Bang-bang control as a design principle for classical and quantum optimization algorithms

Aniruddha Bapat[1,2] and Stephen Jordan[3,4]

[1]Joint Center for Quantum Information and Computer Science, University of Maryland
[2]Department of Physics, University of Maryland
[3]Microsoft, Redmond, WA 98052, USA
[4]University of Maryland, College Park, MD 20742, USA

Abstract

Physically motivated classical heuristic optimization algorithms such as simulated annealing (SA) treat the objective function as an energy landscape, and allow walkers to escape local minima. It has been argued that quantum properties such as tunneling may give quantum algorithms advantage in finding ground states of vast, rugged cost landscapes. Indeed, the Quantum Adiabatic Algorithm (QAO) and the recent Quantum Approximate Optimization Algorithm (QAOA) have shown promising results on various problem instances that are considered classically hard. Here, building on previous observations from [1, 2], we argue that the type of *control* strategy used by the optimization algorithm may be crucial to its success. Along with SA, QAO, and QAOA, we define a new, bang-bang version of simulated annealing, BBSA, and study the performance of these algorithms on two well-studied problem instances from the literature. Both classically and quantumly, the successful control strategy is found to be bang-bang, exponentially outperforming the quasistatic analogues on the same instances. Lastly, we construct O(1)-depth QAOA protocols for a class of symmetric cost functions, and provide an accompanying physical picture.

1 Introduction

As quantum computing enters the so-called NISQ era [3], some focus has started shifting to noisy, shallow digital computations, and a need to re-examine existing quantum heuristic algorithms has emerged. The quantum adiabatic optimization algorithm (QAO), introduced in the previous decade [4], provides a paradigm for quantum speedups in optimization problems, where one performs a quasistatic Schrödinger evolution from an initial quantum state into the ground state of computational or physical interest. Runtime bounds for QAO typically depend, via adiabatic theorems, on the minimum spectral gap between the ground state and first excited state.

The Quantum Approximate Optimization Algorithm (QAOA) provides an alternative framework to designing quantum optimization algorithms, which is based on parameterized families of quantum circuits with adjustable parameters [5, 6]. Such *variational* circuits are parameterized by a depth, an initial quantum state, and a set of Hamiltonian operators under which the state can evolve. An instance of a variational circuit is further specified by a series of (labeled) evolution

1

times that determine which operator is applied and for how long. Along with QAOA, several other recent models of heuristic computation fit into the variational circuit paradigm [7, 8, 9, 10, 11].

A primary distinguishing feature between the quasistatic paradigm of QAO and simulated annealing (SA) and the variational circuit paradigm is in the design of their evolution schedules, from quasistatic to a rapidly switching, or *bang-bang*, schedule. Recently, it was observed [1, 2] that the Pontryagin Minimum Principle [12] implies that variational methods that employ a bang-bang evolution schedule are sufficient for optimality of the optimization protocol. Furthermore, the paper that introduces QAOA [5] also gives evidence pointing to an exponential speedup between QAOA and QAO. This raises two questions: Firstly, can a design shift from quasistatic to bang-bang yield provable superpolynomial improvements in the runtime, or are the two frameworks polynomially equivalent? Secondly, can the same control theoretic reasoning be applied to the design of classical optimization algorithms? In this work, we answer these questions by studying the performance of bang-bang controlled algorithms on certain well-studied instances, and make comparisons to the quasistatic, annealing-type algorithms. We prove that, on these instances, going from quasistatic scheduling to bang-bang can bring about an exponential speedup for both classical and quantum optimization. We also discuss the applicability and potential limitations of the optimal control framework to the problem of designing heuristic optimization algorithms.

2 Summary of results

The main results of this paper may be found in Sec. 8, where we study the performance of four candidate algorithms given in Table 2 on two benchmarking instances, and find that the bang-bang control algorithms exponentially outperform both classical and quantum annealing-based algorithms. These results are also summarized in Table 1.

Instance	Annealing-based		Bang-bang	
	QAO	SA	QAOA	BBSA
Bush, $\lambda \geq 1$	$\text{poly}(n)$ [13]	$\exp(n)$ [13]	$O(1)$§ 8.3.2	$\tilde{O}\left(n^{3.5\dots}\right)$§ 8.2.1
Bush, $\lambda < 1$	$\exp(n)$ [13]	$\exp(n)$ [13]	$O(1)$§ 8.3.2	$\tilde{O}\left(n^{3.5\dots}\right)$§ 8.2.1
Spike, $2a + b \leq 1$	$\text{poly}(n)$ [14]	$\exp(n)$ [13]	$O(1)$§ 8.3.1	$O(n)$§ 8.2.2
Spike, $2a + b > 1$	$\exp(n)$ [14]	$\exp(n)$ [13]	$O(1)$§ 8.3.1	$O(n)$§ 8.2.2

Table 1: Performance of the four algorithms, summarized. For the two instances studied, we distinguish different parameter regimes. For the Bush instance, the performance of QAO depends on the choice of mixer B_λ (see Eq. 23). For Spike, the QAO performance depends on spike parameters a and b. We see that bang-bang control algorithms outperform their (quantum and classical) annealing-based counterparts for these instances. Sources for existing results are cited, and the new contributions are referenced by the relevant sections.

In addition, we study the performance of single-round QAOA (or QAOA1) on a more general class of symmetric cost functions, and give sufficient conditions under which QAOA1 can successfully find minima for these functions. These results are stated in Lemma 1 and Theorem 1. In Sec. 6, we elaborate on the theoretical motivation behind choosing a bang-bang schedule and the caveats therein.

3 Preliminaries

First, we present some notation that will be used throughout the paper. Any problem instance of size n will be given as a constraint satisfaction problem on Boolean strings of length n. An n-bit string will be expressed as a boldfaced variable, e.g. $\mathbf{z} \in \{0,1\}^n$, in analogy with vector quantities. Variables denoting bits of a string will be expressed in normal font (e.g. the i-th bit of $\mathbf{z}$ is z_i). Similarly, the Hamming weight of a string, which is defined as the (integer) 1-norm of the bit string, or the number of 1's in a bit string,

$$|\mathbf{z}| := \sum_{i=1}^{n} z_i \tag{1}$$

will also be represented by non-bold letters such as w, v to indicate that it is a scalar quantity like the value of a bit.

We will be interested in expressing states by labels such as a string variable $\mathbf{z}$, or scalar variables w, z, etc. In either case, the convention will be to use 2-normalized kets $|\cdot\rangle$, or 1-normalized vectors, for which we will use the notation $|\cdot)$. In particular, a state labeled by Hamming weight w will denote the equal superposition over all bit strings with that Hamming weight,

$$|w\rangle := \frac{1}{\sqrt{\binom{n}{w}}} \sum_{|\mathbf{z}|=w} |\mathbf{z}\rangle, \quad |w) := \frac{1}{\binom{n}{w}} \sum_{|\mathbf{z}|=w} |\mathbf{z}) \tag{2}$$

Problem instances are given as a cost function on bit strings,

$$c : \{0,1\}^n \to \mathbb{Z} \tag{3}$$
$$c(\mathbf{z}) = \text{cost of bit string } \mathbf{z}. \tag{4}$$

There is a natural Hamiltonian operator C (and corresponding unitary $\mathcal{C}$) associated with this function that is diagonal in the computational basis, with eigenvalue $c(\mathbf{z})$ for every corresponding eigenvector $\mathbf{z} \in \{0,1\}^n$. Explicitly,

$$C := \sum_{\mathbf{z} \in \{0,1\}^n} c(\mathbf{z})|\mathbf{z}\rangle\langle\mathbf{z}|, \quad \mathcal{C}(\gamma) := e^{-i\gamma C} \tag{5}$$

Classical n-bit strings are naturally representable as vertices of an n-dimensional hypercube graph. This is often the representation of choice, as walks on the hypercube are generated by sequences of bit flips on the string, which correspond to the 1-local quantum operator

$$B := -\sum_{i=1}^{n} X_i, \quad \mathcal{B}(\beta) := e^{-i\beta B} \tag{6}$$

where $X_i \equiv \underbrace{\mathbb{1} \otimes \cdots \otimes \mathbb{1}}_{i-1} \otimes X \otimes \underbrace{\mathbb{1} \otimes \cdots \otimes \mathbb{1}}_{n-i}$, and $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ is the Pauli-X operator. Unitary evolutions of a quantum state under B, C achieve amplitude mixing and coherent, cost-dependent phase rotations, respectively. Canonically, both QAO and QAOA (see Table 2 for full names) use Hamiltonians of the form B and C. However, as discussed towards the end of Sec. 8.1, other choices can affect the performance on a given instance.

Abbreviation	Name of Algorithm	Reference
QAO	Quantum adiabatic optimization (algorithm)	[4]
SA	Simulated annealing	[13]
QAOA	Quantum approximate optimization algorithm	[5]
BBSA	Bang-bang simulated annealing	§ 5.1

Table 2: Table of abbreviations for the algorithms studied in this paper. The last algorithm, BBSA, is introduced in this paper.

Now, we will describe the candidate algorithms listed in Table 2. It will become evident that these algorithms can all be expressed in the control framework given in Appendix A. This connection is important, as it allows us to borrow existing results from optimal control theory to the setting of heuristic optimization.

4 Annealing-based algorithms

4.1 Simulated annealing

Simulated Annealing (SA) is a family of classical heuristic optimization algorithms that seek to minimize a potential via the evolution of a classical probability distribution under a simulated cooling process. The dynamics of the distribution are governed by two competing influences:

- Descent with respect to the cost function $c(\mathbf{z})$.

- Thermal fluctuations that kick the walker in a random uphill direction with Boltzmann probability, defined according to a controlled temperature parameter τ.

In practice, the above dynamics may be achieved via the following random walk:

1. Initialize the walker at location $\mathbf{r}_1$.

2. Run a p-round annealing schedule, where the i-th round is given by $t_i \in \mathbb{Z}_{\geq 0}$ time steps and a temperature parameter $\tau_i \in \mathbb{R}_{\geq 0}$. For index $i \in [p]$, run:

   (a) For t_i iterations, repeat:

       i. Pick direction $\mathbf{e}$ uniformly at random from available local unit displacement vectors.
       ii. Let $\delta_{\mathbf{e}} := c(\mathbf{r}_i \oplus \mathbf{e}) - c(\mathbf{r}_i)$ be the cost increase in moving walker from current position $\mathbf{r}_i$ to new position $\mathbf{r}_{i+1} = \mathbf{r}_i \oplus \mathbf{e}$.
       iii. If $\delta_{\mathbf{e}} \leq 0$, move to new location with certainty. Otherwise, move with Boltzmann probability $e^{-\delta_{\mathbf{e}}/\tau_i}$, where τ_i is the current temperature in the schedule. In other words,
       $$\Pr(\mathbf{r}_i \to \mathbf{r}_i \oplus \mathbf{e}) = \min\left\{1, e^{-\delta_{\mathbf{e}}/\tau_i}\right\}. \tag{7}$$

3. Repeat steps 1-2 several times, and report the minimal sampled configuration $\mathbf{z}^*$ and the corresponding cost $c(\mathbf{z}^*)$.

The temperature schedule $\boldsymbol{\tau} = (\tau_1, \tau_2, \ldots, \tau_p)$ must be optimized in order to achieve a final distribution that is well-supported on low-energy states (including the global minima, ideally). In practice, one applies a finite "cooling" schedule in which the elements of $\boldsymbol{\tau}$ descend from ∞ to 0. At each temperature τ_i, the time steps t_i may be seen as relaxation time steps, where the walker distribution equilibrates under thermal exchange with the simulated bath at temperature τ_i. In the limit of infinitely slow, monotonically decreasing temperature schedules that satisfy certain additional conditions arising from deep local minima in the problem instance, simulated annealing always converges to the lowest-cost configuration [15, 16, 17, 18]. However, finite-time schedules and a finite relaxation time per temperature step can undo the theoretical guarantee.

The position update of the walkers in the above scheme is implemented via the Metropolis-Hastings rule, where uphill motions are suppressed with Boltzmann probability. This implies that steeper climbs quickly become exponentially unlikely, resulting in an effective trapping of walkers in basins of depth $\sim \tau_i$. Within these basins, sufficiently high relaxation times allow the walkers to find deep minima. Intuitively, the walkers are allowed to climb barriers "just high enough" so as to settle into progressively deeper minima, as τ_i decreases through the course of the algorithm.

The above process may be seen as a discretization of an approximately equivalent, continuous-time Markov process. In the parlance of control introduced in Appendix A, the dynamics of the walker distribution is generated by a stochastic operator $H(\tau(t))$ that is singly controlled by the time-dependent temperature parameter $\tau(t)$. For two neighboring positions $\mathbf{z}, \mathbf{z}'$ in the space with a mutual displacement unit vector $\mathbf{e} = \mathbf{z}' - \mathbf{z}$, the corresponding matrix element may be written as

$$H(\tau)_{\langle \mathbf{z}\mathbf{z}' \rangle} = \begin{cases} \mathcal{Z}(\tau, \mathbf{z}), & \text{if } \mathbf{z} = \mathbf{z}' \\ 1, & \text{if } \delta_{\mathbf{e}} \leq 0 \\ e^{-\delta_{\mathbf{e}} c/\tau}, & \text{if } \delta_{\mathbf{e}} > 0 \end{cases} \tag{8}$$

where the diagonal term term $\mathcal{Z}(\tau, \mathbf{z})$ is the negative column sum of the $\mathbf{z}$ column of $H(\tau)$, a condition which ensures stochasticity of the Markov process. Then, the continuous-time dynamics of the probability vector are given by the differential equation

$$\dot{P}(t) = -H(\tau(t))P(t) \tag{9}$$

Under a discretization of the above into small time slices Δt_i (such that $||H(\tau_i)||\Delta t_i < 1$), and approximating the temperature schedule as a piecewise constant function, we may rewrite the continuous process as a Markov chain where the dynamics at the i-th slice are given by the stochastic matrix $\mathbb{1} - H(\tau_i)\Delta t_i$. This corresponds to the i-th step of the discrete random walk.

The infinite-temperature and zero-temperature limits of H are important special cases. At $\tau = \infty$, walkers choose random directions and walk with certainty, independent of the potential. This corresponds to the case of *diffusion*. On the other hand, at $\tau = 0$, walkers walk in a randomly chosen direction if and only if the resulting cost is no greater than the current cost. This is what we may call *randomized gradient descent*. We will denote these operators by D, G respectively and

give their form below:

$$D_{\langle \mathbf{zz'} \rangle} := H(0)_{\langle \mathbf{zz'} \rangle} = \begin{cases} -n(\mathbf{z}), & \text{if } \mathbf{z} = \mathbf{z'} \\ 1, & \text{if } \delta_{\mathbf{e}} \leq 0 \\ 1, & \text{if } \delta_{\mathbf{e}} > 0 \end{cases} \tag{10}$$

$$G_{\langle \mathbf{zz'} \rangle} := H(\infty)_{\langle \mathbf{zz'} \rangle} = \begin{cases} -n_<(\mathbf{z}), & \text{if } \mathbf{z} = \mathbf{z'} \\ 1, & \text{if } \delta_{\mathbf{e}} \leq 0 \\ 0, & \text{if } \delta_{\mathbf{e}} > 0 \end{cases} \tag{11}$$

where $n(\mathbf{z})$ is the number of neighbors, and $n_<(\mathbf{z})$ the number of "downhill" neighbors, of $\mathbf{z}$. (Note: For all bit strings $\mathbf{z}$, $n(\mathbf{z}) = n$ on the usual n-dimensional hypercube.)

4.2 SA with linear update

Under Metropolis-Hastings Monte Carlo, we see that the dynamics evolve under $H(\tau)$, which is an operator controlled by the temperature schedule $\boldsymbol{\tau}$. The obvious bang-bang analogue to this is to alternate between periods of zero- and infinite-temperature Metropolis moves, which is the algorithm introduced in Sec. 5.1. However, to argue that bang-bang control is optimal using the optimal control framework (as in Sec. 6), we must first ensure that the dynamics are linear in the controls. In this section, we present a linearized variant of SA, so that within algorithms of this class, it will be the case that bang-bang control is optimal as a consequence of the Pontryagin Minimum Principle.

Suppose that instead of Metropolis-Hastings probability $\min\left\{1, e^{-\delta_{\mathbf{e}}/\tau}\right\}$, we use a probability $u\Theta(\delta_{\mathbf{e}})$, where $\Theta(\cdot)$ is the Heaviside step function, and $u \in [0,1]$ is a control parameter. That is,

$$\Pr(\mathbf{z} \rightarrow \mathbf{z} \oplus \mathbf{e}) = \begin{cases} 1, & \text{if } \delta_{\mathbf{e}} \leq 0 \\ u, & \text{if } \delta_{\mathbf{e}} > 0 \end{cases} \tag{12}$$

This rule is qualitatively different from Metropolis-Hastings, since it attaches importance not to the exact energy difference between neighboring states, but only to its sign. Furthermore, the update rule is not guaranteed to satisfy physical prerequisites such as detailed balance that guarantee the convergence of the limiting distribution. However, it is a valid update rule, and we will call SA equipped with these dynamics linear update SA.

Importantly, linear update SA is expressible in the linear control framework. It is possible to write the Markov matrix $H(u)$ corresponding to the continuous version of Eq. 12 as a sum of the diffusion matrix D and the randomized gradient descent operator G,

$$H(u) = uD + (1-u)G \tag{13}$$

Finally, we see that, $H(u = 0) = D$ and $H(u = 0) = G$, thus reproducing the operators appearing in standard SA in the limit of infinite and zero temperature (i.e. $u = 0, 1$), which are the relevant parameter values under bang-bang control.

4.3 QAO

The adiabatic algorithm, proposed in 2000 by Farhi, Goldstone and Gutmann [4], is a (quantum) heuristic combinatorial optimization algorithm based upon the adiabatic theorem from quantum

mechanics. The adiabatic theorem, loosely stated, says that a system evolving under a time-varying Hamiltonian, when initialized in a ground state, stays in the instantaneous ground state as the Hamiltonian is varied *slowly* in time. The recipe to turn this statement into an algorithm for finding global minima is as follows:

1. Initialize the system in an easily preparable ground state of a Hamiltonian B.

2. Read the problem instance (cost function $c(\mathbf{z})$), and map it to an equivalent Hamiltonian C, as in Eq. 5.

3. Implement Schrödinger evolution of the state over the time interval $[0, T]$ under a controlled Hamiltonian $H(s) = u_1(s)B + u_2(s)C$, where $s = t/T$ is the scaled time parameter, and u_1, u_2 are functions of s that describe the *annealing schedule*. The schedule satisfies $u_1(0) = 1 - u_2(0) = 1$, and $u_1(1) = 1 - u_2(1) = 0$.

4. Measure the resulting state in the computational basis.

Under adiabaticity (i.e. when the schedule varies slowly in s), the above algorithm evolves the initial state from the ground state of B to that of C, which is a state that encodes the solution to optimization problem. In particular, the algorithm succeeds if the rate is slower than inverse polynomial in the first spectral gap $\lambda(s)$ (i.e. the energy difference between the ground state and the first excited state) at all times. Typically, this yields a condition on the true runtime T [19, 20]:

$$T \gtrsim O\left(\frac{1}{\lambda^2}\right) \tag{14}$$

where $\lambda = \min_s \lambda_s$. Therefore, the guarantee of success of an adiabatic protocol lies in knowing that the minimum gap λ does not scale super-polynomially with n. However, it should be noted that this does not rule out good empirical performance. In fact, by cleverly varying speed as a function of the instantaneous spectral gap, important speedups such as the Grover speedup [21], and the exponential speedup for glued trees, [22] can be recovered.

Like Linear Update SA, the QAO Hamiltonian

$$H(u_1, u_2) = u_1 B + u_2 C \tag{15}$$

fits into the linear control framework [1]. In fact, we may simplify the above Hamiltonian to a singly-controlled Hamiltonian as follows. In practical applications of QAO, there is a maximum magnitude threshold (say J) for the controls, given by hardware constraints. We assume that this cutoff does not scale with the input size of the instance. Assume also that the lower cutoff for both u_1 and u_2 is 0. In other words, $u_1, u_2 \in [0, J]$. These design constraints give us a restricted version of QAO where the controls are non-negative and bounded. This restriction is applied simply to state our algorithms within a uniform, linear control framework. Adiabatic algorithms for the instances studied in later sections fit within this framework.

Then, observe that when $u_1 + u_2 > 0$, we can rescale the controls by factor $u_1(s) + u_2(s)$, giving us the following mapping of the time variable and the controls:

$$\frac{\mathrm{d}s}{\mathrm{d}t} \mapsto \frac{\mathrm{d}s}{\mathrm{d}t} \cdot (u_1(s) + u_2(s)) \tag{16}$$

$$(u_1(s), u_2(s)) \mapsto \left(u := \frac{u_1(s)}{u_1(s) + u_2(s)}, 1 - u\right) \tag{17}$$

Under this mapping, the time parameter is rescaled by a factor of at most $2J$ (corresponding to a constant slowdown), while the parametric Hamiltonian now looks like

$$H(u_1, u_2) \mapsto H(u) = uB + (1-u)C \qquad (18)$$

When $u_1(s) = u_2(s) = 0$, which is the only case not covered by the above mapping, we see that the dynamics "switch off" completely. This feature is useful only when the total time T is greater than the time necessary to complete the algorithm. However, if we study protocols as a function of the time horizon T, this feature becomes unnecessary, and we may safely ignore it.

Therefore, we have successfully mapped QAO to a linear, single control framework with only a constant overhead in the run time. From now on, we assume that QAO possesses the form given in Eq. 18.

5   Bang-bang algorithms

In parallel with the developments in annealing-based methods, extensive studies have been conducted into the problem of optimal control of quantum dynamics (see [23]), particularly in the context of many-body ground state preparation, e.g [24]. It is often found to be that case that, contrary to a quasistatic schedule, a rapidly switching, bang-bang schedule could be engineered to prepare states quickly.

In combinatorial optimization, an alternative framework based on circuits with variable parameters has been investigated, and has recently gained interest with the introduction of the Quantum Approximate Optimization Algorithm (QAOA), [5]. This is in fact an example of bang-bang control, as observed in [1]. The related problem of ground state preparation has also been approached using Variational Quantum Eigensolver (VQE) ansätze [25] that bear close resemblance to QAOA in their setup. The recent work by Hadfield et al. [7] has proposed a relabeling of the acronym QAOA to the 'Quantum Alternating Operator Ansatz' to capture this generality. In this manner, a new path that explores classical design strategies of quantum algorithms, also known as a hybrid approach, has been paved.

In the coming sections, we will formally introduce QAOA, as well as a new, classical bang-bang version of SA which we call bang-bang SA, or BBSA. Then in Sec. 6, we will elaborate on the theoretical motivation behind choosing the bang-bang approach.

5.1   Bang-bang simulated annealing (BBSA)

BBSA is the restriction of linear update simulated annealing (see Sec. 4.2) to bang-bang schedules. In other words, this is an algorithm that alternately applies diffusion and randomized gradient descent to the state. An instance of this algorithm may then be specified by the number of rounds p (where in each round we apply the two operators in succession), and the corresponding evolution times for each round.

Observe that Metropolis-Hastings SA, when restricted to $\tau = 0, \infty$, reduces to bang-bang SA.

5.2   QAOA

The Quantum Approximate Optimization Algorithm (QAOA) was introduced by Farhi *et al.* in 2014, [5], as an alternative ansatz to the QAO. We note (as is done in [1]) that, like QAO, QAOA

is a restriction of the linearly controlled Hamiltonian to the case of bang-bang control, i.e., where we only allow $u = 0, 1$ at any given time.

Restricted in this way, a QAOA protocol effectively implements a series of alternating Hamiltonian evolutions under the mixing operator B, and the cost operator C. Therefore, for a total of p rounds of alternating evolution with evolution angles $\vec{\beta} := (\beta_1, \ldots, \beta_p), \vec{\gamma} := (\gamma_1, \ldots, \gamma_p)$ for B and C respectively, the final state prepared by QAOA may be expressed as

$$|\vec{\beta}, \vec{\gamma}\rangle = \left[\prod_{i=1}^{p} \mathcal{B}(\beta_i)\mathcal{C}(\gamma_i) \right] |\psi_0\rangle \tag{19}$$

where we used the parameterized operators from Eq. 5, 6, and, as in the case of QAO, the initial state $|\psi_0\rangle$ is an easily preparable state such as the equal superposition of bitstrings, $|+^{\otimes n}\rangle$.

QAOA with a fixed number of rounds p, also written as QAOAp, is a scheme for preparing one of a family of trial states of the form $|\vec{\beta}, \vec{\gamma}\rangle$. With the angles as search parameters, a figure of merit such as the energy expectation of the cost operator $E(\vec{\beta}, \vec{\gamma}) = \langle \vec{\beta}, \vec{\gamma}|C|\vec{\beta}, \vec{\gamma}\rangle$ is approximately minimized with the aid of classical outer loop optimization.

6 Conditions for optimality of bang-bang control

Now, we will elaborate on the theoretical motivation for choosing a bang-bang approach to optimization algorithms, expanding on the observations made in [1, 2]. The Pontryagin Minimum Principle (PMP) from optimal control theory [12] provides key insight into the nature of optimal schedules for heuristic optimization algorithms expressible in the control framework. As discussed in Appendix A, PMP gives necessary conditions on the control in the form of a minimization of the control Hamiltonian, which is a classical functional of the state amplitudes and corresponding conjugate "momenta", and depends on the control parameters as well.

When the control Hamiltonian $\mathcal{H}$ is linear in the control vector $\mathbf{u}$, the minimization condition Eq. 53 implies that the optimal control is *extremal*, in the sense that the control only takes values on the boundary of the feasible control set at any given time. When the control parameters are individually constrained to lie in a certain interval, $u_i \in [a_i, b_i]$, then we say that the optimal protocol is bang-bang, i.e. $u_i(t) = a_i$ or b_i. Thus, the individual controls switch between their extremal values through the course of the protocol. While the heuristic algorithms QAO, QAOA and SA with linear update satisfy the condition of linear control, one should exercise caution when stating the optimality of bang-bang control within these frameworks. We note a few important caveats here:

1. PMP simply gives a necessary condition for optimality, it does not provide the optimal protocol. A different control theory tool, the Hamilton-Jacobi-Bellman equation, does provide a way to find the optimal protocol via dynamic programming.

2. There may be an arbitrary number of switches in the optimal bang-bang protocol. In fact, some problems exhibit the so-called *Fuller phenomenon*, in which the optimal control sequence has an infinite number of bangs, and is therefore rendered infeasible.

3. The control Hamiltonian may become *singular* at any point during the protocol. A singular interval is one in which the first derivative of $\mathcal{H}$ with respect to u vanishes. In these intervals, the optimal control is not necessarily bang-bang. The presence of generic singular intervals has

already been observed before in the dynamics of spin systems (see, e.g. [26, 27]). Therefore, in order to guarantee that the optimal control is bang-bang at all times, one must first show that there are no singular intervals during the protocol.

4. The original PMP is stated and proved for dynamics over Euclidean vector spaces over $\mathbb{R}$. However, in quantum optimization the amplitudes take values in $\mathbb{C}$, and the Hilbert space is a complex projective vector space with a non-Euclidean geometry. The generalization must be made with caution.

Despite these caveats, PMP does provide theoretical motivation for using bang-bang control as a design principle for heuristic optimization algorithms. In the following sections, we exhibit examples where bang-bang control exponentially outperforms conventional SA and QAO.

7 The problem instances

Now, we describe the problem instances that will be used as benchmarks for our algorithms. The two following instances have appeared in the context of comparisons between quantum and classical heuristic optimization algorithms, usually to show the inability of the classical algorithm to escape a local minimum and find the true, global minimum [13, 14, 28]. This is often interpreted as evidence of a quantum advantage, such as the ability to tunnel through barriers. In keeping with this tradition, we will select these as our benchmarking instances, and look for general features in the performance of our candidate algorithms.
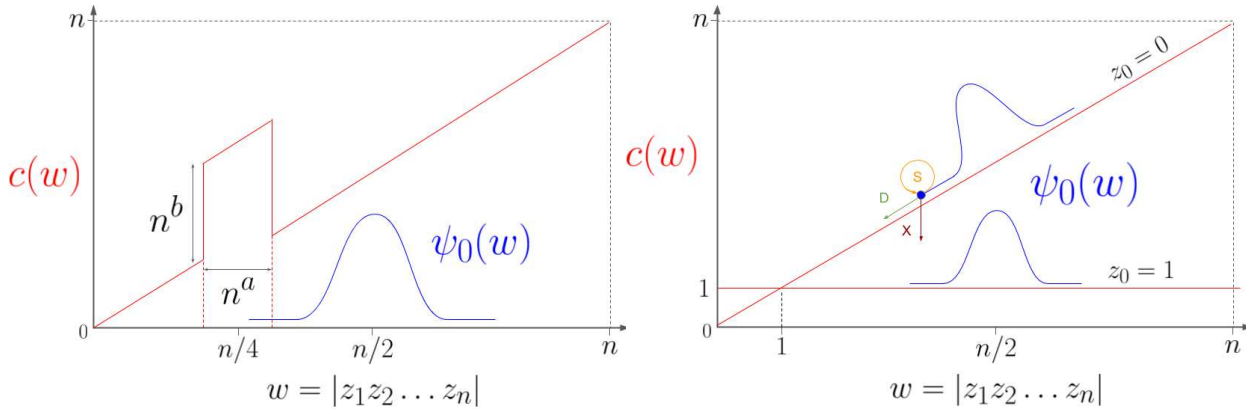

Figure 1: Schematic energy landscapes of the two instances, `Spike` (left) and `Bush` (right). In each diagram, the blue curve indicates the distribution of the initial state, the equal superposition over all bit strings.

7.1 Bush of implications

The bush of implications or `Bush` is an instance first crafted in [13] in order to demonstrate the failure of SA where QAO succeeds, with an exponential separation between the two. In `Bush`, the potential is not fully symmetric under permutation of bits. Instead, the first bit (the "central" bit, indexed by 0) determines the potential acting on the Hamming weight of the remaining n

"peripheral" bits. Specifically,

$$c(\mathbf{z} = z_0 z_1 \ldots z_n) = z_0 + \sum_{i=1}^{n} z_i \left(1 - z_0\right) = z_0 + w \left(1 - z_0\right) \tag{20}$$

where $w = |z_1 \ldots z_n|$. So, the potential is constant and equal to 1 when $z_0 = 1$, and a Hamming ramp, $r(w) = w$ when $z_0 = 0$, as shown in Fig. 1. Note that we adopted a bit-flipped definition of c as compared to the original in [13]. The reason is simply notational convenience. The energy landscape of the bush of implications can be viewed as the number of clauses violated in a constraint satisfaction problem, where each clause takes the form $\neg z_0 \implies \neg z_j$ for $j > 0$, which lends the instance its name.

7.2 Hamming ramp with spike

Next, we present a second family of Hamming-symmetric potentials studied first in [13, 29], the Hamming ramp with a spike. In the general form more recently studied in [14, 30, 28], this potential is given by a ramp $r(w) = w$, plus a rectangular "spike" function $s(w)$ centered at $w = n/4$ with width $O(n^a)$ and height $O(n^b)$, for two exponents $a, b \in [0, 1]$.

$$\text{Ramp:} \quad r(w) = w, \quad \text{Spike:} \quad s(w) = \begin{cases} n^b, & \text{if } w \in [\frac{n}{4} - \frac{n^a}{2}, \frac{n}{4} + \frac{n^a}{2}] \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

$$\text{Full Potential:} \quad c(w) = r(w) + s(w) \tag{22}$$

We will use this form for the `Spike` family of instances.

8 Performance

Now, we will state the performance of the algorithms from Sec. 4, 5 on the instances defined in Sec. 7.1, deriving or using existing results as appropriate. We will find that in both the classical (SA vs. BBSA) and the quantum (QAO vs. QAOA) settings, there exist parameter regimes in which the bang-bang algorithms are exponentially faster than their quasistatic analogues.

8.1 SA and QAO

For both the `Bush` and `Spike` examples, Farhi et al. argue in [13] that simulated annealing gets stuck in local minima, and is exponentially unlikely to reach the global minimum in polynomial time, in the input size $n \to \infty$. Additionally, they argue for the success of QAO on these instances in certain parameter regimes.

For the `Spike` example, [29] and [30] show that when the width and height parameters satisfy $a + b \leq 1/2$, quantum annealing solves `Spike` efficiently. If, on the other hand, $2a + b > 1$, it was shown by [14] that the minimum spectral gap has an exponential scaling in n, implying the failure of quantum annealing in this problem regime. For the `Bush` example, it was shown in [13] that the gap scaling is polynomial in n, thus allowing for an efficient adiabatic algorithm to solve this instance. We note that the performance depends on the choice of the initial mixing Hamiltonian

B. In particular, out of the following family of mixers

$$B_\lambda = -\lambda(n+1)X_0 - \sum_{i=1}^{n} X_i, \qquad (23)$$

QAO is successful when $\lambda \geq 1$. On the other hand, when $\lambda = 1/(n+1) < 1$, we recover the canonical mixing operator B from Eq. 6, and QAO is expected to take exponential time to solve `Bush`.

Despite the caveats, `Bush` and `Spike` are examples of instances where we have an exponential separation between a quantum (QAO) and classical (SA) algorithm. However, in the next section we show that a different, purely classical, bang-bang strategy matches the performance of QAO on the `Bush` and `Spike` instances by solving them in polynomial time.

8.2 Bang-bang simulated annealing

Now, we will show that the bang-bang version of simulated annealing is able to find the ground state of both `Bush` and `Spike` in time polynomial in n, and therefore exponentially outperforms SA (and QAO for certain parameter regimes, see Table 1), on both instances.

8.2.1 Bush

We will now show that BBSA efficiently finds the minimum of `Bush` via BBSA. In fact, the protocol simply involves performing randomized gradient descent (G) without any switches to diffusion. First, we characterize the G matrix for this instance. The natural basis for this problem is a conditional Hamming basis $\{|z_0, w) : z_0 \in [1], w \in [n]\}$ parameterized by the value of the central bit z_0, and the weight of the peripheral string $w = |z_1 \cdots z_n|$. The allowed transitions under G are as given below:

$$|0, w) \to |1, w), \text{ for all } w > 0. \qquad (24)$$

$$|z_0, w) \to |z_0, w-1), \text{ for all } z_0 \in [1], w > 0. \qquad (25)$$

$$|1, 0) \to |0, 0). \qquad (26)$$

In particular, this implies that a walker at the global minimum $|0,0)$ cannot leave under G. Consider a discrete, Markov chain Monte Carlo implementation of G, in which we break up the Markov evolution into $N = 1/\delta t$ steps of size δt. The stepsize δt is an empirical parameter which will be set later, while at the moment we only assume that $\delta t \ll 1$. Then, we may write the Markov evolution as

$$|P_N) = \left[\prod_{i=1}^{N} e^{-G\delta t}\right] |P_0) \simeq \left[\prod_{i=1}^{N} (\mathbb{1} - G\delta t)\right] |P_0) \qquad (27)$$

Each step $\mathbb{1} - G\delta t$ above is a stochastic evolution if δt is sufficiently small, i.e., if all entries of the matrix represent valid probabilities. The requirement that the column sum be 1 is automatically satisfied since G is column-sum-zero. Then, we start with a walker sampled from the initial state $|P_0)$, and, for every step 1 to N, we update the walker's position based on the transition probabilities given by $\mathbb{1} - G\delta t$. This is given in more detail below. We will show that the above procedure transports a fraction of at least $n^{-2.503}$ of walkers to the global minimum, in number of steps

$N = O\left(\frac{1}{\delta t}\log n\right)$. Finally, arguing that it suffices to choose $\delta t = \Theta(n^{-1})$ gives a polynomial runtime of $\Theta(n^{3.503}\log n)$ to have a constant success probability.

In our analysis, we only keep track of the walker in the $z_0 = 0$ subspace, which contains the global minimum. Any walker that starts in or enters the $z_0 = 1$ subspace during the algorithm will be presumed dead, and we terminate its walk. This simplification is allowed, since it may only worsen the success probability obtained through this analysis. Initially, exactly half of the walkers are alive, i.e. in the subspace $z_0 = 0$, and concentrated in a band of width $\sim \sqrt{n}$ around $w = n/2$. For a walker at Hamming weight $w > 0$, there are three possible moves (illustrated in Fig. 1):

1. (D) Descend to weight $w - 1$, with probability $w\delta t$.

2. (S) Stay at the same location with probability $1 - (w + 1)\delta t$.

3. (X) "Die", i.e., escape to the $z_0 = 1$ subspace, with probability δt.

When $w = 0$, the D and X moves are forbidden, and the walker can only stay in place. Additionally, we denote the event of survival (i.e. D or S) by $\bar{X}$. Now, we track the random walk under the stated moves. Let $\hat{m}$ be a random variable representing the total number of moves the walker takes to reach the global minimum, $|0, 0\rangle$. If the walker dies, we say that $\hat{m} = \infty$. Otherwise, $\hat{m}$ is finite and equal to the sum of number of moves spent at each weight $w = 1, 2, \ldots, n$. Defining a corresponding random variable $\hat{m}_w$ for the number of moves spent at each weight, we may write

$$\hat{m} = \sum_{w=1}^{n} \hat{m}_w \tag{28}$$

The expected value of $\hat{m}$ tells us how many moves any given walker needs to reach the global minimum under G. However, since we are only interested in living walkers, we will condition the expectation on the walker staying alive ($\bar{X}$). Then,

$$\mathbb{E}\left(\hat{m} \mid \bar{X}\right) = \sum_{w=1}^{n} \mathbb{E}\left(\hat{m}_w \mid \bar{X}\right) \tag{29}$$

At each weight w, the condition of survival limits the allowed moves to the regular expression S^*D. In other words, the walker stays in place for some number of moves before descending. Note that the probability of not dying in m moves is $(1 - \delta t)^m$. Therefore, the probability of spending m total moves, conditioned on survival, is given by

$$\Pr\left(\hat{m}_w = m \mid \bar{X}\right) = \frac{\Pr\left(S^{m-1}D\right)}{\Pr\left(\bar{X}^m\right)} = \frac{(1 - (w + 1)\delta t)^{m-1}w\delta t}{(1 - \delta t)^m} \tag{30}$$

$$= \left(\frac{1 - (w + 1)\delta t}{1 - \delta t}\right)^{m-1} \cdot \frac{w\delta t}{1 - \delta t} \tag{31}$$

$$\lesssim e^{-w\delta t(m-1)}\frac{w\delta t}{1 - \delta t} \tag{32}$$

where the last inequality follows from a Taylor series comparison of $(1 - (w + 1)\delta t)/1 - \delta t$ under the assumption that $\delta t < 1$. So, the expectation value of $\hat{m}_w$ is

$$\mathbb{E}\left(\hat{m}_w \mid \bar{X}\right) = \sum_{m=1}^{\infty} m \cdot \Pr\left(m \mid \bar{X}\right) \lesssim \frac{w\delta t \cdot e^{w\delta t}}{1 - \delta t}\sum_{m=1}^{\infty} m \cdot e^{-mw\delta t} = \frac{w\delta t}{(1 - \delta t)(1 - e^{-w\delta t})^2} \tag{33}$$

13

Finally, the full expectation value is given by

$$\mathbb{E}\left(\hat{m} \mid \bar{X}\right) \lesssim \frac{1}{1 - \delta t} \sum_{w=1}^{n} \frac{w\delta t}{\left(1 - e^{-w\delta t}\right)^2} \tag{34}$$

Next, using the variable substitution $x = w\delta t, dx = \delta t$, we may turn the above sum into an approximate integral. In fact, the integrand $x/(1 - e^{-x})^2$ is monotonically decreasing, so the sum is upper bounded by

$$\mathbb{E}\left(\hat{m} \mid \bar{X}\right) \lesssim \frac{\delta t}{(1 - \delta t)\left(1 - e^{-\delta t}\right)^2} + \frac{1}{\delta t(1 - \delta t)} \int_{\delta t}^{n\delta t} \frac{x}{\left(1 - e^{-x}\right)^2} dx \tag{35}$$

$$\lesssim \frac{4}{(1 - \delta t)\,\delta t} + \frac{1}{\delta t(1 - \delta t)} \int_{\delta t}^{n\delta t} \frac{4}{x} dx = \frac{4}{(1 - \delta t)\,\delta t} + \frac{4}{\delta t(1 - \delta t)} \log(n) \tag{36}$$

where we used the trick that since $x/2$ and $1 - e^{-x}$ are both monotonically increasing, and $x/2 < 1 - e^{-x}$ for $x = 0, 1$, then it follows that $x/2 < 1 - e^{-x}$ for all $x \in [0, 1]$. In fact, a tighter bound may be obtained by replacing 2 by $e/(e - 1) \approx 1.58$, which yields a scaling of $\mathbb{E}\left(\hat{m} \mid \bar{X}\right) \lesssim \frac{2.503}{\delta t} \log n$. Finally, the expected survival probability is $\Pr\left(\bar{X}\right) \gtrsim e^{-\delta t \cdot \frac{2.503}{\delta t} \log n} = \frac{1}{n^{2.503}}$, which is polynomial in n. Therefore, applying this algorithm for $\frac{1}{\delta t} \log n$ with $\delta t = \Theta(n^{-1})$, yields a polynomial probability of success. Repeating for at most $n^{2.503}$ trials amplifies the success probability to a constant. So, the total time complexity is $On^{3.503} \log n$, which is efficient in the input size n.

In Fig. 2 below, numerics of the continuous-time process (see Eq. 27) confirm that the total time indeed scales as $\log n$.
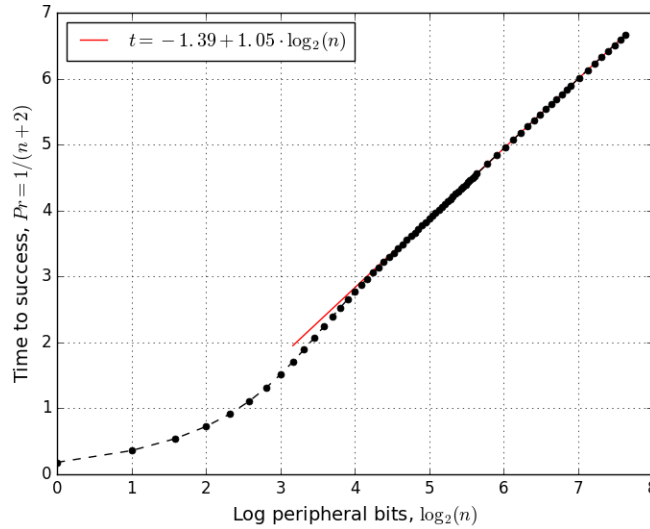

Figure 2: Plot of the input size n vs. total time for success (determined by the time taken for a polynomial fraction of walkers to reach the global minimum). Note that the continuous-time process does not contain the polynomial factors; those arise from discretization into small timesteps δt of order $\lesssim 1/n$.

14

8.2.2 Spike

In the previous section, we showed that `Bush` is a problem instance where classical bang-bang algorithm (BBSA) can outperform a classical quasistatic algorithm (SA) exponentially. While this suffices to show the polynomial inequivalence of SA and BBSA, it is nonetheless interesting to explore further examples where this is the case. The `Spike` problem, as presented in 21, is the second instance where BBSA can exponentially outperform SA and QAO. Since the separation is sensitive to details such as the shape of the spike, we refer the reader to Appendix B for further discussion.

8.3 QAOA

Lastly, we will show that one round of QAOA (or QAOA1) efficiently finds the minimum of the instances `Bush` and `Spike`. In fact, as discussed later in this section, QAOA1 solves a more general class of symmetric instances that includes the `Spike` (and with some more analysis, the `Bush`) example. This is one of the main results of the paper, given in Theorem 1.

8.3.1 Spike

One of the key features of this instance is that the spike has exponentially small overlap with the initial state $|+\rangle^{\otimes n}$. Intuitively, this implies that the state does not "see" the spike, and should therefore behave as if evolving under a pure Hamming ramp. We state this as the following lemma:

Lemma 1. *Let $c(w)$ be a Hamming-symmetric cost function on bitstrings of size n, and let $p(n) \in [0, 1]$ be a problem size-dependent probability. Suppose $c(w) = r(w) + s(w)$, where r, s are two functions satisfying the following:*

1. *$\min_w c(w) = \min_w r(w)$.*

2. *There exist angles β, γ such that QAOA1 with schedule (β, γ) minimizes $r(w)$ with probability at least $p(n)$.*

3. *If the initial state is $|\psi_0\rangle = \sum_w A_w |w\rangle$, then $s(w)$ overlaps weakly with $|\psi_0\rangle$ in the sense that*

$$\sum_{w=1}^{n} 4|A_w|^2 \sin^2\left(\frac{\gamma s(w)}{2}\right) \leq o(p(n))$$

Then, QAOA1 with schedule (β, γ) minimizes $c(w)$ with probability at least $p(n) - o(p(n))$.

For the `Spike` instance, we decompose the cost into a ramp term and a spike, $c(w) = r(w) + s(w)$. First, we compute the success probability of QAOA1 on only the ramp term $r(w)$. This potential may be written as

$$R = \sum_{w=0}^{n} w|w\rangle\langle w| = \sum_{i=1}^{n} \frac{\mathbb{1} - Z_i}{2} = \frac{n}{2}\mathbb{1} - \frac{1}{2}\sum_{i=1}^{n} Z_i \tag{37}$$

which is a 1-local operator on qubits, just like B. It can be seen that the protocol simply applies a rotation from the $|+\rangle$ state to the $|0\rangle$ state on each qubit via a $Z/2$ rotation followed by an X rotation, and succeeds with probability 1. The angles can be read off from the Bloch sphere: $\gamma = 2 \cdot \pi/4 = \pi/2$, and $\beta = \pi/4$.

Then, it follows from Lemma 1 that the effect of the spike $s(w)$ under QAOA1 is negligible if $\sum_w 4\sin^2(\gamma s(w)/2)|A_w|^2$ is small, where A_w are amplitudes of the initial state in the symmetric basis. But this sum may be bounded as

$$\sum_{w=0}^{n} 4\sin^2(\gamma s(w)/2)|A_w|^2 = \frac{1}{2^{n-2}} \sum_{w=n/4-n^a/2}^{n/4+n^a/2} \sin^2(\gamma n^b/2)\binom{n}{w} \tag{38}$$

$$\leq \frac{1}{2^{n-2}} \sum_{w=n/4-n^a/2}^{n/4+n^a/2} \binom{n}{w} = 4 \sum_{w=n/4-n^a/2}^{n/4+n^a/2} B(w;n,1/2) \tag{39}$$

where $B(w;n,1/2)$ is a binomial term corresponding to the probability of n tosses of a fair coin returning exactly w heads. Now, we may use known bounds on tail distributions such as Hoeffding's inequality, and we finally have

$$\sum_{w=0}^{n} 4\sin^2(\gamma s(w)/2)|A_w|^2 = 4 \sum_{w=n/4-n^a/2}^{n/4+n^a/2} B(w;n,1/2) = o(1) \text{ when } a < 1 \tag{40}$$

Then, applying Lemma 1, we conclude that, for a spike with $a \in [0,1)$ and arbitrary b, QAOA1 with angles $(\pi/4, \pi/2)$ finds the global minimum with probability polynomially close to 1.

This QAOA1 protocol is asymptotically successful for any (a,b) chosen from the set $[0,1) \times \mathbb{R}$. In practice, finite n instances will show effects of the finite overlap of the initial state with the spike at a close to 1. But even in this regime, the barrier height is essentially irrelevant, since it appears in the argument of a sinusoid and may only affect the bounds in Eq. 38 by a constant.

8.3.2 Bush

The `Bush` instance is a quasi-symmetric potential, since it depends on the value of the central bit. In the $z_0 = 1$ sector, the potential is a constant, while in the $z_0 = 0$ sector, it is a ramp. So, in analogy with Eq. 37

$$C = |1\rangle\langle 1| \otimes \mathbb{1} + |0\rangle\langle 0| \otimes \left(\frac{n}{2}\mathbb{1} - \frac{1}{2}\sum_{i=1}^{n} Z_i\right) \tag{41}$$

For ease of analysis, separate the mixing operator into the mutually commuting peripheral terms and central term:

$$\mathcal{B}(\beta) = e^{-i\beta B} = (\cos\beta\mathbb{1}_0 - i\sin\beta X_0)\prod_{i=1}^{n}(\cos\beta\mathbb{1}_i - i\sin\beta X_i) \equiv \mathcal{B}_0\mathcal{B}_i$$

As before, the QAOA protocol implements one Z rotation (operator $\mathcal{C}(\gamma) = e^{-i\gamma C}$) followed by an X rotation (operator $\mathcal{B}(\beta)$). Since the `Bush` potential contains a ramp in the relevant sector, we will try the protocol used for the `Spike` instance, $\beta = \pi/4, \gamma = \pi/2$.

The Z-rotation transforms the initial state (on the peripheral bits) into the $+Y$ eigenstate, $|+\rangle^{\otimes n} \to \frac{1}{\sqrt{2^n}}(|0\rangle + i|1\rangle)^{\otimes n}$. So, the full state transforms as

$$\frac{1}{\sqrt{2}}|1\rangle \otimes |+\rangle^{\otimes n} + \frac{1}{\sqrt{2}}|0\rangle \otimes |+\rangle^{\otimes n} \xrightarrow{\mathcal{C}(\pi/2)} \frac{-i}{\sqrt{2}}|1\rangle \otimes |+\rangle^{\otimes n} + |0\rangle \otimes \frac{1}{\sqrt{2^{n+1}}}(|0\rangle + i|1\rangle)^{\otimes n}$$

Next, $\mathcal{B}_i$ transforms the state to

$$\frac{-i}{\sqrt{2}}|1\rangle \otimes |+\rangle^{\otimes n} + |0\rangle \otimes \frac{1}{\sqrt{2^{n+1}}}\left(|0\rangle + i|1\rangle\right)^{\otimes n} \xrightarrow[\mathcal{B}_i(\pi/4)]{} \frac{-ie^{-in\pi/4}}{\sqrt{2}}|1\rangle \otimes |+\rangle^{\otimes n} + \frac{1}{\sqrt{2}}|0\rangle \otimes |0\rangle^{\otimes n}$$

and finally, the central mixing term $\mathcal{B}_0$ gives (with $\omega := e^{-in\pi/4}$)

$$\frac{-i\omega}{\sqrt{2}}|1\rangle \otimes |+\rangle^{\otimes n} + \frac{1}{\sqrt{2}}|0\rangle \otimes |0\rangle^{\otimes n} \xrightarrow[\mathcal{B}_0(\pi/4)]{} \frac{-i}{2}|1\rangle \otimes \left(\omega|+\rangle^{\otimes n} - |0\rangle^{\otimes n}\right) + \frac{1}{2}|0\rangle \otimes \left(\omega|+\rangle^{\otimes n} + |0\rangle^{\otimes n}\right)$$

which is the final state $|\psi_f\rangle$. The success probability is then

$$\Pr(\text{success}) = |\langle \mathbf{0}|\psi_f\rangle|^2 = \frac{1}{4}|1 - \omega\langle 0|+\rangle^n|^2 = 1/4 + O(1/2^n) \tag{42}$$

which is a finite constant and may be boosted polynomially close to 1 with a logarithmic number of repetitions.

8.3.3 Other symmetric instances

The success of QAOA1 on the two chosen instances is in part due to the fact that only the potential on the support of the initial state affects the state dynamics. This feature is absent from the other algorithms studied here. Notably, for the adiabatic algorithm on Spike, while it is true that the spectral gap is minimized at the same point u^* as for the ramp without the spike (see [28]), the size of the gap itself depends on the spike parameters, so that in particular, when the spike is sufficiently broad or tall, the gap becomes exponentially small in n. In stark contrast, the performance of QAOA1 is *independent* of the gap parameters, since the state has vanishing support on the spike.

Now, we will use this feature to give conditions under which a symmetric cost function may be successfully minimized by QAOA1. When the cost can be decomposed into a linear ramp and a super-linear part that has small support on the initial state, one may ignore the super-linear terms and treat the problem as a linear ramp. Suppose we have a Hamming-symmetric cost function $c(\tilde{w}) = c_0 + c_1\tilde{w} + c_2\tilde{w}^2 + \cdots$, written as a Taylor series in $\tilde{w}$, the shifted Hamming weight $\tilde{w} = w - n/2$ (which we henceforth replace with w). Separate the function into a linear part and a super-linear part, $c(w) = r(w) + q(w)$, where

$$r(w) = c_0 + c_1 w \tag{43}$$
$$s(w) = c_2 w^2 + \cdots \tag{44}$$

Under Lemma 1, if it is the case that $s(w)$ overlaps weakly with the initial state (which is roughly supported on weights $n/2 \pm O\left(\sqrt{n}\right)$), and if the addition of $s(w)$ does not change the global minimum of $r(w)$, then such a cost function $c(w)$ can be optimized using a "ramp protocol" for $r(w)$, as was done for the Spike problem in Sec. 8.3.1 (provided the slope of the ramp $|c_1| \geq O(1/\text{poly}(n))$).

However, in this case we can do better (Theorem 1 below): even if the global minimum of $c(w)$ does not coincide with that of $r(w)$, the ramp protocol may be suitably modified to ensure the successful minimization of $c(w)$. Suppose $\min_w c(w) = w^*$. For the ramp $r(w) = c_0 + c_1 w$, the first step of QAOA is evolution under $\mathcal{C}(\pi/(2|c_1|))$. For $c(w)$, we modify γ to γ^* (to be determined),

and keep $\beta = \pi/4$ unchanged. Then, the final state may be written as

$$|\psi_f\rangle = \bigotimes_{i=1}^{n} \left(\sin\left(\gamma^*/2\right) |0\rangle + \cos\left(\gamma^*/2\right) |1\rangle \right) \tag{45}$$

$$= \sum_{w=0}^{n} \left(\sin\left(\gamma^*/2\right) \right)^{n-w} \left(\cos\left(\gamma^*/2\right) \right)^{w} \binom{n}{w}^{1/2} |w\rangle \tag{46}$$

Then, by inspection, γ^* must maximize the success probability, or equivalently, the function $\left(\sin\left(\gamma^*/2\right) \right)^{2(n-w^*)} \left(\cos\left(\gamma^*/2\right) \right)^{2w^*}$. An elementary calculation yields that

$$\gamma^* = \arccos \sqrt{\frac{w^*}{n}} \tag{47}$$

Finally, the success probability is

$$\text{Pr(success)} = \frac{(w^*)^{w^*} (n - w^*)^{w^*}}{n^n} \binom{n}{w^*} = O(1) \tag{48}$$

by Stirling's approximation. So, QAOA1 with $\beta = \pi/4, \gamma = \gamma^*$ successfully optimizes the cost function $c(w)$. Finally, we note that if the minimum w^* of c is unknown, the above QAOA1 protocol may be carried out for all $n + 1$ possible values of w^* until success, which is at most a factor $O(n)$ overhead. Therefore, we have just proven the following result:

Theorem 1. *When $c(w) = r(w) + s(w)$ and r is linear in w with slope $\Omega(1/poly(n))$, and $s(w)$ satisfies the weak overlap condition 3 in Lemma 1, $c(w)$ can be successfully minimized via QAOA1 with at most a polynomial number of classical repetitions.*

There is an intuitive picture for the feature of QAOA discovered in Theorem 1 above. As has been observed before [31, 14], the low energy spectrum of the mixing operator B can be mapped to a suitable harmonic oscillator that treats the Hamming weight w as the position variable. Under this mapping, the initial state $|+\rangle^{\otimes n}$ acts as the vacuum state wavepacket, and a linear ramp with slope a, $C = a \sum_w w|w\rangle\langle w|$ is the analogous position operator. We may then qualitatively work out the action of QAOA on the initial wavepacket. The first round, evolution under C, displaces the vacuum to a state with finite momentum $p = a\gamma$. Then, evolution under the harmonic oscillator Hamiltonian B for time $\beta = \pi/2$ rotates the coherent state so that the final state is one that is displaced in w. So, in a single round of QAOA, the wavepacket gains momentum and propagates to a new location in hamming weight space. (This feature has been recently noted in [11].) While the above method recovers the QAOA1 protocol qualitatively, it gets the angle γ wrong by a factor $2/\pi$. This is due to the curvature of the phase space. In fact, the wavepacket is more accurately described by a *spin-coherent* state, in which the conjugate operators are the total spin operators S_x and S_z. It remains to be seen how this (spin-)coherent state picture may be employed to understand the behavior of QAOA on other (especially non-Hamming symmetric) instances. The simplicity of this description suggests a classical algorithm which simulates the momentum transfer and jump operations of the wavepacket via local gradient measurements of the cost function. This could give rise to a new, quantum-inspired classical search heuristic that escapes local minima more efficiently than existing classical methods.

Appendices

Appendix A The control framework

Given a dynamical equation depending on additional parameters (which we call the *controls*), what properties does a control protocol which optimizes a given cost function satisfy? The relevance of this question extends across many fields where optimal control (with respect to a cost function) is desired. In fact, it has been observed [1, 2] that the optimal control problem also applies to heuristic optimization algorithms, where the controlled dynamics are described precisely by Schrödinger evolution under the annealing Hamiltonian, and the cost function is given by the energy of the final state.

Consider a first-order differential equation describing the dynamics of an n-dimensional real vector $x \in \mathbb{R}^n$, and controlled by m control parameters which we denote by the vector $u \in \mathbb{R}^m$:

$$\dot{x}(t) = f(x(t), u(t)) \tag{49}$$

The functional form f may be very general; we only assume that f is "Markovian" (i.e., depends only on the current state $(x(t), u(t))$), and that there is no explicit time-dependence. Typically, it is further assumed that the control u inhabits a fixed, *compact* subset, $u \in \mathcal{U} \subset \mathbb{R}^m$. The domain $\mathcal{U}$ represents a feasible set of controls.

In order to talk about optimal control, we must first specify a notion of cost. In a real problem such as optimizing the trajectory of a spacecraft, the cost might be expressed in terms of time, amount of fuel used (i.e. a trajectory-dependent cost), and the distance of the final position from the target location (i.e., a final state cost). Thus, the cost function may generally be expressed as a (weighted) sum of three costs:

1. the total time for the process, $T = \int_0^T 1 \mathrm{d}t$

2. the *running* cost, which is given as an integral over the running time, $\int_0^T L\left(x(t), u(t), t\right)\mathrm{d}t$

3. the *terminal* cost, which is a final state-dependent function $K(x(T))$.

The full cost function may be expressed in the general form

$$J = K(x_{final}) + \int_0^\infty L\left(x(t), u(t), t\right)\mathrm{d}t \tag{50}$$

where J is a functional of the control schedule $u(t)$ and the dynamical path $x(t)$. The objective is to find the control function $u(t)$, over all piecewise continuous functions $u : \mathbb{R}_{\geq 0} \to \mathcal{U}$, that minimize the overall cost, i.e. $\arg\min_{u(t)} J(u)$. This is the so-called infinite time horizon formulation of the

problem. Alternatively, one can fix the total time for the protocol T to be finite. Then, we are asked to minimize over all piecewise continuous functions $u : [0, T] \to \mathcal{U}$ the cost

$$J = K(x(T)) + \int_0^T L\left(x(t), u(t), t\right) \mathrm{d}t \tag{51}$$

A wealth of literature in classical control theory discusses the question of optimal control, and we emphasize its potential applicability in the setting of designing efficient heuristic optimizers, both classical and quantum. Here, we will focus on one result, the *Pontryagin Minimum Principle* (PMP), which imposes necessary conditions for a control protocol to be optimal using the so-called control Hamiltonian description.

The control Hamiltonian $\mathcal{H}$ is a classical functional describing auxiliary Hamiltonian dynamics on a set of variables given by x and corresponding *co-state* (or conjugate momentum) variables p. The conjugate momenta depend on the cost function J in Eq. 51, and are introduced as Lagrange multipliers that impose the equations of motion for each coordinate of x. The full cost function (at time t), which includes the cost terms in J and the constraints, is given by the control Hamiltonian $\mathcal{H}$.

$$\mathcal{H} := L(x, u) - p \cdot f(x, u) \tag{52}$$

Then, PMP states that the optimal control is one which minimizes the control Hamiltonian at all times. That is,

$$\mathcal{H}\left(x(t), p(t), u^*\right) \leq \min_{u \in \mathcal{U}} \mathcal{H}\left(x(t), p(t), u\right) \tag{53}$$

In the special case when $\mathcal{H}$ is linear in the control u, the above minimality condition is satisfied only if the control lies on the boundary of the feasible set $\mathcal{U}$. This implies that optimal trajectories are bang-bang, i.e., the controls only take their extremal values. The optimal point(s) on the boundary are determined by the intersection of the constant-$\mathcal{H}$ hyperplanes in control space with the set boundary. However, an important exception arises when the derivative of $\mathcal{H}$ with respect to u vanishes over a finite interval. In this case, the control becomes singular, i.e., its optimal value no long lies solely on the boundary of $\mathcal{U}$.

The control framework described here covers many heuristic optimization algorithms, and we will fix some notation to suit this setting. The dynamical vector of interest will a state $|\psi\rangle$ (quantum) or $|\psi)$ (classical), and the generator of dynamics will be a *controlled* linear operator

$$H(\mathbf{u}) = \sum_{i=0}^{m} u_i H_i \equiv \mathbf{u} \cdot \mathbf{H} \tag{54}$$

where $\mathbf{u}$ and $\mathbf{H}$ are vectors with components u_i and H_i respectively. We assume that individual Hamiltonians H_i are time-independent, and only their overall strength, controlled by the coefficient $u_i(t)$, is time-dependent. We will fix the range of all u_i to $[0, 1]$.

Appendix B   Bang-bang simulated annealing on the `Spike`

For `Spike`, the strategy used for `Bush`, namely, run randomized gradient descent (zero-temperature SA) from start to finish, fails due to the presence of a barrier. So, if we run gradient descent for time $O(n)$ per walker, then we are left with a distribution sharply peaked at the false minimum.

We may now attempt to diffuse across the barrier. For a sufficiently wide barrier, this strategy will again fail, since the diffusion rate across the spike is exponentially small in n^a. However, we instead turn on diffusion for a short time, so that a constant fraction of the walkers "hop on" the barrier, while the rest diffuse away from the barrier. Then, we turn on randomized gradient descent again until the finish. The fraction of walkers on the barrier are now guaranteed to walk to the global minimum in time $O(n)$, as the slope is positive.

So, it can be seen that for the spike problem, an algorithm with the same structure as SA but a schedule that is designed without the adiabaticity constraint, successfully finds the global minimum, and thus exponentially outperforms SA (and QAO for certain parameter regimes, see Table 1) on the same instance. It should be noted that the success of BBSA depends sensitively on the shape of the spike. In particular, we expect success (i.e. at least $1/\text{poly}(n)$ walkers reach the global minimum) when the part of the spike with positive slope (i.e. the "uphill" portion) has width $O(\log n)$.

Appendix C   Proof of Lemma 1

Let $\mathcal{C} = e^{-i\gamma \sum_w c(w)|w\rangle\langle w|}$ and let $\mathcal{R}, \mathcal{S}$ be defined analogously with the cost terms $r(w)$ and $s(w)$, where $c(w) = r(w) + s(w)$. $\mathcal{R}$ and $\mathcal{S}$ are mutually commuting, so $\mathcal{C} = \mathcal{RS}$, and the first step of the QAOA1 protocol may be written as

$$\mathcal{C}|\psi_0\rangle = \mathcal{RS}|\psi_0\rangle = \mathcal{R}\sum_{w=0}^{n} e^{-i\gamma s(w)} A_w|w\rangle = \mathcal{R}|\psi_0\rangle + \mathcal{R}\sum_{w=0}^{n}\left(e^{-i\gamma s(w)} - 1\right) A_w|w\rangle \tag{55}$$

After the mixing operator $\mathcal{B} = e^{-i\beta B}$ is applied, the final state is

$$|\psi_f\rangle = \mathcal{BR}|\psi_0\rangle + \mathcal{BR}\sum_{w=0}^{n}\left(e^{-i\gamma s(w)} - 1\right) A_w|w\rangle \tag{56}$$

The overlap with the global minimum $|\psi^*\rangle$ is

$$\langle\psi^*|\psi_f\rangle = \langle\psi^*|\mathcal{BR}|\psi_0\rangle + \langle\psi^*|\mathcal{BR}\sum_{w=0}^{n}\left(e^{-i\gamma s(w)} - 1\right) A_w|w\rangle \tag{57}$$

$$\implies |\langle\psi^*|\psi_f\rangle - \langle\psi^*|\mathcal{BR}|\psi_0\rangle| = |\langle\psi^*|\mathcal{BR}\sum_{w=0}^{n} 2e^{i\gamma s(w)/2 - i\pi/2}\sin\left(\frac{\gamma s(w)}{2}\right) A_w|w\rangle \tag{58}$$

Now, $p = |\langle\psi^*|\mathcal{BR}|\psi_0\rangle|^2$, and let $p^* = |\langle\psi^*|\psi_f\rangle|^2$, the success probabilities of QAOA1 on $r(w)$ and the full cost function $c(w)$, respectively. We wish to show that p^* is at least $p - o(p)$. Using the triangle inequality $|x| - |y| \leq |x - y|$ on the left side of Eq. 58, and Cauchy-Schwarz inequality $|\langle u|v\rangle| \leq |\langle u|u\rangle|^{1/2}|\langle v|v\rangle|^{1/2}$ on the right side, we get the following:

$$\sqrt{p} - \sqrt{p^*} \leq \sum_{w=1}^{n} 4|A_w|^2 \sin^2\left(\frac{\gamma s(w)}{2}\right)^{1/2} = \sqrt{q} \tag{59}$$

$$\implies p^* \geq p\left(1 - \sqrt{q/p}\right)^2 = p\left(1 - o(1)\right)^2 = p - o(p) \tag{60}$$

which proves the lemma.

References

[1] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.

[2] Z. C. Yang, A. Rahmani, A. Shabani, H. Neven, and C. Chamon. Optimizing variational quantum algorithms using pontryagin's minimum principle. *Physical Review X*, 7(2):1–8, 2017.

[3] J. Preskill. Quantum computing in the NISQ era and beyond. *arXiv preprint arXiv:1801.00862*, 2018.

[4] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*, 2000.

[5] E. Farhi, J. Goldstone, and S. Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[6] T. Hogg and D. Portnov. Quantum optimization. *Information Sciences*, 128(3-4):181–197, 2000.

[7] S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *arXiv preprint arXiv:1709.03489*, 2017.

[8] E. Farhi and H. Neven. Classification with quantum neural networks on near term processors. *arXiv preprint arXiv:1802.06002*, 2018.

[9] I. H. Kim and B. Swingle. Robust entanglement renormalization on a noisy quantum computer. *arXiv preprint arXiv:1711.07500*, 2017.

[10] D. Wecker, M. B. Hastings, and M. Troyer. Training a quantum optimizer. *Physical Review A*, 94(2):022309, 2016.

[11] G. Verdon, J. Pye, and M. Broughton. A universal training algorithm for quantum deep learning. *arXiv preprint arXiv:1806.09729*, 2018.

[12] L. S. Pontryagin. *Mathematical theory of optimal processes*. Routledge, 2018.

[13] E. Farhi, J. Goldstone, and S. Gutmann. Quantum adiabatic evolution algorithms versus simulated annealing. *arXiv preprint quant-ph/0201031*, 2002.

[14] L. T. Brady and W. van Dam. Spectral-gap analysis for efficient tunneling in quantum adiabatic optimization. *Physical Review A*, 94(3):032309, 2016.

[15] D. Mitra, F. Romeo, and A. Sangiovanni-Vincentelli. Convergence and finite-time behavior of simulated annealing. *Advances in applied probability*, 18(3):747–771, 1986.

[16] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

[17] B. Hajek. Cooling schedules for optimal annealing. *Mathematics of operations research*, 13(2):311–329, 1988.

[18] B. Gidas. Nonstationary markov chains and convergence of the annealing algorithm. *Journal of Statistical Physics*, 39(1-2):73–131, 1985.

[19] S. Jansen, M.-B. Ruskai, and R. Seiler. Bounds for the adiabatic approximation with applications to quantum computation. *Journal of Mathematical Physics*, 48(10):102111, 2007.

[20] A. Elgart and G. A. Hagedorn. A note on the switching adiabatic theorem. *Journal of Mathematical Physics*, 53(10):102202, 2012.

[21] J. Roland and N. J. Cerf. Quantum search by local adiabatic evolution. *Physical Review A*, 65(4):042308, 2002.

[22] R. D. Somma, D. Nagaj, and M. Kieferová. Quantum speedup by quantum annealing. *Physical review letters*, 109(5):050501, 2012.

[23] R. Chakrabarti and H. Rabitz. Quantum control landscapes. *International Reviews in Physical Chemistry*, 26(4):671–735, 2007.

[24] Armin Rahmani and Claudio Chamon. Optimal control for unitary preparation of many-body states: Application to luttinger liquids. *Physical review letters*, 107(1):016402, 2011.

[25] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. Obrien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5:4213, 2014.

[26] M. Lapert, Y. Zhang, S. J. Glaser, and D. Sugny. Towards the time-optimal control of dissipative spin-1/2 particles in nuclear magnetic resonance. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 44(15):154014, 2011.

[27] B. Bonnard, S. J. Glaser, and D. Sugny. A review of geometric optimal control for quantum systems in nuclear magnetic resonance. *Advances in Mathematical Physics*, 2012, 2012.

[28] L. Kong and E. Crosson. The performance of the quantum adiabatic algorithm on spike hamiltonians. *International Journal of Quantum Information*, 15(02):1750011, 2017.

[29] B. W. Reichardt. The quantum adiabatic optimization algorithm and local minima. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 502–510, New York, NY, USA, 2004. ACM.

[30] E. Crosson and A. W. Harrow. Simulated quantum annealing can be exponentially faster than classical simulated annealing. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 714–723. IEEE, 2016.

[31] J. Bringewatt, W. Dorland, S. P. Jordan, and A. Mink. Diffusion monte carlo versus adiabatic computation for local hamiltonians. *arXiv preprint arXiv:1709.03971*, 2017.