

A competitive NISQ and qubit-efficient solver for the LABS problem

Marco Sciorilli,^{1,2} Giancarlo Camilo,¹ Thiago O. Maciel,¹
Askery Canabarro,¹ Lucas Borges,^{1,3} and Leandro Aolita¹

¹*Quantum Research Center, Technology Innovation Institute, Abu Dhabi, UAE*

²*Laboratoire d'Informatique de Paris 6, CNRS,*

Sorbonne Université, 4 Place Jussieu, 75005 Paris, France

³*Federal University of Rio de Janeiro, Caixa Postal 652, Rio de Janeiro, RJ 21941-972, Brazil*

Pauli Correlation Encoding (PCE) is as a qubit-efficient variational approach to combinatorial optimization problems. The method offers a polynomial reduction in qubit count and a super-polynomial suppression of barren plateaus. Here, we extend the PCE-based framework to solve the Low Autocorrelation Binary Sequences (LABS) problem, a notoriously hard problem often used as a benchmark for classical and quantum solvers. To illustrate this, we simulate two variants of the PCE quantum solver for LABS instances of up to $N = 45$ binary variables: one with commuting and one with maximally non-commuting sets of Pauli operators. The simulations use 4 qubits and a circuit Ansatz with a total of 30 two-qubit gates. We benchmark our method against the state-of-the-art classical solver and other quantum schemes. We observe improved scaling in the total time to reach the exact solution, outperforming the best-performing classical heuristic while using only a fraction of the quantum resources required by other quantum approaches. In addition, we perform proof-of-principle demonstrations on IonQ's **Forte** quantum processor, showing that the final solution is resilient to noise. Our findings point at PCE-based solvers as a promising quantum-inspired classical heuristic for hard problems as well as a tool to reduce the resource requirements for actual quantum algorithms.

The Low Autocorrelation Binary Sequences (LABS) problem [1–8] is a well-known NP-hard problem with applications in engineering, communications, and statistical physics. It offers a challenging benchmark for optimization algorithms [2], as finding the optimal global solutions becomes intractable even for moderate problem sizes N . Classical methods, though powerful for small instances, face significant challenges in finding good solutions for large N , even when employing massively parallel approaches on CPUs and GPUs [3]. Quantum algorithms have recently been proposed as a potential alternative to classical heuristics for solving combinatorial optimization problems more efficiently [9]. Recent work showed that the Quantum Approximate Optimization Algorithm (QAOA) [9] can solve LABS with a runtime scaling of $\mathcal{O}(1.46^N)$, which can be further boosted to $\mathcal{O}(1.21^N)$ if combined with a Grover-like quantum minimum-finding subroutine [4]. However, such a quadratic improvement requires a large-scale fault-tolerant quantum computer, which is currently not available. This letter investigates the effectiveness of solving the LABS problem with NISQ devices using Pauli correlation encoding (PCE), a qubit-efficient technique to encode binary variables into quantum states recently introduced in [10]. Notably, PCE encodes N binary variables using a polynomially smaller number n of qubits, *i.e.*, $N = \text{poly}(n)$, opening up the possibility of exploring large LABS instances with a number of qubits feasible in near-term devices. In particular, using numerical simulations we show that

the PCE-based LABS solver has the runtime scaling of $\mathcal{O}(1.330^N)$ for even and $\mathcal{O}(1.326^N)$ for odd instances, which improves over the QAOA scaling and is marginally better even than the state-of-the-art classical heuristics, given by Tabu search [11]. In addition, we do a proof-of-principle deployment on IonQ's QPU for instance size $N = 120$ (to the best of our knowledge, the largest instance of the problem demonstrated on quantum hardware to date). We observed that training requires only a modest number of shots and that the final solution quality is robust to device errors.

The LABS problem. Given an integer $N > 2$, the LABS problem consists in finding a binary sequence $x \in \{-1, +1\}^{\otimes N}$ of length N that minimizes the *sidelobe energy*

$$E(x) = \sum_{\ell=1}^{N-1} C_\ell^2(x), \quad C_\ell(x) = \sum_{i=1}^{N-\ell} x_i x_{i+\ell}, \quad (1)$$

where $C_\ell(x)$ is the ℓ -th *autocorrelation* of x . Notice that $C_0(x) = N$ for all x , while $(N - \ell) \bmod 2 \leq |C_\ell(x)| \leq N - \ell$ for $\ell > 0$. In statistical physics terms, $E(x)$ can be interpreted as the energy of a system of N spins subject to long-range 2- and 4-body interactions known as the Bernasconi model [5]. Equivalently, the problem amounts to maximizing the *merit factor*,

$$F(x) = \frac{N^2}{2E(x)}, \quad (2)$$

which is (half) the ratio between $C_0(x)^2$ (the *peak*) and the sum of all the other squared autocorrelations

(the *sidelobes*). For this reason, sequences with low autocorrelation provide a large peak-to-sidelobe ratio that is a desirable feature for various engineering applications involving pulse modulation [12].

The LABS problem is known to be NP-hard. Solutions are degenerate, and the optimization landscape has a peculiar “golf course” structure [13], plagued by a large number of local minima and extremely isolated global minima [3]. For small enough N , the optimal sequence can be found by brute force enumeration of all 2^N possible sequences, but the exponential growth in search space quickly becomes impractical. For odd N , exact solutions can often (but not always) be found by restricting the search to sequences satisfying the skew-symmetry condition $x_{(N+1)/2+\ell} = (-1)^\ell x_{(N+1)/2-\ell}$ for all $\ell \in [1, (N-1)/2]$, which reduces the effective search space size to $2^{N/2}$ [1]. In practice, provably optimal solutions are only known for $N \leq 66$ [1]. For larger instances, different heuristic algorithms with milder exponential scalings but no optimality guarantees have been employed [14–18]. The best performing one, Memetic Tabu Search [11], has been empirically shown to achieve a runtime scaling of $\mathcal{O}(1.34^N)$ in the range $N \leq 40$ [4, 6, 7]. Although these milder scalings encourage attempting larger instances, for $N > 200$, the solution quality of classical heuristics has been observed to degrade significantly, and the asymptotic behavior of the optimal merit factor $F_N := \max_{x \in \{-1,+1\}^{\otimes N}} F(x)$ for large N remains an open question. Golay conjectured based on the ergodicity postulate the asymptotic upper bound $F_N \lesssim 12.3248/(8\pi N)^{\frac{3}{2N}}$ as $N \rightarrow \infty$ [8], but the best known heuristic sequences achieve the much smaller value $F_N \approx 6.3421$ [19, 20]. Fig. 3 in App. 1 shows a visual summary of the state-of-the-art in LABS solutions using classical solvers.

Qubit-efficient LABS solver. Here we propose a qubit-efficient solver for the LABS problem following the *Pauli-correlation encoding* (PCE) framework recently introduced in [10]. Given a set $\Pi \equiv \{\Pi_i\}_{i \in [N]}$ of n -qubit Pauli strings Π_i , PCE encodes binary variables x_i into correlators from this set as

$$x_i := \text{sgn}(\langle \Pi_i \rangle), \quad i \in [N], \quad (3)$$

where sgn is the sign function and $\langle \Pi_i \rangle := \langle \Psi | \Pi_i | \Psi \rangle$ denotes the expectation value of Π_i on a quantum state $|\Psi\rangle$. Since there are $4^n - 1$ available traceless Pauli operators, this allows a compressed encoding using $n \ll N$ qubits. Below we discuss two explicit choices for the set Π used throughout this work. We take $|\Psi\rangle$ as the output of a parameterized quantum circuit with a brickwork architecture [10] and parameters θ to be classically optimized using a variational approach (see App. 2 for details). The goal of the

parameter optimization is to minimize the non-linear loss function

$$\mathcal{L} = \sum_{\ell=1}^{N-1} \sum_{i,j=1}^{N-\ell} \tilde{x}_i \tilde{x}_{i+\ell} \tilde{x}_j \tilde{x}_{j+\ell} - \beta \sum_{i=1}^N \tilde{x}_i^2, \quad (4)$$

where $\tilde{x}_i := \tanh(\alpha \langle \Pi_i \rangle)$ is a real-valued relaxation of the binary variable (3) better-suited for gradient-based optimization. The rescaling factor $\alpha > 1$ is introduced to restore the non-linear behaviour of the \tanh , given that the n qubit correlators have magnitude decreasing polynomially with n . The last term in (4) is a regularization term forcing all correlators away from zero, which was observed to improve the solver’s performance (details on the choice of regularization and the hyperparameters α, β are presented in App. 3). Once the training is complete, the circuit output state is measured and a bit-string x is obtained via Eq. (3). In the following we benchmark the performance of PCE with two different choices for the Pauli correlator set Π . The first, $\Pi^{(C)}$, contains randomly selected but mutually *commuting* Paulis. The second, $\Pi^{(NC)}$, in contrast, is designed so as to maximize the number of mutually *non-commuting* Pauli operators. The intuition is that non-commutativity gives rise to mutually unbiased measurements [21], *i.e.*, measurement outcomes that provide as much new information about the underlying state $|\Psi\rangle$ as possible [22], which should increase the independence (hence the expressivity) of the assigned variables x_i in Eq. (3). We refer to App. 6 for an explicit description of the procedure to construct $\Pi^{(C)}$ and $\Pi^{(NC)}$.

Numerical results. We perform numerical simulations using the two PCE-based quantum solvers for the LABS instances of sizes $N \leq 45$ using the Pauli correlation sets $\Pi^{(C)}$ and $\Pi^{(NC)}$ with $n = 4$ qubits each. $\Pi^{(NC)}$ performed consistently better than $\Pi^{(C)}$, so we chose it to benchmark against the other solvers. We tuned the circuit depth to achieve the best possible approximation ratio as explained in detail in App. 2. Notably, the optimal circuit depth scales linearly with n (hence sublinearly with N). In both cases, we fixed the number of two-qubit gates to 30, corresponding to 15 layers of the brickwork circuit and a total of 150 parameters.

As our figure of merit, we adopt the time-to-solution (TTS), here defined as the number of cost function evaluations required to observe the exact solution, following the approach outlined in [7]. Moreover, although the time required to reach optimality is a rigorous metric, we assess also the effectiveness of *approximating* LABS solutions by analyzing the time required to reach the second- and third-best objective function values, referred as

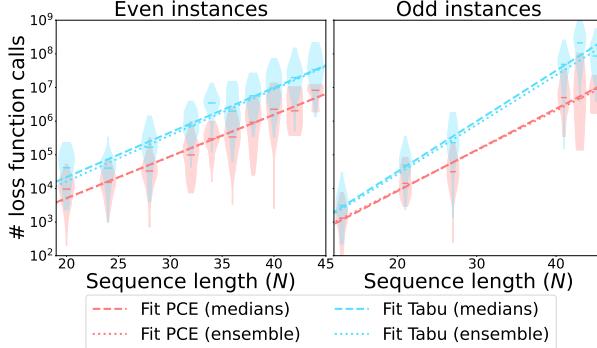


FIG. 1: Time-to-solution benchmark. Scaling comparison of the TTS for the quantum LABS solver based on Pauli Correlation Encoding using $\Pi^{(NC)}$ (PCE, red) vs. Tabu Search (Tabu, blue) for even (left) and odd (right) values of N . Here, TTS is defined as the total number of cost function evaluations required to reach the exact solution [7]. Dashed-dotted lines denote linear regression over the list of median values for each N , while dotted curves correspond to linear regression over 50 points per instance size (except for $N \in [41, 45]$, with 10 runs each). The distribution of points regarding different initializations is depicted by violin plots. The fits show a clear exponential scaling in all the cases, with both median and ensemble statistics consistent with a power-law advantage of PCE over the classical baseline. The precise scaling exponents and confidence intervals are shown in Table II in App. 4.

$\text{TTS}_{1\text{st}}$ and $\text{TTS}_{2\text{nd}}$, respectively. For many combinatorial optimization problems, finding a good approximation of the objective function is also hard. To obtain our empirical scalings, we used problem instances with unique canonical solutions (*i.e.*, the ground state is only degenerate due to the inherent symmetries of the problem [7]): $N \in \{20, 24, 28, 32, 34, 36, 38, 40, 42, 44\}$ for even and $N \in \{13, 21, 27, 41, 43, 45\}$ for odd instances.

Heuristic	N	b	CI	R^2
QAOA [4]	All	1.46	1.42–1.50	>0.94
Tabu [3, 7]	Even	1.358	1.297–1.422	0.97
	Odd	1.409	1.317–1.508	0.97
PCE	Even	1.330	1.276–1.387	0.97
	Odd	1.325	1.273–1.380	0.97

TABLE I: TTS fit parameters. Estimated base coefficient b controlling the exponential scaling of the TTS as in Eq. (5), with respective confidence intervals (CI) and R^2 values of the linear fit of $\log(\text{TTS})$ for the solvers (PCE, QAOA, Tabu Search) for even and odd instances.

We benchmark different solvers using the same set of instances, running the optimization 50 times per instance size N (except for $N \in [41, 45]$, with 10

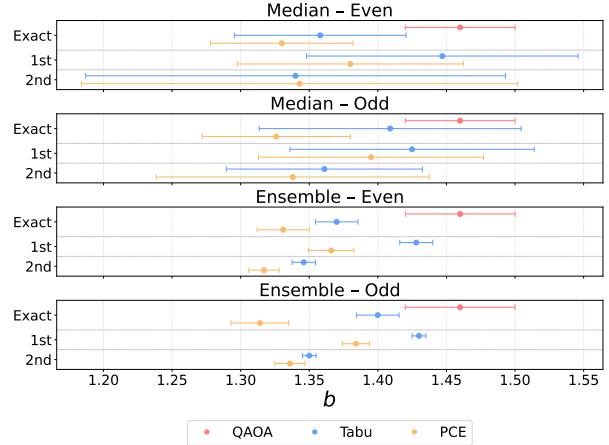


FIG. 2: TTS scaling for different solvers. Fitted basis coefficient b of the exponential scaling in Eq. (5) for our PCE quantum solver using $\Pi^{(NC)}$, QAOA [4], and Tabu Search (classical) LABS solvers. The corresponding fit data appears in Table II in App. 4. We report the results for TTS, $\text{TTS}_{1\text{st}}$, and $\text{TTS}_{2\text{nd}}$ results, corresponding respectively to the exact solution (Exact) and to approximate solutions given by the first (1st) and second (2nd) excited energy levels.

runs each), separating the even and odd cases since they are expected to have the same scaling, but with different constant factors [7] (see Fig. 1). We then used this data to fit the exponential curve

$$\text{TTS} = c \cdot b^N \quad (5)$$

via linear regression on $\log(\text{TTS})$ using both the median TTS for each sequence length N , and also the full data ensemble (and similarly for $\text{TTS}_{1\text{st}}$ and $\text{TTS}_{2\text{nd}}$). The median fit data for the TTS basis coefficient b is presented in Tab. I, while the complete fit data (median and ensemble) for TTS, $\text{TTS}_{1\text{st}}$ and $\text{TTS}_{2\text{nd}}$, are detailed in Tab. II in App. 4. The results are summarized in Fig. 2.

Our results show a clear scaling advantage of PCE compared with the QAOA quantum solver of [4] for both even and odd instances. For a fair comparison, the final quantum amplitude amplification step suggested in [4] was omitted here, since takes the solver beyond the scope of near-term devices. In addition, we used Tabu without its memetic component, which was observed to not offer any scaling advantage [7], and tenure parameter chosen from the recent GPU implementation in [3]. All the TTS scalings were obtained by local simulations. The PCE results are competitive (within error bars), and in the case of approximate solutions even consistently better, with respect to Tabu search, considered the state-of-the-art classical heuristics for LABS [7], revealing a separation that exceeds the error bars, see

App. 4. While minor polynomial advantage factors do not alter the overall exponential scaling, in practice even minor improvements in the basis might play an important role when considering larger problem instances [10] (see details in App. 9). Thus, the approach is promising even as a quantum-inspired algorithm, although in that case one must factor in the significant overhead from classical simulation of the quantum circuit. Finally, assuming the observed TTS scalings remain the same for larger instances, to run PCE in a quantum hardware, we still need to account the overhead of the gate operations of running quantum circuits multiple times to estimate the loss function and its gradients to a given precision ϵ . The cost of gate operations are platform-specific, but assuming all gates done sequentially, it scales as $\mathcal{O}(N)$ (see App. 2); the sample complexity is $\mathcal{O}(N^7/\epsilon^2)$ (see App. 8); and to estimate the gradient we need $\mathcal{O}(N)$ calls. Thus, the quantum approach incurs an overall overhead of $\mathcal{O}(N^9/\epsilon^2)$. Even considering the overall overhead of $\mathcal{O}(N^9)$ when running PCE in a quantum hardware, it is still possible to obtain a preliminary estimate of a crossover point where quantum advantage might be achieved. In App. 9, we showed that, from instance sizes of order of thousands, we would expect advantage in terms of query access to the exact solution, but since the classical solvers can not output exact solutions on instances larger than $N > 66$, we might see better (approximate) solutions from the quantum approach sooner.

Experimental results. In addition, we perform a proof-of-principle experimental demonstration on the IonQ’s **Forte** quantum processor. For that, instead of using the randomly selected sets of Paulis used in the numerical demonstrations, here we used the same set used in [10] of k -body correlators on n qubits of the form $X^k \mathbb{1}^{n-k}, Y^k \mathbb{1}^{n-k}, Z^k \mathbb{1}^{n-k}$ for a fixed k (here $n = 10$ and $k = 2$, which corresponds to a quadratic compression in the number of qubits). From the experimental perspective, this is more convenient than $\Pi^{(NC)}$ since it requires only 3 measurement setups. We have deployed a classically pre-trained circuit on the $N = 120$ LABS instance and compared the effect of hardware noise on the dispersion of the loss function (with respect to different initializations) for the first and last optimization steps. The full details are presented in App. 5. The results show that shot noise has only a minor impact on solution quality, *i.e.*, a few thousand shots already yield sufficient statistics for experimental training. Hardware noise however, while not preventing training, significantly degrades the final solutions and can undermine the expected TTS scaling advantage. This makes error mitigation, or

higher compression rate, a key lever for potential training on near-term QPUs.

Concluding remarks. Two highlights of our approach are its qubit-efficiency and the strong performance obtained on a hard problem like LABS. With only 4 qubits and a total of 30 two-qubit gates, our solver attains a competitive run-time scaling on a hard sequence-length regime. Our results show that the performance of PCE is improved through a careful choice of the Pauli correlation set II so as to maximize the expressivity of the binary variables encoded via Eq. (3). Even though PCE-based variational methods have no performance guarantees, the fact that they consistently display such a strong performance on hard problems such as MaxCut and LABS urges for further investigations. In addition, we deployed our solver on a commercially available QPU for a large size $N = 120$ instance, showing that training can be done with modest shot budgets, and with a final solution that is robust against hardware noise. Numerical simulations suggest that incorporating quantum error mitigation could make full training possible on near-term devices. Moreover, we notice that PCE can also be used as a warm start for classical solvers [23]: in App. 7, we show preliminary numerical explorations indicating that the runtime scaling can be further decreased by running classical solvers on top of the PCE solution. We note that, for the hardware-friendly variant used in the experiment, $n = 8$ qubits would be enough to tackle the largest sequence lengths for which LABS has known exact solutions ($N = 66$), while $n \leq 20$ qubits are enough to challenge even the largest instances solved by state-of-the-art classical heuristics [24]. This regime offers an interesting testbed for our solver as a quantum-inspired heuristic. A challenge there would be the high number of loss-function evaluations required. This could be mitigated by incorporating problem symmetries into the circuit ansatz and combining this with exact geodesic transport for parameter updates [25]. Other research directions may be problems beyond LABS, such as those in the Quantum Optimization Benchmark Library (QOBLIB) [26]. Our results raise the question of whether PCE has a potential quantum advantage. Although the constant overhead of simulating a full quantum state vector may delay the crossover point after which a runtime advantage becomes evident to larger instance sizes, our method still shows promising performance even in a quantum-inspired setting. All these are exciting open questions for future work.

Acknowledgments. We thank Allan Tosta for fruitful discussions.

-
- [1] T. Packebusch and S. Mertens, Low autocorrelation binary sequences, *Journal of Physics A: Mathematical and Theoretical* **49**, 165001 (2016).
- [2] I. P. Gent and T. Walsh, CSPLib: A Benchmark Library for Constraints (1999), available online at <http://www.csplib.org/> as prob005.
- [3] Z. Zhang, J. Shen, N. Kumar, and M. Pistoia, New improvements in solving large labs instances using massively parallelizable memetic tabu search (2025), arXiv:2504.00987 [cs.DC].
- [4] R. Shaydulin, C. Li, S. Chakrabarti, M. DeCross, D. Herman, N. Kumar, J. Larson, D. Lykov, P. Minssen, Y. Sun, Y. Alexeev, J. M. Dreiling, J. P. Gaebler, T. M. Gatterman, J. A. Gerber, K. Gilmore, D. Gresh, N. Hewitt, C. V. Horst, S. Hu, J. Johansen, M. Matheny, T. Mengle, M. Mills, S. A. Moses, B. Neyenhuis, P. Siegfried, R. Yalovetzky, and M. Pistoia, Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem, *Science Advances* **10**, 10.1126/sciadv.adm6761 (2024).
- [5] Bernasconi, J., Low autocorrelation binary sequences : statistical mechanics and configuration space analysis, *J. Phys. France* **48**, 559 (1987).
- [6] J. E. Gallardo, C. Cotta, and A. J. Fernández, Finding low autocorrelation binary sequences with memetic algorithms, *Applied Soft Computing* **9**, 1252 (2009).
- [7] B. Bošković, F. Brzlez, and J. Brest, Low-autocorrelation binary sequences: On improved merit factors and runtime predictions to achieve them, *Applied Soft Computing* **56**, 262 (2017).
- [8] M. Golay, The merit factor of long low autocorrelation binary sequences, *IEEE Transactions on Information Theory* **28**, 543 (1982).
- [9] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm (2014), arXiv:1411.4028 [quant-ph].
- [10] M. Sciorilli, L. Borges, T. L. Patti, D. García-Martín, G. Camilo, A. Anandkumar, and L. Aolita, Towards large-scale quantum optimization solvers with few qubits, *Nature Communications* **16**, 476 (2025).
- [11] F. Glover, Tabu search–part i, *ORSA Journal on Computing* **1**, 190 (1989).
- [12] S. W. Golomb and R. A. Scholtz, Generalized barker sequences, *IEEE Transactions on Information Theory* **11**, 533 (1965).
- [13] S. Mertens and C. Bessenrodt, On the ground states of the bernasconi model, *Journal of Physics A: Mathematical and General* **31**, 3731–3749 (1998).
- [14] Z. Liu, H. Deng, and S. Yang, Search of binary sequences with low aperiodic autocorrelation based on improved multi-objective genetic algorithm, *IET Radar, Sonar & Navigation* (2024).
- [15] Z. Wang and Y. Zhang, Low autocorrelation binary sequences: Best-known peak sidelobe level values, *IEEE Transactions on Aerospace and Electronic Systems* **58**, 3632 (2022).
- [16] B. Jia, F. Ji, W. Jia, and M. Li, Computational search of long skew-symmetric binary sequences with high merit factors, *IEEE Access* **10**, 26269 (2022).
- [17] M. A. Shoaib, S. A. Al-Agtash, and H. M. Al-Khateeb, A hybrid modified sine cosine algorithm using inverse filtering and clipping methods for low autocorrelation binary sequences, in *2022 13th International Conference on Information and Communication Systems (ICICS)* (IEEE, 2022) pp. 324–329.
- [18] J. K.-L. Gancayco, H. R. Tee, J. D. L. Paggabao, and E. P. Dadios, Efficient generation of low autocorrelation binary sequences, in *2020 IEEE Region 10 Conference (TENCON)* (IEEE, 2020) pp. 1018–1023.
- [19] J. Jedwab, D. J. Katz, and K.-U. Schmidt, Advances in the merit factor problem for binary sequences, *Journal of Combinatorial Theory, Series A* **120**, 882 (2013).
- [20] J. Jedwab, D. J. Katz, and K.-U. Schmidt, Littlewood polynomials with small l^4 norm (2013), arXiv:1205.0260 [math.NT].
- [21] Bandyopadhyay, Boykin, Roychowdhury, and Vatan, A new proof for the existence of mutually unbiased bases, *Algorithmica* **34**, 512 (2002).
- [22] W. K. Wootters and B. D. Fields, Optimal state-determination by mutually unbiased measurements, *Annals of Physics* **191**, 363 (1989).
- [23] A. G. Cadavid, P. Chandarana, S. V. Romero, J. Trautmann, E. Solano, T. L. Patti, and N. N. Hegade, Scaling advantage with quantum-enhanced memetic tabu search for labs (2025), arXiv:2511.04553 [quant-ph].
- [24] B. Pšeničnik, R. Mlinarić, J. Brest, and B. Bošković, Dual-step optimization for binary sequences with high merit factors, *Digital Signal Processing* **165**, 105316 (2025).
- [25] A. J. Ferreira-Martins, R. M. S. Farias, G. Camilo, T. O. Maciel, A. Tosta, R. Lin, A. Alhajri, T. Haug, and L. Aolita, Variational quantum algorithms with exact geodesic transport (2025), arXiv:2506.17395 [quant-ph].
- [26] T. Koch, D. E. B. Neira, Y. Chen, G. Cortiana, D. J. Egger, R. Heese, N. N. Hegade, A. G. Cadavid, R. Huang, T. Itoko, T. Kleinert, P. M. Xavier, N. Mohseni, J. A. Montanez-Barrera, K. Nakano, G. Nannicini, C. O’Meara, J. Pauckert, M. Proissl, A. Ramesh, M. Schicker, N. Shimada, M. Takeori, V. Valls, D. V. Bulck, S. Woerner, and C. Zoufal, Quantum optimization benchmark library – the intractable decathlon (2025), arXiv:2504.03832 [quant-ph].
- [27] B. Bošković, F. Brzlez, and J. Brest, A GitHub Archive for Solvers and Solutions of the `labs` problem, For updates, see https://github.com/borkob/git_labs. (2016).

- [28] R. H. Barker, Group synchronization of binary digital system, *Communication theory*, 273 (1953).
- [29] D. Kraft, *A Software Package for Sequential Quadratic Programming*, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht (Wiss. Berichtswesen d. DFVLR, 1988).
- [30] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization (2017), arXiv:1412.6980 [cs.LG].
- [31] M. Hollander, D. A. Wolfe, and E. Chicken, *Non-parametric Statistical Methods*, 3rd ed. (John Wiley & Sons, Hoboken, NJ, 2013) includes detailed treatment of the Kolmogorov–Smirnov test.
- [32] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, *Nature Communications* **9**, 10.1038/s41467-018-07090-4 (2018).
- [33] F. F. Ferreira, J. F. Fontanari, and P. F. Stadler, Landscape statistics of the low-autocorrelation binarystring problem, *Journal of Physics A: Mathematical and General* **33**, 8635 (2000).
- [34] R. Sarkar and E. van den Berg, On sets of commuting and anticommuting paulis (2019), arXiv:1909.08123 [quant-ph].
- [35] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, *Phys. Rev. A* **99**, 032331 (2019).
- [36] M. AbuGhanem, Ibm quantum computers: evolution, performance, and future directions, *The Journal of Supercomputing* **81**, 687 (2025).
- [37] I. Staff, IonQ Aria: Practical Performance, <https://ionq.com/resources/ionq-aria-practical-performance> (2025), last updated January 8, 2025.
- [38] Intel Corporation, Intel Core i9-14900KS (36M Cache, up to 6.20GHz) – Specifications (2024).

SUPPLEMENTARY INFORMATION

1. Classical state-of-the-art for LABS

Here we summarize the state-of-the-art of solutions to the LABS problem obtained by classical solvers. We use the data are publicly available at [27], combined with recent improvements in particular instances reported in [3]. For $N \leq 66$, solutions with the optimal solution merit factor

$$F_N := \max_{x \in \{-1,+1\}^{\otimes N}} F(x) \quad (6)$$

can be found using exhaustive search or branch and bound methods. For larger N , only heuristic solutions are known and there is no proof of optimality. The best known solutions were obtained using the Memetic Tabu Search meta-heuristics [7]. Fig. 3 depicts the best known solutions for $2 \leq N \leq 449$. The largest known merit factor is achieved for $N = 13$

(namely $F_{13} = 169/12 \approx 14.083$) by a so-called *Barker sequence* [28].

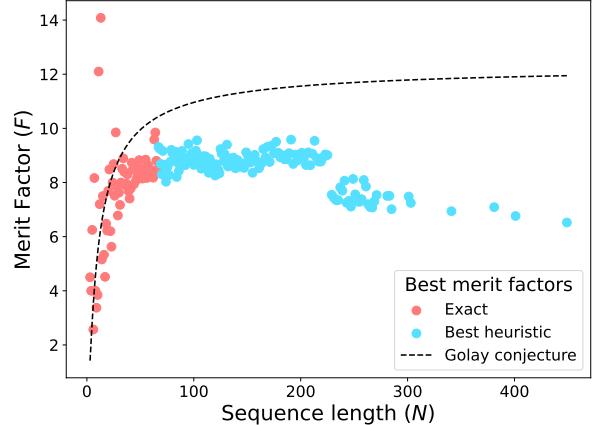


FIG. 3: **LABS state-of-the-art.** Best merit factor vs. problem size N for classical solvers. Red dots indicate the optimal merit factor $F_N := \max_{x \in \{-1,+1\}^{\otimes N}} F(x)$ obtained via exact solvers, while blue dots indicate the best known approximation to F_N obtained via heuristic solvers. The dashed curve is Golay’s conjectured asymptotic upper bound $\approx 12.3248/(8\pi N)^{3/2}$ [8].

2. Circuit complexity

Here we provide details on the variational circuit ansatz, the classical optimizer, and strategy to fix the circuit depth for the PCE quantum solver.

Variational circuit. For the variational quantum circuit, we employ a hardware-efficient ansatz with the same gate configuration described in [10], namely, RX, RY, and RZ rotations as single-qubit gates and the Mølmer–Sørensen (MS) gate – the native entangling gate of trapped-ion quantum computing platforms, which feature currently the highest fidelities among NISQ devices – for two-qubit interactions.

Classical optimizer. For the classical parameter optimization, we used the SLSQP algorithm [29], although analogous performance was observed using the Adam optimizer [30].

Circuit depth scaling. To determine a suitable circuit depth, we chose per each circuit size 4 different hard instances N (namely, 2 even and 2 odd) as mentioned in the main text and solved them using the PCE quantum solver with a circuit ansatz having an increasing number of layers. In each instance and number of layers, we run the solver with 1000 different random initializations and, for each run, record the corresponding fraction of the optimal merit factor F_N achieved (*i.e.*, approximation ratio).

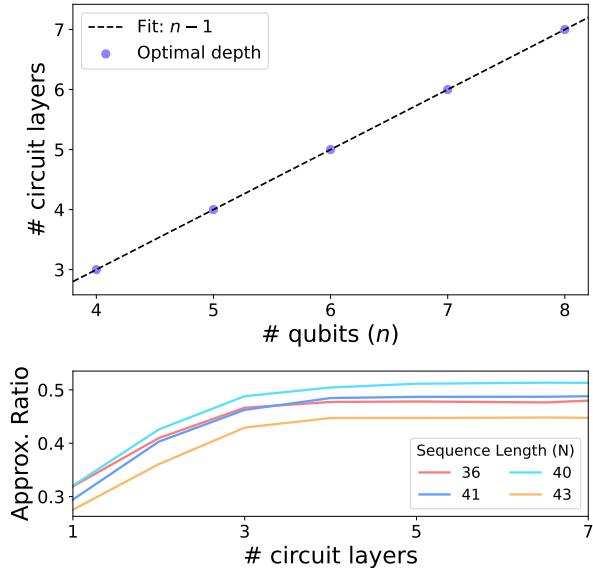


FIG. 4: **Circuit depth scaling.** (*Top*) Linear scaling of the optimal circuit depth with the number of qubits n for the qubit-efficient solver with quadratic compression using $\Pi^{(C)}$. This corresponds to a $\mathcal{O}(\sqrt{N})$ scaling with the problem size N . (*Bottom*) For each n (here illustrated for $n = 4$), the optimal circuit depth was determined by increasing the number of circuit layers until no significant improvement in the average approximation ratio was observed over 1000 runs.

Then we used the Kolmogorov–Smirnov test [31] to identify the point after which there were no more statistically significant improvements in the distributions of the approximation ratios, which we refer to as optimal circuit depth. As shown in Fig. 4, the optimal depth was observed to scale linearly with the number of qubits n , and therefore sublinearly with the instance size N . Interestingly, a linear scaling of the depth in n is also known to lead to approximate Haar randomness [32].

3. Regularization term and hyperparameters

In this section, we discuss the form of the regularization term and the tuning of hyperparameters α and β appearing in the loss function (4).

Regularization term. We first notice from Eq. (1) that the energy spectrum of LABS is strictly positive. In our formulation of the problem, however, the binary variables are relaxed to (hyperbolic tangents of) expectation values of Pauli operators, which take real values in the interval $[-1, 1]$. Therefore, if one tries to minimize the loss function 4 naively without any penalization term ($\beta = 0$), we quickly converge to the trivial solution $\tilde{x}_i = 0$ for all

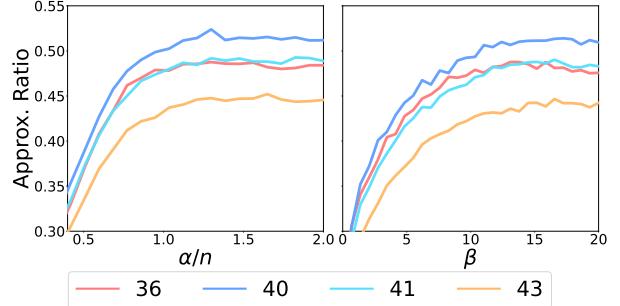


FIG. 5: **Hyperparameters tuning.** (*Left*) Tuning of the rescaling parameter α ; (*Right*) Tuning of the penalization constant β . In each case, we fixed the number of qubits to $n = 4$, the circuit depth to 10, and the optimal value was determined by increasing the corresponding parameter until no significant improvement in the average approximation ratio was observed over 1000 runs.

i (where all the expectation values go to 0), which does not offer any valuable insight for the binary version of the problem. For this reason, we introduced a regularization term with a positive coefficient β penalizing this strong attractor. For simplicity, we used the simple ℓ_2 -norm penalization $\beta \sum_{i=1}^N \tilde{x}_i^2$.

Tuning of α and β . For each instance and hyperparameter size, we ran the solver using 1000 distinct random initializations. For every run, we recorded the achieved fraction of the optimal merit factor F_N , referred to as the *approximation ratio*, as reported in Fig. 5. Next, we used the Kolmogorov–Smirnov test [31] to identify the hyperparameter size beyond which further increases did not result in statistically significant improvements in the approximation ratio distributions. As shown in Fig. 5 for the case of $n = 4$ qubits, this resulted in values $\alpha = 1.5 n$ and $\beta = 15$. For the purpose of our estimation of the Time-To-Solution (TTS), these were the values used through our numerical benchmarks.

4. PCE as a LABS approximate solver

For many combinatorial optimization problems, finding a good approximation of the objective function is also hard. Unlike the MaxCut problem, a rigorous proof of the approximation hardness for the LABS problem is still lacking. However, numerical studies and its unique optimization landscape strongly suggest that LABS is likely difficult to approximate (see [33]).

Even though the time required to reach optimality is a rigorous metric, it does not fully cover an algorithm's overall performance.

Notwithstanding, we evaluate our solver's performance in approximating solutions to the LABS

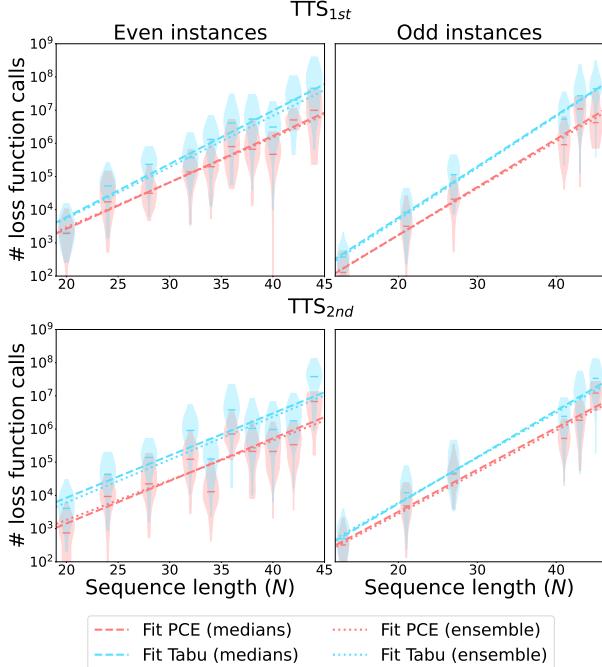


FIG. 6: Time-to-solution benchmark. Scaling comparison of the TTS_{1st} and TTS_{2nd} for the quantum LABS solver based on Pauli Correlation Encoding (PCE, red) vs. Tabu Search (Tabu, blue) for even (left) and odd (right) values of N . Here, TTS is defined as the total number of cost function evaluations required to reach the exact solution [7]. Dashed-dotted lines denote linear regression over the list of median values for each N , while dotted curves correspond to linear regression over the full dataset of 50 points per instance size. The distribution of points over different initializations is depicted by violin plots at each N . The fits show a clear exponential scaling in all the cases, with both median and ensemble statistics consistent with a power-law advantage of PCE over the classical baseline. The precise scaling exponents and confidence intervals are shown in Tab. II.

problem by analyzing the time-to-solution (TTS) required to reach the second- and third-best objective function values (*i.e.*, the first- and second-excited energy levels of Eq. 1) to assess approximate solutions to the LABS problem, here referred as TTS_{1st} and TTS_{2nd} , respectively.

In both estimates, we observed an exponential behavior similar to the exact solution. However, the advantage over Tabu Search revealed a separation that exceeds the error bars, see Fig. 6, with the corresponding exponential bases reported in Tab. II. We were unable to obtain the corresponding data for QAOA due to the highly demanding nature of its classical simulations at this scale [4].

5. Experimental deployment

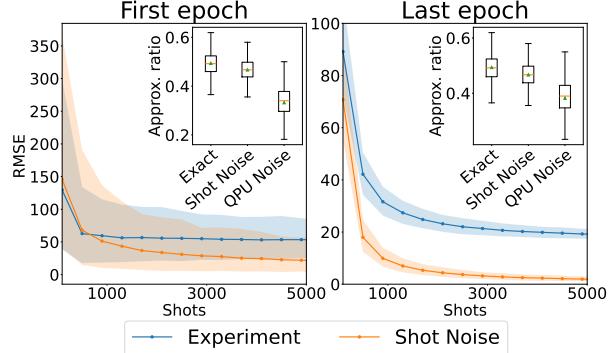


FIG. 7: Experiment. *Main panels.* Relative mean-squared error (RMSE) of the measured loss values across three quantum states, evaluated at the first optimization epoch (left) and at the final epoch (right), as a function of the number of measurement shots. Experimental data were collected on the IonQ trapped-ion Forte-1 QPU. For comparison, we also report the corresponding shot-noise-only RMSE obtained from numerical simulations. The circuit uses 10 qubits and 30 two-qubit gates and targets a LABS instance of size $N = 120$. *Insets.* Boxplot distributions of the mean approximation ratio obtained over 500 random initializations by simulating the full optimization under three noise models: exact statevector simulation (Exact), shot-noise-only (Shot noise), and hardware-noise matching the observed QPU error rates (QPU noise). Each inset correspond to simulations using the RMSE evaluations in the main panels.

We experimentally demonstrate our quantum solver on IonQ’s Forte-1 36-qubit trapped-ion processor using a LABS instance of size $N = 120$. The device is based on trapped ytterbium ions and offers all-to-all qubit connectivity. We employ a standard VQA brickwork ansatz expressed in the hardware-native gate set, consisting of alternating layers of parametrized entangling RZZ gates and single-qubit rotation layers built from the native GPI and GPI2 operations. The circuit depth is fixed to 10 and includes 50 two-qubit gates. We use the quadratic Pauli-correlation encoding and, for this experiment, restrict to the same-letter Pauli observables (XX, YY, ZZ) used originally in [10], which reduces the measurement overhead to three bases. Circuit parameters are optimized offline via classical simulation, and the resulting pre-trained circuit is then executed on the hardware. For three independent random initializations of the ansatz parameters, we measure the loss function at the first and last optimization epochs. For experimental run, we accumulated measurement statistics until the estimated values of the loss no longer changed. Figure 7 reports the relative error of the loss estimate as a function

Solution	Heuristic	Type	N	b	CI	c	CI	R^2
Exact	QAOA [4]	All	All	1.46	1.42–1.50	—	—	>0.94
		Tabu	Median	Even	1.358	1.297–1.422	47.57	9.738–232.4
				Odd	1.413	1.334–1.496	31.42	4.539–217.5
		Ensemble	Even	1.370	1.355–1.386	29.54	19.76–44.16	0.78
			Odd	1.404	1.392–1.417	29.67	21.96–40.09	0.88
	PCE	Median	Even	1.331	1.276–1.387	16.99	4.010–71.98	0.97
			Odd	1.325	1.273–1.380	29.54	7.470–116.8	0.99
		Ensemble	Even	1.332	1.313–1.351	16.36	10.05–26.65	0.77
			Odd	1.314	1.293–1.335	36.96	21.46–63.64	0.85
		PCE + Tabu	Median	Even	1.244	1.233–1.255	2288.2	1767.0–3030.2
				Odd	1.330	1.323–1.337	500.3	428.1–591.7
1st	Tabu	Median	Even	1.447	1.351–1.549	3.684	0.3434–39.53	0.95
			Odd	1.427	1.369–1.487	4.652	1.146–18.89	0.99
		Ensemble	Even	1.428	1.416–1.440	4.383	3.279–5.858	0.77
			Odd	1.431	1.426–1.435	3.923	3.522–4.371	0.98
	PCE	Median	Even	1.395	1.301–1.465	2.217	0.5327–32.91	0.95
			Odd	1.374	1.301–1.465	2.935	0.5328–32.91	0.98
		Ensemble	Even	1.366	1.349–1.382	5.795	3.834–8.759	0.73
			Odd	1.384	1.375–1.394	2.559	2.005–3.267	0.89
2nd	Tabu	Median	Even	1.340	1.196–1.502	24.32	0.4693–1260	0.81
			Odd	1.381	1.314–1.452	8.855	1.621–48.37	0.99
		Ensemble	Even	1.346	1.338–1.355	16.03	12.84–20.00	0.65
			Odd	1.367	1.363–1.371	11.61	10.42–12.93	0.91
	PCE	Median	Even	1.343	1.193–1.512	3.923	0.06551–235.0	0.81
			Odd	1.338	1.243–1.441	9.486	0.7658–117.5	0.97
		Ensemble	Even	1.318	1.306–1.329	7.319	5.450–9.829	0.61
			Odd	1.336	1.325–1.347	8.477	6.473–11.10	0.86

TABLE II: TTS for the exact solution, first-, and second-excited energy. Estimated base value b and constant c controlling the exponential scaling in Eq. (5), and associated R^2 values for the TTS linear fit across four algorithms (QAOA, Tabu Search, PCE, and PCE + Tabu) for even and odd instances.

of the number of measurement shots. At the circuit depth that we tested, the effect of barren plateaus should be modest. Nevertheless, we observe a noticeable change in relative error between the first and the last optimization epoch. This effect is primarily due to the random initialization of the circuit parameters: at start, the state produces near-zero Pauli expectation values. In this regime, hardware noise and finite-shot fluctuations contribute a comparatively larger fraction of the measured value, thereby amplifying the relative error. As optimization progresses and the expectation values move away from zero, the signal-to-noise ratio improves and the relative error decreases accordingly. Furthermore, we simulated the full optimization procedure assuming an average relative error on the loss function equal to the one we observed experimentally. The result-

ing performance degradation is reported in the insets. We observe that shot-noise alone causes only a marginal loss in solution quality, suggesting that a few thousand measurement shots already provide sufficient statistics for experimental training. In contrast, hardware noise does not prevent training, but it noticeably degrades the final solution quality, potentially eroding the expected scaling advantage in time-to-solution (TTS). On current QPUs, this can be addressed with polynomial-time quantum error-mitigation techniques. Alternatively, increasing the encoding compression rate could further reduce quantum resource requirements and exposure to noise, although this may introduce a trade-off with scaling performance. Overall, the results indicate a practical path forward: shot budgets are already manageable, but mitigating hardware noise

is the crucial factor to preserve the observed advantage. Finally, in the last optimization epoch, in the rounding step that maps the relaxed experimental solution back to binary values, we observe a bitstring error rate of only 1.6%, indicating that the hardware error does not meaningfully affect the final solution quality.

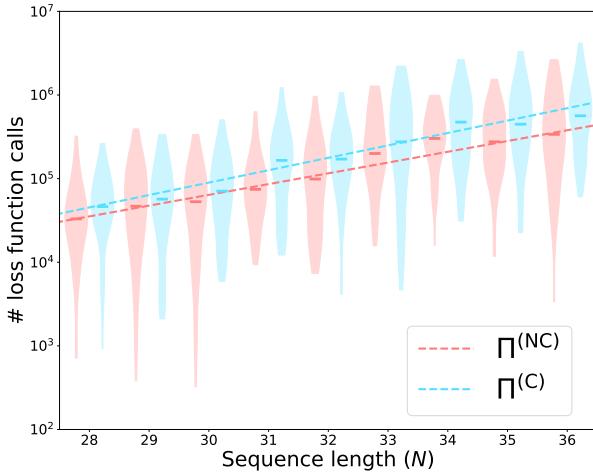


FIG. 8: $\Pi^{(C)}$ vs $\Pi^{(NC)}$. Scaling comparison of the TTS for the quantum LABS solver based on Pauli Correlation Encoding using pauli sets chosen from $\Pi^{(NC)}$ (red) vs. pauli sets chosen $\Pi^{(C)}$ (blue) for values of $N \in [28, 36]$. Here, TTS is defined as the total number of cost function evaluations required to reach the exact solution [7]. Dashed-dotted lines denote linear regression over the list of median values for each N . The distribution of points over different initializations is depicted by violin plots at each N . The TTS distributions of $\Pi^{(C)}$ almost always dominate over the distributions of $\Pi^{(NC)}$. Furthermore, the fits show a exponential scaling difference between the two approach, with a value of b of $\mathcal{O}(1.345^N)$ for $\Pi^{(NC)}$ against $\mathcal{O}(1.407^N)$ for $\Pi^{(C)}$

6. Building the Pauli sets $\Pi^{(C)}$ and $\Pi^{(NC)}$

The two Pauli sets are randomly constructed as follows. First, we split the $4^n - 1$ traceless Pauli strings in the Pauli group into subsets of maximally commuting operators using the procedure proposed in Ref. [21]. There are $2^n + 1$ such sets, each having $2^n - 1$ mutually commuting Paulis. Then, for each LABS instance size N and for each optimization run, we repeatedly pick one such set and one Pauli string P from it at random and accept or reject P according to the following criteria:

1. for $\Pi^{(C)}$: accept P if and only if it commutes with every other previously selected Pauli. Repeating

this eventually leads to a set of N mutually commuting Paulis, as long as $N < 2^n$;

2. for $\Pi^{(NC)}$: accept P if and only if it anti-commutes with every other previously selected Pauli until reaching the maximum possible number $2n + 1$ [34]). After that, since it's impossible to anti-commute with all the others, accept those that anti-commute with the maximum possible number of selected Paulis.

In Fig. 8, we compare the performance of both sets and observed that $\Pi^{(NC)}$ led to better TTS scaling and chose it as a figure of merit for the benchmark against other solvers.

7. Comparison to State-of-the-Art Quantum Methods

For small instance sizes, the estimated time-to-solution (TTS) can vary a lot depending on which instance sizes are considered. To make a fair comparison, we report our method's performance on the same instances interval used for two state-of-the-art quantum methods Ref [23] and Ref [4]. We also compare against tabu search. Furthermore, we investigate whether our solver can provide warm-start solutions to the memetic tabu search, following an analogous approach in Ref [23]. Specifically, we take the best solution obtained across 150 independent PCE runs and copy it 50 times to initialize the population of the memetic tabu search. All hyperparameters of the classical algorithm are kept the same as in [23]. For the time-to-solution (TTS) estimate, we include all loss-function queries used during the 150 PCE optimization runs, in addition to those performed by the memetic tabu search, all using the same bootstrap procedure detailed in [23] to estimate the errors. The results are reported in Table III

8. Shot complexity

Given $\varepsilon, \delta > 0$, and a vector $\boldsymbol{\theta}$ of variational parameters, consider the problem of estimating \mathcal{L} up to additive precision ε and with statistical confidence $1 - \delta$. For each Pauli correlator $\langle \Pi_i \rangle, i \in [N]$, assume that, with confidence $1 - \tilde{\delta}$, one has an unbiased estimator Π_i^* with statistical error at most $\eta > 0$, *i.e.*,

$$|\Delta\langle \Pi_i \rangle| := |\langle \Pi_i \rangle - \Pi_i^*| \leq \eta. \quad (7)$$

The corresponding error in the loss function is $\Delta\mathcal{L} := \mathcal{L} - \mathcal{L}^*$, with \mathcal{L}^* given by (4) calculated using

Heuristic	Type	TTS	CI	R^2
Tabu Search	Median	1.42	1.26-1.60	0.83
	Ensemble	1.45	1.41-1.49	0.43
PCE	Median	1.34	1.29-1.40	0.97
	Ensemble	1.33	1.29-1.37	0.35
PCE + Mem. Tabu	Median	1.29	1.27-1.30	0.88
QE-MTS [23]	Median	1.24	1.23-1.25	0.88

TABLE III: Estimated base value b , controlling the exponential scaling $\mathcal{O}(b^N)$, and associated R^2 values for the TTS linear fit across the considered heuristics, reported for two instance-length ranges : $L \in [27, 38]$ on left and $L \in [28, 39]$ on the right. The two selected instance intervals match those used to benchmark the two state-of-the-art quantum methods.

Π_i^* instead of $\langle \Pi_i \rangle$. The multivariate Taylor theorem ensures that there is a $\xi \in [-1, 1]^N$ such that

$$\Delta \mathcal{L} = \sum_{i \in [N]} \frac{\partial \mathcal{L}}{\partial u_i} \Big|_{u_i=\xi} \Delta \langle \Pi_i \rangle. \quad (8)$$

For the loss function (4), using $\tanh(x) \leq 1$ and $\frac{d}{dx} \tanh(x) = \text{sech}^2(x) \leq 1$, one can show that $\left| \frac{\partial \mathcal{L}}{\partial u_i} \right|_{\xi_i} \leq 2\alpha[N(N-1) + \beta]$. As a result,

$$|\Delta \mathcal{L}| \leq 2\eta\alpha N[N(N-1) + \beta] \quad (9)$$

follows immediately from the triangle inequality together with (7). To ensure $|\Delta \mathcal{L}| \leq \varepsilon$ we then require that

$$\eta \leq \frac{\varepsilon}{2\alpha[N(N-1) + \beta]}. \quad (10)$$

The minimum number S of samples needed to achieve such precision can be upper-bounded by standard arguments using the union bound and Hoeffding's inequality, which gives $S \leq (2/\eta^2) \log(2N/\delta)$. Then, by virtue of Eq. (10), it suffices to take

$$S \geq \frac{8\alpha^2 N^2}{\varepsilon^2} [N(N-1) + \beta]^2 \log\left(\frac{2N}{\delta}\right). \quad (11)$$

This is the general form of our upper bound and guarantees $P(|\Delta \mathcal{L}| \leq \varepsilon) \geq 1-\delta$. Recall from App. 3 that $\alpha = 1.5n = \mathcal{O}(\sqrt{N})$. We also reiterate that this a loose upper bound and, in practice, the actual number of samples is expected to have a better scaling in N .

9. Quantum-classical crossover point

In our numerical results, we observed a competitive TTS scaling compared to Tabu Search in terms of query access to the loss function. However, each quantum query incurs an overhead given by gate

Heuristic	Type	TTS	CI	R^2
Tabu Search	Median	1.44	1.29-1.62	0.86
	Ensemble	1.50	1.46-1.54	0.49
PCE	Median	1.37	1.32-1.43	0.97
	Ensemble	1.35	1.31-1.39	0.37
PCE + Mem. Tabu	Median	1.31	1.29-1.32	0.92
QAOA [4]	Median	1.46	1.42-1.50	0.94
QAOA+QMFI [4]	Median	1.21	1.19-1.23	—

operations and the sample complexity to guarantee that all expectation values are estimated within desired precision. Furthermore, our approach also need an estimation of the gradient, usually done in the context of VQA using parameter-shift-rule [35]. Assuming the observed TTS scalings remain the same for larger instances, we can use the circuit and sample complexity estimates in Apps. 2 and 8 to determine the crossover point in which we expect to observe a quantum advantage in query complexity. Our estimation in Tab. IV considers the scaling obtained for the the median of the even instances, as it has a better value of the R^2 , and the crossing points for advantage are given in terms of query complexity and walltime for two different quantum hardware platforms: superconducting (SC) and trapped-ion (TI) qubits. For superconducting qubits, we take each layer to be executed in parallel, while, for

Solution	QPU	Adv. Type	N_*
Exact	SC	Walltime	1175
		Queries	1099
	TI	Walltime	1431
		Queries	1207
1st	SC	Walltime	3705
		Queries	3469
	TI	Walltime	4407
		Queries	3763
2nd	SC	Walltime	2597
		Queries	2433
	TI	Walltime	3107
		Queries	2645

TABLE IV: **Crossover points.** Estimated instance size N_* after which the PCE solver is expected to outperform the classical Tabu search solver in terms of total walltime or total number of queries to the loss function for two different quantum hardware platforms: superconducting qubits (SC) and trapped ion (TI) qubits.

trapped-ions, each gate is considered as executed sequentially. For the estimation of the walltime, we assumed an overhead of 5×10^2 for superconducting qubits [36], and 6×10^6 for trapped ion [37], when comparing to a cycle of a current day CPU[38]. In addition, we disregarded any extra overhead due to error mitigation that the quantum approach may need. This was done both because a detailed estimation would be heavily platform dependent, and because the actual impact that error noise have in the training of our specific VQA (and so the level of error mitigation actually needed) is still an open question. While the instance sizes in Tab. IV indicate the expected scaling crossover between the two methods, PCE also shows gains in producing high-quality ap-

proximate solutions. This suggests that PCE may outperform the classical baseline at substantially smaller sizes, especially given Tabu Search’s sharp performance degradation for instances of a few hundred variables. Furthermore, as also observed in [10], our shot-complexity estimate is an upper bound and is likely conservative. In practice, we expect fewer circuit executions to suffice, which would lower the crossover point accordingly.

Finally, the algorithm relies on many circuit re-initializations but requires only a modest number of qubits. This makes it naturally suited to massively parallel execution across many small QPUs, potentially further reducing threshold needed to observe a walltime advantage.