Quantum Semidefinite Programming with the Hadamard Test and Approximate Amplitude Constraints

Taylor L. Patti[1,2], Jean Kossaifi[2], Anima Anandkumar[3,2], and Susanne F. Yelin[1]

[1]Department of Physics, Harvard University, Cambridge, Massachusetts 02138, USA

[2]NVIDIA, Santa Clara, California 95051, USA

[3]Department of Computing + Mathematical Sciences (CMS), California Institute of Technology (Caltech), Pasadena, CA 91125 USA

Semidefinite programs are optimization methods with a wide array of applications, such as approximating difficult combinatorial problems. We introduce a variational quantum algorithm for semidefinite programs that uses only n qubits, a constant number of circuit preparations, and $O(n^2)$ expectation values in order to solve semidefinite programs with up to $N = 2^n$ variables and $M = 2^n$ constraints. Efficient optimization is achieved by encoding the objective matrix as a properly parameterized unitary conditioned on an auxilary qubit, a technique known as the Hadamard Test. The Hadamard Test enables us to optimize the objective function by estimating only a single expectation value of the ancilla qubit, rather than separately estimating exponentially many expectation values. Similarly, we illustrate that the semidefinite programming constraints can be effectively enforced by implementing a second Hadamard Test, as well as imposing $\sim n^2/2$ Pauli string amplitude constraints. We demonstrate the effectiveness of our protocol by devising an efficient quantum implementation of the Goemans-Williamson algorithm, which is a useful approximation for various NP-hard problems, such as MaxCut. Our method exceeds the performance of analogous classical methods on a diverse subset of well-studied MaxCut problems from the GSet library.

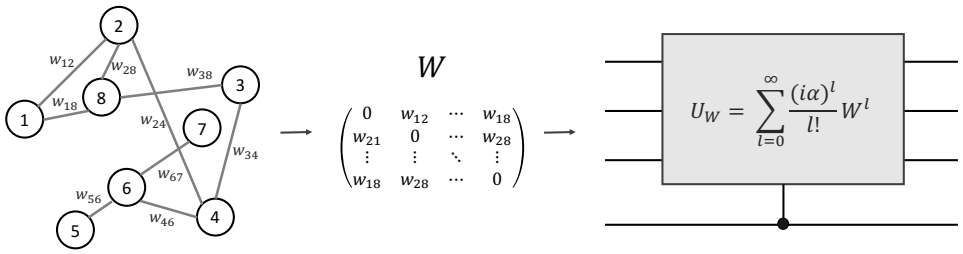
Taylor L. Patti: taylorpatti@g.harvard.edu

1 Introduction

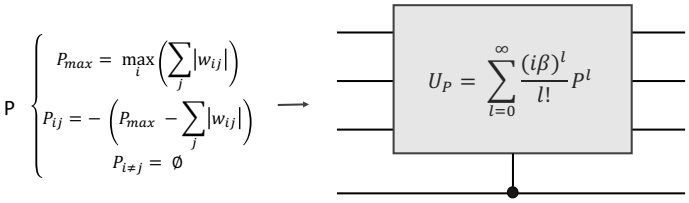
Semidefinite programming (SDP) is a variety of convex programming wherein the objective function is extremized over the set of symmetric positive semidefinite matrices $\mathbb{S}^+$ [1]. Typically, an N-variable extremization problem is upgraded to an optimization over N vectors of length N, which form the semidefinite matrices of $\mathbb{S}^+$. A versatile technique, SDP can be used to approximately solve a variety of problems, including combinatorial optimization problems (e.g., NP-hard problems, whose computational complexity grows exponentially in problem size) [2], and is heavily used in fields such as operations research, computer hardware design, and networking [3, 4]. In many such cases, semidefinite programs (SDPs) are integer programming relaxations, meaning that the original objective function of integer variables is reformed as a function of continuous vector variables [5]. This allows the SDP to explore a convex approximation of the problem. Although such solutions are only approximate, SDPs are useful because they can be efficiently solved with a variety of techniques. These include interior-point methods, which typically run in polynomial-time in the number of problem variables N and constraints M [6].

An additional advantage of optimization with SDPs is that many have performance guarantees in the form of approximation ratios. Approximation ratios are a provable worst-case ratio between the value obtained by an approximation algorithm and the ground truth global optimum [7]. In short, SDPs represent an often favorable compromise between computational complexity and solution quality.

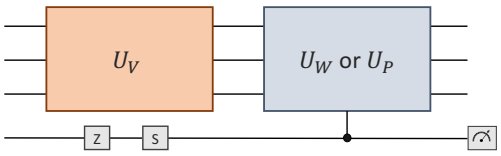
Despite the favorable scaling of classical SDPs, they still become intractable for high-dimensional



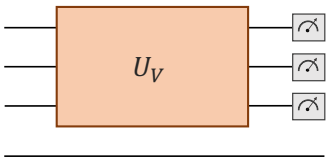
(a) **Controlled-unitary objective function**



(b) **Controlled-unitary population balancing**



(c) **Estimating Hadamard Test loss function terms**



(d) **Estimating Pauli string amplitude constraints**

Figure 1: **Diagram of HTAAC-QSDP** with $n = 3$ non-auxiliary qubits. (1a) A classical problem of N variables (here an N-vertex MaxCut problem where $N = 8$). The weight matrix W is used to generate the unitary U_W, which is rotated about the angle α and implemented as a controlled-unitary conditioned on the $n+1$th-qubit (auxilary qubit). (1b) The population-balancing unitary U_P is generated by the diagonal matrix P, which offsets the asymmetric edge weights on certain vertices in proportion to some constant β. (1c) The Hadamard Test is used to efficiently evaluate the objective function and population balancing constraints. The n-qubit state $|\psi\rangle = U_V|\mathbf{0}\rangle$ is prepared with a variational quantum circuit U_V, and the $n+1$th (auxilary) qubit is initialized as $(|0\rangle - i|1\rangle)/\sqrt{2}$. Subsequently, the Hadamard Test is carried out: U_W or U_P is implemented as a controlled-unitary conditioned on the auxilary qubit, which is then measured to compute $\langle \sigma_{n+1} \rangle_{W,t} = \mathrm{Im}[\langle \psi | U_W | \psi \rangle]$ or $\langle \sigma_{n+1} \rangle_{P,t} = \mathrm{Im}[\langle \psi | U_P | \psi \rangle]$. (1d) The $M = 2^n$ SDP amplitude constraints constraints are approximately enforced with only $m \sim n^2/2$ Pauli string constraints (Eq. 12). These are computed by collecting n-qubit Pauli-z measurements and using marginal statistics to estimate the m expectation values.

problems. A variety of quantum SDP algorithms (QSDPs) that sample n-qubit Gibbs states to solve SDPs with up to $N = 2^n$ variables and $M = 2^n$ constraints have been devised [15–19] (Table 1), as have methods for approximately preparing Gibbs states with near-term variational quantum computers [21–23]. The former of these algorithms are based on the Arora-Kale method [24] and provide up to a quadratic speedup in N and M. However, they scale significantly poorer in terms of various algorithm parameters, such as accuracy, and are not suitable for near-term quantum computers. Quantum interior-point methods have also been proposed [25, 26], in close analogy to the leading family of classical techniques.

Variational methods have long played a role in quantum optimization protocols [27] (Table 1), such as adiabatic computation [8–10], annealing [11, 12], the Quantum Approximate Optimization Algorithm (QAOA) [13], and Boson Sampling [14]. However, only recently have variational QSDPs been proposed [20, 28]. Patel et al [20] addresses the same optimization prob-

Table 1: Comparison of common quantum methods for classical optimization. The number of potential variables N and constraints M are given in terms of qubits n. Whether or not the method provides guarantees on its solutions is discussed, as is its suitability for near-term quantum devices. Our Hadmard Test objective function and Approximate Amplitude Constraint Quantum SDP (HTAAC-QSDP, Fig. 1) ensures SDP approximation ratios, is suitable for near-term variational quantum devices, and provides efficient objective function evaluation (via the Hadamard Test) and constraints (via a second Hadamard Test and $O(n^2)$ Pauli string constraints).

Method	N, M **Scaling**	**Solution Guarantee**	**Near-Term Devices**
Quantum Adiabatic [8–10]	n	If Infinitely Slow	Sometimes
Quantum Annealing [11, 12]	n	No	Yes
QAOA [13]	n	Sometimes	Yes
Boson Sampling [14]	n	No	Yes
QSDPs [15–19]	2^n	SDP Approx. Ratios	No
Variational QSDPs [20]	2^n	SDP Approx. Ratios	Yes, $O(2^n)$ exp. vals./epoch
HTAAC-QSDP (Ours)	2^n	SDP Approx. Ratios	Yes, $O(n^2)$ exp. vals./epoch

lems as the quantum Arora-Kale and interior-point based methods, but instead uses variational quantum circuits, which are more feasible in the near-term. Like other SDPs, this method offers specific performance guarantees in the form of approximation ratios [7]. However, the proposed circuit optimization in the method relies on full computational-basis tomography of a variationally prepared N-state wavefunction $|\psi\rangle$, which implies the estimation of $O(2^n)$ observables for each training epoch.

Our Approach: We propose a new variational quantum algorithm for solving QSDPs that uses Hadamard Test objective functions and Approximate Amplitude Constraints (HTAAC-QSDP, Fig. 1). HTAAC-QSDP uses $n+1$ qubits, a constaint number of quantum measurements, and $O(n^2)$ classical calculations to solve SDPs with up to $N = 2^n$ variables, a nearly exponential reduction in required computation compared to similar quantum techniques [20]. As described in Sec. 2, we achieve this, in part, through a unitary objective function encoding with the Hadamard Test [29] (Fig. 1a), which allows for the extremization of the entire N-dimensional objective by estimating only a single quantum expectation value (Fig. 1c). As a concrete example of HTAAC-QSDP, we implement the Goemans-Williamson algorithm (Algorithm 1) [30]. This SDP requires $M = N = 2^n$ amplitude constraints, which we effectively enforce with only 1) a constant number of quantum measurements from a second Hadamard Test (Fig. 1b) and 2) the estimation of a polynomial number $m \sim n^2/2$ of properly selected, commuting Pauli strings (Fig. 1d).

In Sec. 3, we demonstrate the success of the HTAAC-QSDP Goemans-Williamson algorithm (Algorithm 1) by solving MaxCut [31] for large-scale graphs from the well-studied GSet graph library [32] (Fig. 5). In addition to satisfying the 0.878 MaxCut approximation ratio [30], HTAAC-QSDP achieves cut values that are commensurate with the leading gradient-based classical SDP solvers [33], implying that we reach the global optima of these SDP objective functions.

Finally, in Sec. 4 we establish a lower bound (Theorem 1) on the phase α of our Hadamard encoding, such that our technique is a high-quality approximation of the original SDP. The purpose of this lower bound is to demonstrate that tractably large values of the unitary phase α are permissible (i.e., α need not become arbitrarily small) for encoding a wide variety of useful and large-scale graph optimization problems. Specifically:

Theorem 1 *Our approximate Hadamard Test objective function $U_W \sim i\alpha W$ (Sec. 2.1) holds for graphs with randomly distributed edges if*

$$\alpha^2 \lesssim \frac{N^4}{e^3} = \frac{N}{\xi^3},$$

where e is the number of non-zero edge weights and ξ is the average number of edges per vertex.

We can view the criteria of Theorem 1 in two ways: for SDPs with arbitrarily many variables N, the size of α can be kept reasonably large while the Hadamard Test objective function (see Sec. 2.1) remains valid as long as 1) N does not

Algorithm 1 Goemans-Williamson Algorithm for MaxCut with HTAAC-QSDP

Input: Optimization matrix W, Hadamard Test coefficients α, β, learning rate η (and additional optimization hyperparameters, e.g., β_1 and β_2 with ADAM), constraint magnitude λ, number of epochs T.

Randomly initialize the variational quantum circuit $U_V(1)$
for t in **range**(T) **do**
  $|\psi_t\rangle \otimes |0\rangle = U_V(t)|\mathbf{0}\rangle \otimes |0\rangle$
  Hadamard Test (phase α) $U_W \rightarrow \langle\sigma_{n+1}^z\rangle_{W,t}$
  Hadamard Test (phase β) $U_P \rightarrow \langle\sigma_{n+1}^z\rangle_{P,t}$
  Measure $\Pi_{i=1}^n \sigma_i^z \rightarrow \langle\sigma_j^z, \rho_t\rangle^2, \langle\sigma_j^z\sigma_k^z, \rho_t\rangle^2$
  $\mathcal{L}(t) = \langle\sigma_{n+1}\rangle_{W,t} + \langle\sigma_{n+1}\rangle_{P,t} + \lambda(\sum_j \langle\sigma_j^z, \rho_t\rangle^2 + \sum_{k\neq j}\langle\sigma_j^z\sigma_k^z, \rho_t\rangle^2)$
  Backprop $\eta\nabla\mathcal{L}(t)$: $U_V(t) \rightarrow U_V(t+1)$

Output: SDP MaxCut vertex solution $|\psi_T\rangle$, $\text{sign}(|\psi_T\rangle) = \left[v_1, ..., v_N\right]^T$.

grow slower than the total number of edges e, or 2) N does not grow slower than the the cube of ξ. Both of these conditions hold for graphs that are not too dense, meaning that they are widely satisfiable because the majority of interesting and demonstrably difficult graphs for MaxCut are relatively sparse [7, 31, 32, 34, 35]. We note that for graphs where edge-density is unevenly distributed, Theorem 1 should hold for the densest region of the graph, i.e., ξ should be the average number of edges per vertex for the most highly connected vertices.

2 Efficient Quantum Semidefinite Programs

The standard form of an N-variable, M-constraint SDP is [1, 2]

$$\begin{aligned} &\text{minimize}_{X\in\mathbb{S}^+} \ \langle W, X\rangle \\ &\text{subject to } \langle A_\mu, X\rangle = b_\mu, \ \ \forall\mu \leq M \\ &X \succeq 0, \end{aligned} \tag{1}$$

where W is an $N \times N$ symmetric matrix that encodes the optimization problem and A_μ (b_μ) are $N \times N$ symmetric matrices (scalars) that encode the problem constraints. $\langle A, B\rangle$ denotes the trace inner product

$$\langle A, B\rangle = \text{Tr}\left[A^T B\right] = \sum_{i,j}^{N} A_{ij}B_{ij}. \tag{2}$$

In this section, we detail a method of efficient optimization of the above SDP objective and constraints using quantum methods (Fig. 1), specifically by implementing Hadamard Tests and imposing a polynomial number of Pauli constraints. We provide a concrete example in the form of the Goemans-Williamson [30] algorithm for MaxCut [31], as summarized in Algorithm 1.

2.1 The Hadamard Test as a Unitary Objective

In quantum analogy to the objective function of Eq. 1, we wish to minimize $\langle W, X\rangle$ over the n-qubit density matrices $\rho = |\psi\rangle\langle\psi|$, which are positive semidefinite by definition. We define quantum states $|\psi\rangle = U_V|\mathbf{0}\rangle$, where U_V is a variational quantum circuit and $|\mathbf{0}\rangle = \Pi_{i=1}^n|0\rangle$ is a trivial input state in the computational basis. This yields the quantum objective function

$$\text{minimize } \langle W, \rho\rangle = \text{minimize } \langle\psi|W|\psi\rangle. \tag{3}$$

The Hadamard Test (Fig. 1c) is a quantum computing subroutine for arbitrary n-qubit states $|\psi\rangle$ and n-qubit unitaries U [29]. It allows the real or imaginary component of the 2^n-state inner product $\langle\psi|U|\psi\rangle$ to be obtained by estimating only a single expectation value $\langle\sigma_{n+1}^z\rangle$, which is the z-axis Pauli spin on the n+1th (auxiliary) qubit. For example, to obtain the imaginary component of $\langle\psi|U|\psi\rangle$, we prepare the quantum state $|\psi\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$ and apply a controlled-U from the n+1th qubit to $|\psi\rangle$, followed by a Hadamard gate on the n+1th qubit. This produces the state

$$\frac{1}{2}\left[(I - iU)|\psi\rangle \otimes |0\rangle + (I + iU)|\psi\rangle \otimes |1\rangle\right] \tag{4}$$

upon which projective measurement yields

$$\langle \sigma^z_{n+1} \rangle = \mathrm{Im}\left[\langle \psi | U | \psi \rangle\right]. \tag{5}$$

Rather than estimate the 2^n expectation values required to characterize ρ and optimize the loss function of Eq. 3, our method efficiently encodes the N-dimensional objective matrix W as the imaginary part of an n-qubit unitary $U_W = \exp(i\alpha W)$ (Fig. 1a). Here, the phase α is a constant scalar. U_W is then conditioned on the $n{+}1$th (or auxilary) qubit as a controlled-unitary. We then use the Hadamard Test to calculate the objective term in the loss function

$$\langle \sigma^z_{n+1} \rangle_W = \mathrm{Im}\left[\langle \psi | U_W | \psi \rangle\right] = \mathrm{Im}\left[\langle U_W, \rho \rangle\right]. \tag{6}$$

The intuition for this objective function is that, for sufficiently small α, $\mathrm{Im}[U_W] \approx \alpha W$. By restricting ourselves to quantum circuits with real-valued output states, we render the single expectation value $\langle \sigma^z_{n+1} \rangle_W$ proportional to the objective function of Eq. 3, which requires N expectation values to estimate. In Sec. 4, we analytically prove that, for many optimization problems, α has a practical lower bound such that $\mathrm{Im}[U_W] \approx \alpha W$ with a reasonably large α, even for arbitrarily large W.

2.2 Quantum Goemans-Williamson Algorithm

We now illustrate how $\mathrm{Im}[U_W]$ can be a close approximation of αW, including for optimization problems with an arbitrarily large number of variables N. For concreteness, we select the NP-complete problem MaxCut [31], and specifically focus on the corresponding NP-hard optimization problem [37]. This problem is of particular interest due to its favorable 0.878-approximation ratio with semidefinite programming techniques, notably the Goemans-Williamson algorithm [30], for which we now derive an efficient quantum implementation. The Goemans-Williamson algorithm is also applicable to to numerous other optimization problems, such as MaxSat and Max Directed Cut [30].

For a MaxCut problem with N vertices v_i, v_j, let W be the matrix that encodes the up to $N(N-1)/2$ non-zero edge weights in its entries W_{ij}. As the vertices lack self-interaction, $W_{ii} = 0$. The optimization problem is then defined as

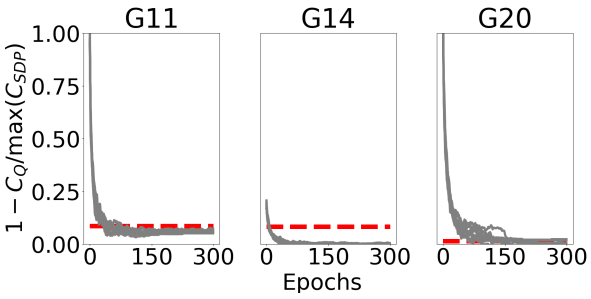


Figure 2: The cut values C_Q obtained by HTAAC-QSDP with order-$k \leq 2$ Pauli constraints compared to $\max(C_{\mathrm{SDP}})$, the best results of classical gradient-based SDPs (specifically, interior points methods) [33]. Our performance on the skewed binary and integer graphs narrowly exceeds that of the classical method ($\max(C_{\mathrm{SDP}}) < C_Q$), while the classical method narrowly outperforms our quantum method for the toroid graphs ($\max(C_{\mathrm{SDP}}) > C_Q$). Overall, the performance of our HTAAC-QSDP implementation and its classical counterpart are commensurate. HTAAC-QSDP exceeds the $C_Q/C_{\mathrm{MAX}} > 0.878$ MaxCut approximation ratio (red dashed line) for all graphs, where C_{MAX} is the true MaxCut of the graph. In this work, we assume C_{MAX} as the largest-known cuts of the GSet graphs, which were obtained from intensive and repeated heuristic searches [36].

$$\begin{aligned} &\text{maximize} \sum_{j \neq i} W_{ij} \frac{1 - v_i v_j}{2} \\ &\text{subject to } v_i = \pm 1, \end{aligned} \tag{7}$$

which can be mapped to the classical SDP with $M = N$ constraints

$$\begin{aligned} &\text{minimize}_{X \in \mathbb{S}^+} \ \langle W, X \rangle \\ &\text{subject to } \ X_{ii} = 1, \ \ \forall i \leq N. \end{aligned} \tag{8}$$

As described by Eq. 3, we can transform the optimization portion of Eq. 8 by substituting the classical positive semidefinite matrix X for the quantum density operator ρ. The solution to the SDP is then stored in $|\psi\rangle$, i.e., $v_i = \mathrm{sign}(\psi_i)$ (for more details, see Sec. 2.3). As detailed in Sec. 2.1, the evaluation of this objective function can be optimized by estimating a single expectation value with the Hadamard Test. Likewise, we now introduce an efficient quantum alternative to the constraint $X_{ii} = 1$ from Eq. 8. First, note that due to the orthonormality of quantum states, the exact quantum equivalent of Eq. 8 is

$$\rho_{ii} = 1/2^n = N^{-1}. \tag{9}$$

Table 2: MaxCut statistics for all 800-vertex graphs studied by the leading gradient-based classical SDP (interior points) method [33]. The highest known MaxCut values (C_{MAX}, found by intensive heursitics [36]) are greater the highest results obtained by the classical method ($\max(C_{\mathsf{SDP}})$), but the approximation ratio $\max(C_{\mathsf{SDP}})/C_{\mathsf{MAX}} > 0.878$ is satisfied. The largest cut values of our HTAAC-QSDP method ($\max(C_Q)$) are comparable with $\max(C_{\mathsf{SDP}})$, as are the average results ($\mathrm{mean}(C_Q)$).

Graph	$C_{\mathbf{MAX}}$	$\mathbf{max}(C_{\mathbf{SDP}})$	$\mathbf{max}(C_Q)/\mathbf{max}(C_{\mathbf{SDP}})$	$\mathbf{mean}(C_Q)/\mathbf{max}(C_{\mathbf{SDP}})$
G11	564	542	0.967	0.940
G12	556	540	0.982	0.953
G13	582	564	0.972	0.933
G14	3064	2922	1.011	1.000
G15	3050	2938	1.009	0.996
G20	941	838	1.007	0.983
G21	931	841	1.001	0.978

This rescaling changes neither the effectiveness nor the guarantees of the semidefinite program, because the salient feature of the constraint is that all of the quantum states have the same amplitude magnitude $|\psi_i|$, such that all of the vertices are of equal magnitude and none are disproportionately favored. The solutions ρ and X differ only by a constant factor, such that $\rho = X/N$. This yields the quantum MaxCut SDP

$$\begin{aligned} &\text{minimize } \langle W, \rho \rangle \\ &\text{subject to } \rho_{ii} = N^{-1}, \ \ \forall i \leq N. \end{aligned} \tag{10}$$

As graph weights are real-valued and symmetric (i.e., $W_{ij} = W_{ji}$), W is Hermitian. We can thus use it as the generator of U_W such that (Fig. 1a)

$$\begin{aligned} U_W &= \exp(i\alpha W) = \sum_l \frac{(i\alpha)^l}{l!} W^l \\ &= I + \frac{i\alpha}{1!}W - \frac{\alpha^2}{2!}W^2 - \frac{i\alpha^3}{3!}W^3 + \mathcal{O}(W^4). \end{aligned} \tag{11}$$

As W is real, the odd powers of l in Eq. 11 are the imaginary components. The condition that $\mathrm{Im}[U_W] \propto W$ is upheld *iff*, for the vast majority of variables i, j, $\alpha W_{ij} \gg \frac{\alpha^3}{6}(W^3)_{ij}$. In Sec. 4, we prove Theorem 1, demonstrating that this condition is achievable with a tractable α (i.e., α larger than some fixed finite value that is constant in problem size N) for a wide variety of graphs.

Next, we note that enforcing the $M = N = 2^n$ amplitude constraints $\rho_{ii} = N^{-1}$, $i \leq N$ would require the estimation of all z-axis Pauli strings of order $k \leq n$ (all Pauli strings with $k \leq n$ Pauli-z operators) of the state $|\psi\rangle$. This would be a total of $N - 1$ expectation values. As an alternative to this large overhead, HTAAC-QSDP proposes the use of *Approximate Amplitude Constraints* (Fig. 1d). For example, consider the set of $m = n(n-1)/2 + n \sim n^2/2$ Pauli strings of length $k \leq 2$

$$\begin{aligned} &\langle \sigma_a^z, \rho \rangle = 0, \ \ \forall a \leq n \\ &\langle \sigma_a^z \sigma_b^z, \rho \rangle = 0, \ \ \forall b \neq a, \ \ a, b \leq n, \end{aligned} \tag{12}$$

as constraints for the n-qubit output state $|\psi\rangle$. This set of $m \sim n^2/2$ constraints approximates the same restrictions as the set of $M = N$ constraints of Eq. 10 by limiting quantum correlations, as these generally correspond to unequal state populations. The $k = 1$ (i.e., $\langle \sigma_a^z, \rho \rangle = 0$) constraints ensure that, up to some phase, each qubit is in an equal superposition of $|0\rangle$ and $|1\rangle$, without which all 2^n state components cannot have amplitudes of equal magnitudes. Likewise, the $k = 2$ (i.e., $\langle \sigma_a^z \sigma_b^z, \rho \rangle = 0$) constraints prevent 2-qubit amplitude correlations that would satisfy the $k = 1$ terms yet underpopulate some state components. For example, the $k = 2$ constraint $\langle \sigma_1^z \sigma_2^z, \rho \rangle = 0$ is needed to disallow the Bell State $(|00\rangle + |11\rangle)/\sqrt{2}$, which satisfies the $k = 1$ constraints $\langle \sigma_1^z, \rho \rangle = 0$ and $\langle \sigma_2^z, \rho \rangle = 0$, but has zero amplitude for two of four total states. The Pauli string constraints of Eq. 12 are commuting, such that they can be estimated as m different marginal distributions from a single set of n-qubit z-axis measurements.

We again emphasize that these $k \leq 2$ constraints only approximately enforce the SDP con-

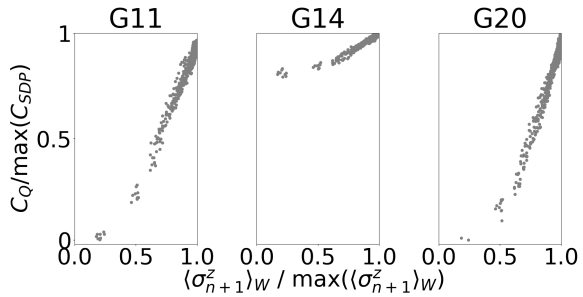


Figure 3: The cut value C_Q obtained by HTAAC-QSDP vs its SDP objective $\langle\sigma^z_{n+1}\rangle_W$ for the G11 (toroid), G14 (binary), and G20 (integer) graphs. As with classical SDP methods, low loss function values are correlated with high cut values. The strong correlation between optimized loss and resultant cut illustrates the convergence of HTAAC-QSDP despite its approximate nature.

straint $\rho_{ii} = N^{-1}$. Fully satisfying this constraint would require restricting Pauli-z correlations between *any* subset of the n qubits, such that no states of unequal amplitude magnitudes are permitted. Eq. 9 can be fully satisfied if we constrain $|\psi\rangle$ with all of the Pauli strings of length $k \leq n$. However, as there exist n choose k z-axis Pauli strings of order k, this requires estimating $\sum_{k=1}^{n} \binom{n}{k} = 2^n - 1$ different expectation values and greatly decreases the efficiency of the algorithm. Sec. 3 details that, in practice, competitive results are obtained using only $k \leq 2$ constraint terms (Fig. 2 and Table 2), and optimization performance is largely saturated with terms $k \leq 4$ (Fig. 4 and Table 3).

In order to explicitly see how the $m \sim n^2/2$ constraints of Eq. 12 largely enforce the constraint $\rho_{ii} = 1/2^n$, let us take the example of a three-qubit state ($n = 3$), which can encode up to eight vertices ($N = 2^n = 8$) using HTAAC-QSDP. Any real-valued $n = 3$ state can be written generically as

$$|\psi\rangle = \sum_{r,s,p=0}^{1} \psi_{rsp}|\psi_{rsp}\rangle$$

and its constraints from Eq. 12 are

$$\begin{aligned}
\langle\sigma^z_1,\rho\rangle &= \sum_{r,s}|\psi_{0rs}|^2 - \sum_{r,s}|\psi_{1rs}|^2 = 0\\
\langle\sigma^z_2,\rho\rangle &= \sum_{r,s}|\psi_{r0s}|^2 - \sum_{r,s}|\psi_{r1s}|^2 = 0\\
\langle\sigma^z_3,\rho\rangle &= \sum_{r,s}|\psi_{rs0}|^2 - \sum_{r,s}|\psi_{rs1}|^2 = 0\\
\langle\sigma^z_1\sigma^z_2,\rho\rangle &= \sum_{p}(\sum_{s=r}|\psi_{rsp}|^2 - \sum_{s\neq r}|\psi_{rsp}|^2) = 0\\
\langle\sigma^z_1\sigma^z_3,\rho\rangle &= \sum_{p}(\sum_{s=r}|\psi_{rps}|^2 - \sum_{s\neq r}|\psi_{rps}|^2) = 0\\
\langle\sigma^z_1\sigma^z_2,\rho\rangle &= \sum_{p}(\sum_{s=r}|\psi_{prs}|^2 - \sum_{s\neq r}|\psi_{prs}|^2) = 0.
\end{aligned} \tag{13}$$

Combined with the normalization constraint $\langle\psi|\psi\rangle = 1$, the above system of equations *nearly* guarantees that Eq. 9 is fulfilled. However, it still permits a small subset of states that do not satisfy Eq. 9 due to three-qubit correlations, e.g.,

$$|\psi^*\rangle = \frac{1}{2}\left[\pm1, 0, 0, \pm1, 0, \pm1, \pm1, 0\right]^T.$$

States with higher-order correlations such as $|\psi^*\rangle$, which neither satisfy Eq. 9 nor are disallowed by Eq. 12, can be avoided by adding higher-order Pauli string constraints. For the above $n = 3$ example, we would add the $k = n = 3$ constraint $\langle\sigma^z_1\sigma^z_2\sigma^z_3,\rho\rangle = 0$, which would disallow $|\psi^*\rangle$ as $\langle\sigma^z_1\sigma^z_2\sigma^z_3,\rho^*\rangle = 1$.

Eq. 9 can also be systematically undermined by the unequal distribution of graph edges among the quantum states. For instance, the asymmetrically distributed edge-weights in skewed graphs (Fig. 5 left and Sec. 3). With such graphs, the minimization of the loss function can lead to outsized state populations for quantum states that encode high-degree (high edge-weight) vertices. Moreover, as the number of classical variables will not generally be a power of two, there will often be quantum states that are not encoded with a classical variable. For example and as detailed in Sec. 3, we use $n = 10$ qubits ($N = 2^n = 1024$ states) to solve the 800-vertex GSet graphs, such that the states 801 to 1024 are absent from the objective function. In such cases, the minimization of the loss function can lead to outsized state populations of quantum states that are present in the optimization function. In principle, these imbalances can be addressed by increasing the magnitude of the Pauli string amplitude constraints,

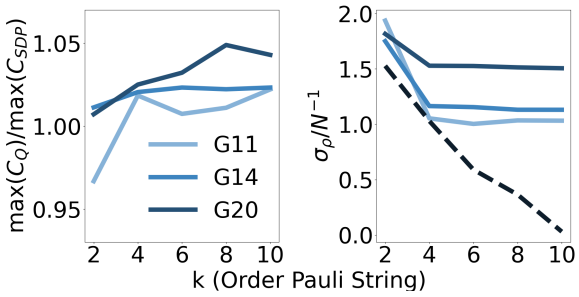


Figure 4: The effect of including higher-order HTAAC-QSDP Pauli string amplitude constraints in MaxCut optimization on the G11 (toroid), G14 (binary), and G20 (integer) graphs [32]. (Left) the performance of HTAAC-QSDP increases as higher-order Pauli strings are used to constrain state amplitudes. The algorithm's performance saturates with $k \approx 4$, indicating that the benefits saturate with less than a polynomial number of Pauli string constraints ($k \leq n$). As illustrated by this work (e.g., Fig. 2 and Table 2), $k = 2$ ($m \approx n^2/2$) is often sufficient for competitive SDP optimization. (Right, solid lines) the variance of state magnitude $\sigma_\rho = \text{var}(\rho_{ii}) = \text{var}(|\psi_i|^2)$ vs the order k of Pauli strings constraints. As k increases, σ_ρ decreases considerably, although this effect is largely saturated by $k \approx 4$. (Right, dashed line) in the absence of competing dynamics (i.e., $\langle\sigma_{n+1}\rangle_W$ and $\langle\sigma_{n+1}\rangle_P$), the Pauli string constraints are fully enforced such that $|\psi_i| \to N^{-1/2}$ ($\sigma_\rho \to 0$) as $k \to n$.

but this is known to cause poor objective function convergence [38].

To redress this systematic skew, we add a population-balancing unitary U_P (Fig. 1b), which is implemented on $|\psi\rangle$ via a second Hadamard Test (Sec. 2.1, Fig. 1c) and adds the loss function term $\langle\sigma_{n+1}\rangle_P$. Specifically, $U_P = \exp(i\beta P)$ where P is some diagonal operator of edge weights $P_{ii} = -(P_{max} - \sum_j |\omega_{ij}|)$, where β is an adjustable hyperparameter and $P_{max} = \max_i(\sum_j |\omega_{ij}|)$ is the maximum magnitude of edge weights for any given vertex. U_P works to balance the state populations by premiating the occupation of states that are lesser represented by or absent from the objective function $\langle\sigma_{n+1}\rangle_W$.

Combining both the efficient Hadamard Test objective function and the Approximate Amplitude Constraints, we can use simple gradient descent-based penalty methods [38] to find the solution. Specifically, we minimize the HTAAC-QSDP loss function

$$\mathcal{L}(t) = \langle\sigma_{n+1}\rangle_{W,t} + \langle\sigma_{n+1}\rangle_{P,t} + \lambda\left[\sum_j \langle\sigma_j^z, \rho_t\rangle^2 + \sum_{k\neq j}\langle\sigma_j^z\sigma_k^z, \rho_t\rangle^2\right] \tag{14}$$

at each time step t by preparing a quantum state ρ_t on a variational quantum computer. The scalar λ is the penalty hyperparameter. While for simplicity we have chosen a single, time-constant λ for all constraints, in principle each constraint j could be parameterized with a distinct λ_j, each of which could also vary as a function of t. The number of quantum circuit preparations required to optimize our HTAAC-QSDP protocol is constant with respect to the number of qubits n (and thus to the maximum number of vertices $N = 2^n$), as $\langle U_W, \rho_t\rangle$ and $\langle U_P, \rho_t\rangle$ each require only the Pauli-z measurement $\langle\sigma_{n+1}^z\rangle$ on the auxilary qubit, and the $m \sim n^2/2$ amplitude constraint terms can be calculated from a single set of n-qubit measurements on the state $|\psi\rangle$. The classical complexity of each training step scales as just $m \sim n^2/2$, as one marginal expectation value is calculated from the $|\psi\rangle$ measurements for each of the m constraints.

2.3 Retrieving SDP Solutions

At the end of our protocol, the SDP solution is encoded into $|\psi\rangle$. Like in other QSDP protocols, $|\psi\rangle$ may either be used to extract the full N-variable solution or for less computationally intensive analysis (i.e., to characterize the features of the solution or as an input state for further quantum protocols). If the full solution $|\psi\rangle$ is desired, then full real-space tomography of $|\psi\rangle$ must be conducted by calculating the N marginal distributions of all $k \leq n$ Pauli strings along the z and x-axes. We now show that once $|\psi\rangle$ is determined, it suffices to assign the partition of each vertex as $v_i = \text{sign}(\psi_i)$, or the sign of the state component ψ_i.

In classical semidefinite programming algorithms, such as the Goemans-Williamson algorithm [30], the optimal solution X^* is factorized by Cholesky decomposition into the product $X^* = T^\dagger T$, where T is an upper diagonal matrix. The sign of each vertex v_i is then designated as

$$v_i = \begin{cases} 1, & \text{if } \mathbf{t}_i \cdot \mathbf{g} \geq 0 \\ -1, & \text{otherwise,} \end{cases} \tag{15}$$

where $\mathbf{t}_i$ are the column vectors of T and $\mathbf{g}$ is a length-N vector of normally distributed random variables $g_j \sim \mathcal{N}(0,1)$.

We define the quantum parallel by noting that as $\rho = |\psi\rangle\langle\psi|$, its Cholesky decomposition is simply the $N \times N$ matrix that has the first row $\langle\psi|$ and and all other entries equal to zero. In this decomposition, Eq. 15 reduces to

$$v_i = \begin{cases} 1, & \text{if } \psi_0 \times g_0 \geq 0 \\ -1, & \text{otherwise.} \end{cases} \tag{16}$$

As MaxCut has $\mathbb{Z}_2$ symmetry, the cut values are symmetric under inversion, or flipping the sign of all vertices. This makes the sign of the normally distributed g_0 irrelevant to the graph partitioning. Without loss of generality, we can therefore set $g_0 = 1$ and classify each vertex as $v_i = \text{sign}(\psi_i)$.

2.4 Extensions to Other SDPs

As explained above, the Goemans-Williamson algorithm [30] can be applied to numerous other optimization algorithms, such as MaxSat and Max Directed Cut [30]. Moreover, HTAAC-QSDP can be adapted to accommodate the constraints of various other SDPs. As one example, consider the Min/Max Bisection problems [39]. Min/Max Bisection are particularly relevant to very-large-scale integration (VLSI) for integrated circuit design [40], a vital application area for large-scale SDPs.

The SDPs for estimating the Max Bisection problem has the standard form:

$$\begin{aligned} &\text{minimize}_{X \in \mathbb{S}^+} \ \langle W, X \rangle \\ &\text{subject to} \ \sum_{i,j} X_{ij} \leq -N/2, \\ &\text{and} \ \ X_{ii} = 1, \ \ \forall i \leq N. \end{aligned} \tag{17}$$

The first constraint is equivalent to requiring that half of the variables of X be partitioned equally, hence the term "bisection". In analogy with Eq. 10, Eq. 17 can be written as

$$\begin{aligned} &\text{minimize} \ \langle W, \rho \rangle \\ &\text{subject to} \ \sum_{i,j} \rho_{ij} \leq -N/2 \\ &\text{and} \ \ \rho_{ii} = N^{-1}, \ \ \forall i \leq N. \end{aligned} \tag{18}$$

The second of these two constraints can be enforced by the Pauli strings constraints of Eq. 12. For large N and assuming no systematic correlations between the ordering of the vertices, the first constraint can be ensured by adding any single Pauli string constraint

$$\langle \mathcal{O}_x \rangle = 0, \tag{19}$$

where $\mathcal{O}_x$ is any Pauli string of σ^x operators. To see how Eq. 19 enforces the first constraint of Eq. 10, consider that any operator $\mathcal{O}_x$ induces a bit-flip on a subset of qubits, such that each state ψ_i is mapped to another state $\psi_{i'}$. This means that $\langle \mathcal{O}_x \rangle = \langle \psi | \mathcal{O}_x | \psi \rangle$ is the sum of $N/2$ products $2\psi_i^* \psi_{i'}$, where for each i, $|\psi_i| \approx N^{-1/2}$, as enforced by the Pauli-Z constraints of Eq. 12. If the probability that a random state ψ_i of $|\psi\rangle$ is positive is p, then in the limit of large N and uncorrelated vertex assignment

$$\langle \mathcal{O}_x \rangle = p^2 + (1-p)^2 - 2p(1-p). \tag{20}$$

The above yields $\langle \mathcal{O}_x \rangle = 0$ *iff* $p = 1/2$, which would correspond to the equal partitioning of the vertices required by the Bisection problems. In the case of correlated vertex encodings, the average of several Pauli-X strings $\langle \mathcal{O}_x \rangle$ can be considered. We note that Eq. 19 can be modified to enforce any partition ratio by solving for $\langle \mathcal{O}_x \rangle$ (Eq. 20) with the desired p.

3 Simulations and Results

The viability of our HTAAC-QSDP method is displayed in Fig. 2 and Table 2. We compare C_Q, the cut values obtained by HTAAC-QSDP, to $\max(C_{SDP})$, the best results obtained by the leading gradient-based classical method [33]. We study all of the 800-vertex MaxCut problems explored in [33] (Table 2) in order to make an extensive comparison with the leading classical gradient-based interior point method. These graphs represent a broad sampling from the well-studied GSet graph library [32]. Graphs G11, G12, and G13 have vertices that are connected to nearest-neighbors on a toroid structure and ± 1 weights (Fig 5, right), while G14 and G15 (G20 and G21) have binary weights 0 and 1 (integer weights ± 1) and randomly distributed edge density that is highly skewed towards the lower numbered vertices (Fig 5, left).

Table 3: MaxCut statistics for the 800-vertex graphs G11 (toroid), G14 (skew binary), and G20 (skew integer) for different orders of Pauli string constraints k. We compare the best cut value $\max(C_{SDP})$ produced by the leading classical method [33] compared to those produced by HTAAC-QSDP, with each entry providing the ratio $\max(C_Q)/\max(C_{\text{SDP}})$ ($\text{mean}(C_Q)/\max(C_{\text{SDP}})$). With relatively few Pauli string constraints ($k = 4$), our method exceeds the performance of classical methods on all graphs studied.

Graph	$k = 2$	$k = 4$	$k = 6$	$k = 8$	$k = 10$
G11	0.967 (0.940)	1.019 (0.984)	1.007 (0.999)	1.011 (0.995)	1.022 (0.998)
G14	1.011 (1.000)	1.021 (1.009)	1.023 (1.010)	1.022 (1.014)	1.023 (1.012)
G20	1.007 (0.983)	1.025 (0.993)	1.032 (0.992)	1.049 (1.000)	1.043 (0.993)

HTAAC-QSDP with $k \leq 2$-Pauli term constraints exceeds the performance of its classical counterpart on skewed binary and skewed integer graphs, and falls narrowly short of classical performance on toroid graphs (Fig. 2 and Table 2). All trajectories converge above the 0.878-approximation ratio $C_Q/C_{\max}$ (dashed red line) guaranteed by classical semidefinite programming, where $C_{\max}$ is the highest known cut of each graph found by intensive, multi-shot, classical heuristics [36]. As SDPs are approximations of the optimization problem, the extremization of the loss function and the figure of merit (here, cut value) are correlated, but may not have a fully one-to-one correspondence. Fig. 3 demonstrates the strong correlation between the cut values C_Q and the efficiently encoded objective function $\langle\sigma^z_{n+1}\rangle_W = \text{Im}\,[\langle\psi|U_W|\psi\rangle]$, indicating that our method is a close SDP approximation with good convergence.

The addition of Pauli string amplitude constraints with $k > 2$ can better enforce Eq. 9, leading to more accurate SDP results. Fig. 4 and Table 3 demonstrate that increasing k produces moderately higher C_Q values, until the performance increases saturate $k \approx 4$. Moreover, we note that at k values ≈ 4, HTAAC-QSDP outperforms the analogous classical algorithm for all graph types. Likewise, the population variance (solid lines) $\sigma_\rho = \text{var}(|\psi_i|^2)$ decreases substantially until saturating near $k \approx 4$ at around $\sigma_\rho \approx N^{-1}$. In the absence of the competing objective function ($\langle\sigma^z_{n+1}\rangle_W$) and population balancing ($\langle\sigma^z_{n+1}\rangle_P$) dynamics, all Pauli-$z$ correlations become restricted as $k \to n$. This results in the complete constraint $|\psi_i| = N^{-1/2}$, $\forall i$, such that $\sigma_\rho \to 0$ (black dashed line in Fig. 4, left).

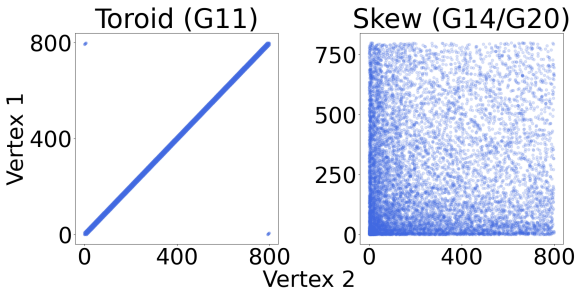


Figure 5: The structure of the G11 and G14/G20 GSet graphs, where non-zero edges between two vertices are marked as blue dots. Left) the edges of the toroid graphs (G11, G12, G13) follow a fixed connectivity, with edges extending between neighboring vertices on a torus structure. Right) the skewed graphs have connectivity that is drawn from a random distribution, with edges extending between arbitrary vertices (G14 and G15 binary weights 1 and 0, G20 and G21 integer weights ± 1). The degrees of each vertex are disproportionately biased towards the vertices of lower index, with edge density decaying as vertex number increases. We compare with all 800-vertex graphs considered in the leading classical analog [33].

3.1 Simulation Details

All simulations are done using a one-dimensional ring qubit connectivity, such that each qubit has two neighbors and the nth qubit neighbors the 1st qubit. The circuit ansatz simulations is 120 repetitions of two variationally parameterized y-axis rotations interleaved with CNOT gates, alternating between odd-even and even-odd qubit control. The TensorLy-Quantum simulator [41, 42] is used. Gradient descent was conducted an ADAM [43] optimizer, with learning rate $\eta = 0.01$, as well as hyperparameters $\beta_1 = 0.9$, and $\beta_2 = 0.999$.

The evolution angle α was set as $\alpha = 0.01$ for all graphs. The values of β used in this work were $\beta = 1/1.2$ for the toroid graphs and $\beta = 1/3$ for the skew binary and skew integer graphs. β values should be chosen such that $\beta < 1$, as di-

agonal entries are always satisfiable (i.e., some population can always be placed on the state, lowering the loss function), in contrast to edge cuts, which are not (i.e., not every edge can be cut with any given partition for general graphs). β values can be tuned on the device in the real time by monitoring the Pauli string constraints and choosing a β that leads to largely satisfied Pauli constraints with relatively small coefficients λ, such that the convergence of the algorithm is not hindered by large constraints that outweigh the objective function or lead to unstable convergence. In this work, we set $\lambda \propto \alpha/m$, to keep the total influence of m constraint terms in proportion to the objective term $\langle\sigma^z_{n+1}\rangle_W \approx \alpha W$. Specifically, we choose $\lambda = 100\alpha/m$ for the toroid and skew binary graphs and $\lambda = 50\alpha/m$ for the skew integer graphs.

4 Theoretical Analysis of Hadamard Test Objective Function

In this section, we derive Theorem 1, which we here restate for completeness:

Theorem 1 *Our approximate Hadamard Test objective function $U_W \sim i\alpha W$ (Sec. 2.1) holds for graphs with randomly distributed edges if*

$$\alpha^2 \lesssim \frac{N^4}{e^3} = \frac{N}{\xi^3},$$

where e is the number of non-zero edge weights and ξ is the average number of edges per vertex.

As discussed above, Theorem 1 can be understood in two ways: that α satisfies the approximation of Eq. 21 while remaining tractably large for SDPs of arbitrary N, as long as N does not 1) grow slower than the total number of edges e, or 2) grow slower than the the cube of ξ. We again note that Theorem 1 should hold for the densest graph region if the edge density is assymetrically distributed, i.e., ξ should be the average number of edges for the densest vertices. As the conditions of Theorem 1 hold for graphs that are not too dense, they are widely satisfiable as the majority of interesting and demonstrably difficult graphs for MaxCut are relatively sparse [7, 31, 32, 34, 35]. Many classes of graphs for which MaxCut is NP-hard satisfy Theorem 1 with tractably large α, even for arbitrarily large N.

As an example, we consider non-planar graphs, for which optimization problems like MaxCut are typically NP-complete. While planar graphs can be solved in polynomial time [44], a graph is guaranteeably non-planar when $e > 3N - 6$, which reduces to $\xi \geq 3$ in the limit of large N [45][1]. In accordance with Theorem 1, constant values of ξ actually permit α to *grow* as $N^{1/2}$, while for constant α ξ can grow as $N^{1/3}$, such that our approximation is valid for a wide variety of large-scale non-planar graphs. Indeed, most standard benchmarking graph sets have a small average number of edges per vertex, e.g., $\xi = 3$ [32, 34, 35], as sparse edge-density is common among graphs with real-world applications. In fact, solving MaxCut with many classes of dense graphs (i.e., graphs with nearly all non-zero edges) is provably less challenging, and therefore less interesting, than with their relatively sparse counterparts [47].

We here sketch a brief proof of Theorem 1 for Erdös–Rényi random graphs [48] with edge weights $W_{ij} \sim \mathcal{U}_{[0,b]}$, where $\mathcal{U}_{[0,b]}$ is the uniform distribution on the interval $[0, b]$. The edge density of a graph is described as $d = e/E$, where e is the number of non-zero edges e and $E = N(N-1)/2$ is the number of total possible edges. We provide a detailed proof of this and other graph types in the Appendix A.

Proof sketch of Theorem 1:

- The Hadamard Test encoding is a good approximation when $U_W \propto i\alpha W$.
- This is satisfied when $\frac{\alpha^3}{3!}|W^3|_{ij} \ll \frac{\alpha}{1!}|W|_{ij}$ for typical edges between vertices i,j.
- The mean of the non-zero elements in W is $\overline{W_{ij}} = b/2$ [2].
- Elements $(W^3)_{ij}$ are the sum of $\sim N^2$ terms $W_{ij}W_{jk}W_{kl}$, with expectation value

[1] Other families of easy graphs are even more restrictive, such as graphs that lack a giant component. In the limit of large N, these graphs only occur in more than a negligible fraction of all possible graphs when $d \geq 1/N$ and thus $\xi \geq 1/2$ [46].

[2] The mean value of *all* elements of W is $\overline{W_{ij}}' = db/2$, however the relevant comparison is between the elements of W^3 and the non-zero elements of W.

$\overline{W_{ij}W_{jk}W_{kl}} = b^3d^3/8$.

- $\frac{\alpha^3}{3!}\overline{(W^3)_{ij}} \ll \frac{\alpha}{1!}\overline{W_{ij}} \to \alpha^2 \ll 24/(N^2b^2d^3)$.
- Substituting $d = 2e/N(N-1) \approx 2e/N^2$ and $\xi = e/N$, we obtain Theorem 1.

5 Conclusion

The efficient optimization of very large-scale SDPs on variational quantum devices has to the potential to revolutionize their use in operations, computer architecture, and networking applications. In this manuscript, we have introduced HTAAC-QSDP, which uses $n+1$ qubits to solve SDPs of up to $N = 2^n$ variables and $M \sim 2^n$ constraints by taking only a constant number of quantum measurements and a polynomial number $m \sim n^2/2$ of classical calculations per epoch. As we approximately encode the SDP objective function into a unitary operator, the Hadamard Test can be used to optimize arbitrarily large SDPs by estimating a constant number of expectation values. Likewise, we demonstrate that the constraints of many SDPs can also be efficiently enforced with approximate amplitude constraints. Devising a quantum implementation the Goemans-Williamson algorithm, we approximately enforce the $M = 2^n$ constraints with a population-balancing Hadamard Test and the estimation of as few as $m \sim n^2/2$ Pauli string expectation values. We demonstrate our method on a wide array of graphs from the GSet library [32], approaching and often exceeding the performance of the leading gradient-based classical SDP solver on all graphs [33]. Finally, we note that by increasing the order k of our Pauli string constraints, we improve the accuracy of our results, exceeding the classical performance on all graphs while still estimating only polynomially-many expectation values.

Due to the immense importance of SDPs in scientific and industrial optimization, as well as the ongoing efforts to generate effective quantum SDP methods that are often limited by poor scaling in key parameters such as accuracy and problem size, our work provides a variational alternative with tractable overhead. In particular, the largest SDPs solved via classical methods, which required over 500 teraFLOPs on nearly ten-thousand CPUs and GPUs [49], could be addressed by our method with just ~ 20 qubits.

In future work, the techniques of this manuscript can be extended to additional families of SDPs. For instance, SDPs that extremize operator eigenvalues are a natural application for quantum circuits [50]. Similarly, variational quantum linear algebra techniques [51] can potentially be adapted to enforce the more general constraints

$$\langle A_\mu, X\rangle = b_\mu, \;\; \forall \mu \leq M$$

of Eq. 1. In many cases, more general constraints are likewise satisfiable with the Pauli string constraints, as suggested in this work. For instance, when the number of requisite constraints M is much smaller than the number of variables N, or, as is the case with our quantum implementation of the Goemans-Williamson algorithm, by enforcing a relatively small subset of the constraints.

6 Acknowledgments

At CalTech, A.A. is supported in part by the Bren endowed chair, and Microsoft, Google, Adobe faculty fellowships. S.F.Y. thanks the AFOSR and the NSF for funding. The authors would like to thank Madelyn Cain, Omar Shehab, and Antonio Mezzacapo for useful conversations.

References

[1] Stephen P. Boyd and Lieven Vandenberghe. "Convex optimization". Cambridge Press. (2004). url: web.stanford.edu/ boyd/cvxbook/.

[2] Michel X. Goemans. "Semidefinite programming in combinatorial optimization". Mathematical Programming **79**, 143–161 (1997).

[3] Lieven Vandenberghe and Stephen Boyd. "Applications of semidefinite programming". Applied Numerical Mathematics **29**, 283–299 (1999). url: web.stanford.edu/ boyd/papers/sdp-apps.html.

[4] Wenjun Li, Yang Ding, Yongjie Yang, R. Simon Sherratt, Jong Hyuk Park, and Jin Wang. "Parameterized algorithms of fundamental np-hard problems: a survey". Human-centric Computing and Information Sciences **10**, 29 (2020).

[5] Christoph Helmberg. "Semidefinite programming for combinatorial optimization". Konrad-Zuse-Zentrum fur Informationstechnik Berlin. (2000). url: opus4.kobv.de/opus4-zib/frontdoor/index/index/docId/602.

[6] Florian A. Potra and Stephen J. Wright. "Interior-point methods". Journal of Computational and Applied Mathematics **124**, 281–302 (2000).

[7] David P. Williamson and David B. Shmoys. "The design of approximation algorithms". Cambridge University Press. (2011). url: www.designofapproxalgs.com/.

[8] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. "Quantum computation by adiabatic evolution" (2000).

[9] Tameem Albash and Daniel A. Lidar. "Adiabatic quantum computation". Rev. Mod. Phys. **90**, 015002 (2018).

[10] Sepehr Ebadi, Alexander Keesling, Madelyn Cain, Tout T. Wang, Harry Levine, Dolev Bluvstein, Giulia Semeghini, Ahmed Omran, Jinguo Liu, Rhine Samajdar, Xiu-Zhe Luo, Beatrice Nash, Xun Gao, Boaz Barak, Edward Farhi, Subir Sachdev, Nathan Gemelke, Leo Zhou, Soonwon Choi, Hannes Pichler, Shengtao Wang, Markus Greiner, Vladan Vuletic, and Mikhail D. Lukin. "Quantum optimization of maximum independent set using rydberg atom arrays" (2022).

[11] Tadashi Kadowaki and Hidetoshi Nishimori. "Quantum annealing in the transverse ising model". Phys. Rev. E **58**, 5355–5363 (1998).

[12] Elizabeth Gibney. "D-wave upgrade: How scientists are using the world's most controversial quantum computer". Nature**541** (2017).

[13] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. "A quantum approximate optimization algorithm". arXiv (2014). arXiv:1411.4028.

[14] Juan M Arrazola, Ville Bergholm, Kamil Brádler, Thomas R Bromley, Matt J Collins, Ish Dhand, Alberto Fumagalli, Thomas Gerrits, Andrey Goussev, Lukas G Helt, et al. "Quantum circuits with many photons on a programmable nanophotonic chip". Nature **591**, 54–60 (2021).

[15] Fernando G. S. L. Brandão, Amir Kalev, Tongyang Li, Cedric Yen-Yu Lin, Krysta M. Svore, and Xiaodi Wu. "Quantum SDP Solvers: Large Speed-Ups, Optimality, and Applications to Quantum Learning". 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019) **132**, 27:1–27:14 (2019).

[16] Joran van Apeldoorn and András Gilyén. "Improvements in quantum sdp-solving with applications". In ICALP. (2019).

[17] Joran van Apeldoorn, Andràs Gilyèn, Sander Gribling, , and Ronald de Wolf. "Quantum sdp-solvers: Better upper and lower bounds". Quantum **4**, 230 (2020).

[18] Fernando G.S.L. Brandao and Krysta M. Svore. "Quantum speed-ups for solving semidefinite programs". In 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS). Pages 415–426. (2017).

[19] Fernando G.S L. Brandã o, Richard Kueng, and Daniel Stilck França. "Faster quantum and classical SDP approximations for quadratic binary optimization". Quantum **6**, 625 (2022).

[20] Dhrumil Patel, Patrick J. Coles, and Mark M. Wilde. "Variational quantum algorithms for semidefinite programming" (2021).

[21] Anirban N Chowdhury, Guang Hao Low, and Nathan Wiebe. "A variational quantum algorithm for preparing quantum gibbs states" (2020).

[22] Taylor L. Patti, Omar Shehab, Khadijeh Najafi, and Susanne F. Yelin. "Markov chain monte-carlo enhanced variational quantum algorithms" (2021).

[23] Youle Wang, Guangxi Li, and Xin Wang. "Variational quantum gibbs state preparation with a truncated taylor series". Physical Review Applied **16**, 054035 (2021).

[24] Sanjeev Arora, Elad Hazan, and Satyen Kale. "The multiplicative weights update method: A meta-algorithm and applications". Theory of Computing **8**, 121–164 (2012).

[25] Iordanis Kerenidis and Anupam Prakash. "A quantum interior point method for lps and sdps". ACM Transactions on Quantum Computing**1** (2020).

[26] Brandon Augustino, Giacomo Nannicini,

Tamás Terlaky, and Luis F. Zuluaga. "Quantum interior point methods for semidefinite optimization" (2021).

[27] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. "Variational quantum algorithms" (2020). arXiv:2012.09265.

[28] Kishor Bharti, Tobias Haug, Vlatko Vedral, and Leong-Chuan Kwek. "Nisq algorithm for semidefinite programming" (2021).

[29] Dorit Aharonov, Vaughan Jones, and Zeph Landau. "A polynomial quantum algorithm for approximating the jones polynomial". Algorithmica **55**, 395–421 (2009).

[30] Michel X. Goemans and David P. Williamson. "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming". J. ACM **42**, 1115–1145 (1995).

[31] Clayton W. Commander. "Maximum cut problem, max-cutmaximum cut problem, max-cut". Pages 1991–1999. Springer US. Boston, MA (2009).

[32] Steven J. Benson, Yinyu Yeb, and Xiong Zhang. "Mixed linear and semidefinite programming for combinatorial and quadratic optimization". Optimization Methods and Software **11**, 515–544 (1999).

[33] Changhui Choi and Yinyu Ye. "Solving sparse semidefinite programs using the dual scaling algorithm with an iterative solver". Manuscript, Department of Management Sciences, University of Iowa, Iowa City, IA**52242** (2000). url: web.stanford.edu/ yyye/yyye/cgsdp1.pdf.

[34] Angelika Wiegele. "Biq mac library – a collection of max-cut and quadratic 0-1 programming instances of medium size" (2007).

[35] Stefan H. Schmieta. "The dimacs library of mixed semidefinite-quadratic-linear programs". 7th DIMACS Implementation Challenge (2007).

[36] Yoshiki Matsuda. "Benchmarking the max-cut problem on the simulated bifurcation machine" (2019).

[37] R. M. Karp. "Reducibility among combinatorial problems". Springer US. Boston, MA (1972).

[38] Dimitri P Bertsekas. "Constrained optimization and lagrange multiplier methods". Academic press. (2014). url: www.mit.edu/ dimitrib/Constrained-Opt.pdf.

[39] Alan Frieze and Mark Jerrum. "Improved approximation algorithms for maxk-cut and max bisection". Algorithmica **18**, 67–81 (1997).

[40] Clark David Thompson. "A complexity theory for vlsi". Carnegie Mellon University. (1980). url: www.cs.auckland.ac.nz/ cthombor/Pubs/cmu-cs-80-140/cmu-cs-80-140scans.pdf.

[41] Taylor L. Patti, Jean Kossaifi, Susanne F. Yelin, and Anima Anandkumar. "Tensorly-quantum: Quantum machine learning with tensor methods" (2021).

[42] Jean Kossaifi, Yannis Panagakis, Anima Anandkumar, and Maja Pantic. "Tensorly: Tensor learning in python". Journal of Machine Learning Research **20**, 1–6 (2019). url: http://jmlr.org/papers/v20/18-277.html.

[43] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization" (2014).

[44] Brahim Chaourar. "A linear time algorithm for a variant of the max cut problem in series parallel graphs". Advances in Operations Research (2017).

[45] Yury Makarychev. "A short proof of kuratowski's graph planarity criterion". Journal of Graph Theory **25**, 129–131 (1997).

[46] Béla Bollobás. "The evolution of random graphs—the giant component". Page 130–159. Cambridge Studies in Advanced Mathematics. Cambridge University Press. (2001). 2 edition.

[47] Sanjeev Arora, David Karger, and Marek Karpinski. "Polynomial time approximation schemes for dense instances of np-hard problems". Journal of computer and system sciences **58**, 193–210 (1999).

[48] Rick Durrett. "Erdös–rényi random graphs". Page 27–69. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press. (2006).

[49] Katsuki Fujisawa, Hitoshi Sato, Satoshi Matsuoka, Toshio Endo, Makoto Yamashita, and Maho Nakata. "High-performance general solver for extremely large-scale semidef-

inite programming problems". In SC '12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis. Pages 1–11. (2012).

[50] Adrian S. Lewis and Michael L. Overton. "Eigenvalue optimization". Acta Numerica **5**, 149–190 (1996).

[51] Xiaosi Xu, Jinzhao Sun, Suguru Endo, Ying Li, Simon C. Benjamin, and Xiao Yuan. "Variational algorithms for linear algebra". Science Bulletin **66**, 2181–2188 (2021).

A Appendix: Theoretical Analysis of Hadamard Test Objective Function

We now derive Theorem 1 in detail. In order for the efficient encoding $U_W = \exp(i\alpha W) \propto i\alpha W$ to hold, it is sufficient to enforce that the third-order term in Eq. 11 is substantially smaller than the first-order term. That is

$$\frac{\alpha^3}{3!}|W^3|_{ij} \ll \frac{\alpha}{1!}|W|_{ij} \rightarrow \frac{\alpha^2}{6}|W^3|_{ij} \ll |W|_{ij} \quad (21)$$

for typical edges between vertices i,j. By induction, the criterion in Eq. 21 also guarantees that odd (imaginary) powers >3 will likewise be smaller than the first order term, and are thus also negligible. While this condition can always be satisfied with an arbitrarily small α, we in practice require that α maintain some finite size to avoid unitary rotations with vanishingly small gate times $\tau \propto \alpha$ and imaginary components $\langle\sigma_{n+1}\rangle_W \propto \alpha$. We now demonstrate that this criteria can be met for a wide array of graphs with NP-complete MaxCut optimization complexity.

First, we consider Erdös–Rényi random graphs [48] with elements $W_{ij} \sim \mathcal{U}_{[0,b]}$, which are uniformly distributed on the interval $[0, b]$. The graphs are said to have edge density d, which is the fraction of non-zero edges e over total possible edges $E = N(N-1)/2$. Typical elements $(W^3)_{ij}$ are the sum of $\sim N^2$ terms $W_{ij}W_{jk}W_{kl}$, with expectation value $\overline{W_{ij}W_{jk}W_{kl}} = b^3d^3/8$, such that the matrix elements of W^3 have the expectation value $\overline{(W^3)_{ij}} = N^2b^3d^3/8$. As the mean of the non-zero elements in W is $\overline{W_{ij}} = b/2$, the criterion of Eq. 21 becomes

$$\frac{\alpha^2N^2b^3d^3}{48} \ll \frac{b}{2} \rightarrow \alpha^2 \ll \frac{24}{N^2b^2d^3}. \quad (22)$$

We can rewrite this criterion in terms the number of non-zero edges e by noting that graph density d scales as $d = e/E$, where $E = N(N-1)/2 \approx N^2/2$ is the number of non-zero edges possible for an N vertex graph. Likewise, the average number of edges per vertex is then $\xi = e/N$, and Eq. 22 can be rewritten as

$$\alpha^2 \ll \frac{3N^4}{b^2e^3} = \frac{3N}{b^2\xi^3}. \quad (23)$$

For graphs where edge density d is not uniformly distributed, the above conditions should hold for the most densely connected vertices of the graph.

We briefly illustrate how our approximation holds for a few other classes of graphs. For instance, graphs with elements $W_{ij} \sim \mathcal{U}_{[-b,b]}$ drawn from uniform distributions with both positive and negative components generally require α ranges that are even more permissible (i.e., can be even larger) than those of the positive case, with the criterion of Eq. 22 serving as a small lower-bound.

Similar proofs of implementability can also be done for graphs with normally distributed weights $W_{ij} \sim \mathcal{N}(\mu, \sigma^2)$ of mean μ and variance σ^2. For the case $\mu \not\ll \sigma$, $\overline{(W^3)_{ij}} = N^2\mu^3d^3$ and α need only satisfy

$$\frac{\alpha^2N^2\mu^3d^3}{6} \ll \mu \rightarrow \alpha^2 \ll \frac{6}{N^2\mu^2d^3}, \quad (24)$$

which requires the same permissive scaling between N and d (e or ξ) as the condition Eq. 21 (Eq. 22) for positive uniform distributions. Likewise, for normal distributions where $\sigma \gg \mu$, Eq. 24 with $\mu \rightarrow \sigma$ would be a large upper bound.