, N. (2021). Amplitude estimation via maximum likelihood on noisy quantum computer. Quantum Information Processing, 20, 293. https://doi.org/10.1007/s11128-021-03215-9

Venkateswaran, R., & O'Donnell, R. (2021). Quantum approximate counting with nonadaptive Grover iterations. In 38th International Symposium on Theoretical Aspects of Computer Science (STACS), LIPIcs vol. 187, pp. 59:1–59:12. https://doi.org/10.4230/LIPIcs.STACS.2021.59

Yoder, T. J., Low, G. H., & Chuang, I. L. (2014). Fixed-point quantum search with an optimal number of queries. Physical Review Letters, 113, 210501. https://doi.org/10.1103/PhysRevLett.113.210501

ISSN: © 2023 INATGI (Institute of Advanced Technology and Green Innovation). Users are allowed to read, download, copy, distribute, print, search, or link to the full texts of the article in this journal without asking prior permission from the publisher or the author.
See: https://inatgi.in/index.php/rft/index for more information.