Evolutionary Algorithms for Hard Quantum Control

Ehsan Zahedinejad,1 Sophie Schirmer,2 and Barry C. Sanders1,3,

1Institute for Quantum Science and Technology, University of Calgary, Alberta, Canada T2N 1N4
2College of Science, Swansea University, Singleton Park, Swansea SA2 8PP, Wales, United Kingdom
3Hefei National Laboratory for Physical Sciences at the Microscale,
University of Science and Technology of China, Anhui 230026, China

(Dated: March 6, 2014)

Although quantum control typically relies on greedy (local) optimization, traps (irregular critical points) in the control landscape can make optimization hard by foiling local search strategies. We demonstrate the failure of greedy algorithms to realize two fast quantum computing gates: a qutrit phase gate and a controlled-not gate. Then we show that our evolutionary algorithm circumvents the trap to deliver effective quantum control in both instances. Even when greedy algorithms succeed, our evolutionary algorithm delivers a superior control procedure because less time resolution is required for the control sequence.

PACS numbers: 42.50.Gy, 42.50.Ex

Quantum control aims to steer quantum dynamics towards closely realizing specific quantum states or operations [1,2] with applications to femtosecond lasers [3,4], nuclear magnetic resonance [5,6] and other resonators [7–9], laser-driven molecular reactions [10,11], and quantum gate synthesis for quantum computing [12]. Control is achieved by varying the strengths of different contributing processes (external fields) over time such that the resultant evolution closely approximates the desired evolution. The quality of a given quantum control procedure is quantified by its fitness [13] such as fidelity or distance for the approximated quantum state [14] or quantum gate [15] and the target state or gate.

A key goal in quantum control is to reach the fittest procedure possible within the target time $T$ while minimizing the number $K$ of control parameters representing the time resolution $T/K$ for time-domain quantum control. In practice the upper bound for $T$ is tightly constrained, and lower bounds for $T$ are central to questions about fundamental “quantum speed limits” to operations in quantum computing, quantum metrology and quantum chemistry [16,19].

Choosing control parameters to maximize the procedure fitness is an optimization problem. Early quantum control employed non-greedy approaches such as the genetic algorithm [20,21]. Today greedy algorithms dominate methodology as local optimization strategies usually have lower computational cost than global search algorithms. Furthermore, the fitness landscape (the fitness as a function of control parameters) generally appears to be sufficiently favorable for these algorithms to succeed [22].

Unfortunately, greedy algorithms often fail even for quantum control problems involving low-dimensional systems with simple Hamiltonians when $T$ must be short. This seemingly innocuous constraint eliminates any guarantee we might have regarding the global optimality of local extrema. Similarly, greedy algorithms often struggle to find solutions if the time-resolution of the controls is limited.

Although tempting to attribute failure to find a satisfactory control procedure to infeasibility of the constrained problem, we show that this failure can instead be due to restricting strategies to greedy algorithms. To make our case, we present examples of control problems involving simple systems for which greedy algorithms overwhelmingly fail. We show that these hard quantum control problems can be fully solved using global optimization techniques based on the differential evolution (DE) algorithm [23], which succeeds in finding effective controls up to the computational-power limits (machine error) even for very short $T$ and very few controls.

We compare greedy vs non-greedy algorithms for realizing two different quantum computing gates: the original qutrit phase gate [24,25] and the two-qubit controlled-not (CNot) gate [26], which are key elements of standard quantum computing instructions sets for qutrits and for qubits, respectively. We show that, for each gate and given our selected drift and control Hamiltonians, the greedy algorithm fails to find a high-fitness quantum control procedure for short target time $T$ whereas the non-greedy algorithm succeeds while requiring smaller $K$ than do the greedy algorithms for the case of larger $T$ where the greedy algorithm works.

Mathematically, for a closed system, the Hamiltonian

$$\hat{H} \left[ \varepsilon(t) \right] = \hat{H}^{dr} + \varepsilon(t) \cdot \hat{H}^e = \hat{H}^{dr} + \sum_{\ell=1}^{L} \varepsilon_\ell(t) \hat{H}_\ell^e, \quad (1)$$

acting on Hilbert space $\mathcal{H}$, describes the dynamics [27]. The drift Hamiltonian $\hat{H}^{dr}$ describes the free (uncontrolled) evolution of the system, which we treat as being time-independent here. The control Hamiltonians, represented by the vector operator $\hat{H}^e(t) = (\hat{H}_\ell^e)$ for $\{\ell\}$ the control field labels) should steer the system towards the desired evolution with time varying (here piecewise constant) control amplitudes $\{\varepsilon_\ell(t)\}$ combined in the control vector $\varepsilon(t)$. 
The resultant unitary evolution operator is formally

\[ U[\varepsilon(t);T] = T \exp \left\{ -i \int_0^T H(\varepsilon(t))dt \right\} \quad (2) \]

with \( T \) the time-ordering operator \([25]\). Our aim here is to approximate a target unitary evolution operator \( U \) within duration \( T \) by \( U[\varepsilon(t);T] \) with minimum distance between the realized evolution and the target.

The quality of a candidate quantum control procedure is quantified by the fitness functional

\[ \mathcal{F}[\varepsilon(t)] = \mathcal{F}(U[\varepsilon(t);T]) = 1 - \|U - \tilde{U}[\varepsilon(t);T]\| \quad (3) \]

with \( \| \cdot \| \) the operator norm and the final term in \([4]\) the trace distance \([29]\) between the target and actual evolution operators. The optimization problem is to maximize \( \mathcal{F}[\varepsilon(t)] \), i.e., to reduce the distance \( \|U - \tilde{U}[\varepsilon(t);T]\| \).

Optimization capability is constrained by two considerations: whether the algorithm reaches the desired \( \mathcal{F} \) given unlimited resources and the actual limit subject to resource bounds such as computational run-time.

Evaluating and comparing algorithms for optimization should be conducted fairly and clearly. Using run-time directly as a cost criterion obscures fundamental issues in comparing the intrinsic differences. Therefore, we evaluate and compare algorithms based on whether the algorithm yields a sufficiently optimal solution over many attempts, here called “runs”. Each run is allowed to iterate until it succeeds or fails in which case the run aborts.

The iteration number of run \( r \) is \( i \), and the total number of iterations for run \( r \) is denoted \( I_r \) with \( I_R \) the maximum iteration number over all \( R \) runs. For \( R \) runs, we determine and tabulate the best and worst fitness value obtained over these runs, and we characterize the statistics of error values according to the median error and the probability \( \wp \), or percentage, of runs whose error is less than some threshold value.

We compare the performance of the optimization algorithms based on the bounds and statistics of the statistics of runs, and these statistics are analyzed in plots that depict the fidelity vs \( i \) for each of the many runs. A plot of \( \mathcal{F} \) vs \( i \) is overly crowded to reveal clearly the key features. Therefore, we stretch the plots by introducing the monotonically related “logarithmic infidelity” \( L := \log_{10}(1-\text{Re}(\mathcal{F})) \) plotted against \( i \) for each run.

Logarithmic infidelity is zero for perfect infidelity \( (\mathcal{F} = 0) \), approximately bounded by \(-16\) for machine precision, and ideally \(-\infty\) for perfect fidelity \( (\mathcal{F} = 1) \). The algorithm for run \( r \) is deemed successful if the final \( \mathcal{F}_r \) exceeds a minimum threshold \( L^k \), which we set to \(-4\) commensurate with the widely accepted gate fidelity required for scalable quantum computing \([30]\). Our algorithm aborts a run after \( I_r \) iterations only if \( L^k \) is reached or if \( \mathcal{F} \) converges to a constant within machine error. The percentage of runs that beat \( L^k \) is denoted \( \wp^k \).

Now we discuss how the computation works. Numerically this fitness functional is evaluated by discretizing the control function vector \( \varepsilon \) by expressing it as a sum of \( K \) orthonormal functions over the time domain \([0,T]\). The \( K \) control parameters refer to choosing various weightings of these control functions. For our analysis, these \( K \) orthonormal functions are non-overlapping rectangular functions with identical durations \( T/K \); i.e., the control functions are expressed as a weighted series of time bins. An appealing alternative is to decompose into \( K \) monochromatic functions to solve in the frequency domain.

For rectangular time bins, the time-ordered integral \([2]\) is straightforward to evaluate and then to exponentiate. The next step is to optimize \( \mathcal{F} \) over \( \varepsilon(t) \) within target time \( T \) with the number \( K \) of time bins being small. Greedy algorithms are tools of choice nowadays, but non-greedy algorithms such as evolutionary algorithms can be a viable alternative when greedy algorithms fail as we aim to show here for constrained quantum control.

Greedy algorithms include the Nelder-Mead technique \([32]\) and the quasi-Newton method, which employs the Broyden-Fletcher-Goldfarb-Shanno (BFGS) approximation of the Hessian \([33,37]\). These greedy algorithms commence with a guessed \( \varepsilon(t) \) and then use the gradient and Hessian of the Hamiltonian to find local optima over many iterations. Lie group techniques can help to determine the gradient analytically \([35,39]\), and numerical techniques work generally. Greedy algorithms are especially successful if both \( T \) and \( K \) are sufficiently large. For constrained \( T \), second-order traps such that the Hessian is negative semidefinite arise \([40,42]\) and numerical evidence arguably exists for the presence of other traps \([13,43,44]\).

As an alternative to greedy algorithms, we favor evolutionary algorithms for quantum control. Evolutionary algorithms are stochastic optimization algorithms, inspired by the process whereby biological organisms adapt and evolve \([45]\). These algorithms require less information than greedy algorithms: they only require the fitness functional and not the onerous inputs of its first \((\nabla \mathcal{F})\) and second derivatives \((\nabla^2 \mathcal{F})\) with respect to the elements of control function \( (\varepsilon(t)) \). We employ the two most successful types of evolutionary algorithms here: Particle Swarm Optimization (PSO) \([46]\) and DE \([23]\), which are superior to historic approaches such as genetic algorithms \([47]\).

Both PSO and DE employ the initial condition of multiple guesses (test functions) known as particles in PSO and chromosomes in DE. Each test function evolves iteratively along trajectories in parameter space and experiences different fitness values. In PSO the particle evolves according to a Langevin equation that includes a random kick, an attractive force to its previous best fitness, a force pulling to the particle to the fittest particle in its neighborhood (where the size of the neighborhood is log-
arithmetic in the number of particles), and neighborhoods overlap such that they do not partition into distinct sets.

In DE each chromosome breeds with three other randomly chosen chromosomes from the same generation to produce a ‘daughter’, and the fittest of the original vs the bred ‘daughter’ survives to the next generation. Specifically we employ three PSO variants labeled here as PSO1, PSO2 [48] and PSO3 [49]. We also use a DE variant that incorporates mutation scaling factor μ and crossover rate ξ [23]. Mutation scaling factor μ is constant factor from [0, 2].

In each generation the difference between two randomly chosen target vectors is weighted by factor μ and added to a third randomly selected target vector to generate the new set of vectors called donors. This factor also determines the step size that DE explores the configuration space. Elements of donor vector incorporates into the target vectors by a probability of μ so to generate the trial vectors. The fittest among the target and trial vectors succeed to the next generation.

Now we proceed to the two quantum control examples beginning with the original qutrit phase gate. The Hilbert space is H = span{0, 1, 2}, and the target gate is U = e^{-iR\hat{H}t} (e^{i\phi}|0\rangle - i e^{-i\gamma}|1\rangle - e^{i\gamma}|2\rangle|2\rangle) with objective parameters corresponding to the phases φ and γ and Hdr is the drift Hamiltonian defined in (4). As our interest is in hard quantum control problems, we choose a challenging T-dependent drift Hamiltonian and a single control given by [13]

\[
\hat{H}_{\text{dr}} = \begin{pmatrix} 1 + \frac{\pi}{3} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \quad \hat{H}_{c} = \begin{pmatrix} a & 1 & 0 \\ 1 & b & 1 \\ 0 & 1 & c \end{pmatrix},
\]

respectively.

This choice of control and drift Hamiltonian provides a rich lode for studying hard quantum control because, for any target time T, many choices of a, b, c, φ, and γ lead to ε(t) ≡ 0 being a critical point. The resultant criticality results in Re[\mathcal{F}[\varepsilon(t)]] < 1 for which the Hessian becomes strictly negative definite. We consider the specific choice a = 2, b = 2, and c = 1. Then the phase choices γ = \frac{\pi}{3} and sin φ = -\frac{b+2c}{a} ensure that ε(t) ≡ 0 is a critical point i.e. a point in which \nabla^2 \mathcal{F}(\varepsilon(t)) = 0 and \nabla^2 \mathcal{F}(\varepsilon(t)) < 0. Therefore, a strong trap in the fitness landscape is deliberately set [13].

The second example concerns the two-qubit CNot gate [26]. Inspired by the one-dimensional linear Ising-ZZ model [50], the drift and control Hamiltonians are

\[
\hat{H}_{\text{dr}} = J/2 Z \otimes Z, \quad \hat{H}_{c} = \begin{pmatrix} X & \mathbb{I} \\ \mathbb{I} & X \end{pmatrix}, \quad \hat{H}_{c} = \begin{pmatrix} X \otimes \mathbb{I} \\ \mathbb{I} \otimes X \end{pmatrix}, \quad \hat{H}_{c} = \begin{pmatrix} Y \otimes \mathbb{I} \\ \mathbb{I} \otimes Y \end{pmatrix},
\]

respectively, for

\[
X = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad Y = \frac{1}{2} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.
\]

The second example concerns the two-qubit CNot gate [26]. Inspired by the one-dimensional linear Ising-ZZ model [50], the drift and control Hamiltonians are

\[
\hat{H}_{\text{dr}} = J/2 Z \otimes Z, \quad \hat{H}_{c} = \begin{pmatrix} X & \mathbb{I} \\ \mathbb{I} & X \end{pmatrix}, \quad \hat{H}_{c} = \begin{pmatrix} X \otimes \mathbb{I} \\ \mathbb{I} \otimes X \end{pmatrix}, \quad \hat{H}_{c} = \begin{pmatrix} Y \otimes \mathbb{I} \\ \mathbb{I} \otimes Y \end{pmatrix},
\]

respectively, for

\[
X = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad Y = \frac{1}{2} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.
\]

respectively, for

\[
X = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad Y = \frac{1}{2} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.
\]

the Pauli matrices and \(\mathbb{I}\) the 2 \times 2 identity matrix. We choose \(J = 1\) (equivalent to chosen time units in 1/J).

The time-dependent four-dimensional control vector \(\varepsilon(t)\), explained in Eq. (1), is optimized so that the resultant evolution \(\hat{U}(\varepsilon(t))\) approximates CNot with high \(\mathcal{F}\).

Now we generate and analyze the results. As evolutionary algorithms are typically more computationally expensive than are local optimization algorithms, we let \(R = 40\) for evolutionary algorithms and the much larger \(R = 80\) for greedy cases. These choices work well for our Matlab® (version R12118) implementation; other choices would be made for different software.

Figure 1 depicts the logarithmic infidelity \(\{L_r\}\) as a function of \(r\) for each \(r\) given various parameters, corresponding to (a) the qutrit phase gate and to (b) the CNot gate using the quasi-Newton method. In both cases, most quasi-Newton runs converge rapidly within machine precision \((L = -15.65)\) to the target gate for large \(T\) and for small time resolution \(T/K\). For small \(T\) and \(K\), a majority of runs becoming trapped at low fitness (high \(L\)) values. Evidently the quasi-Newton method fails for short-time and fine-time-resolution constraints.

In Fig. 2 we compare the greedy simplex and quasi-Newton methods against the evolutionary DE, Common PSO, PSO1, PSO2, and PSO3 algorithms. Specifically we depict best-run performance and median-run performance in terms of final \(L\). For this constrained \(T\) and low \(K\) problem, the greedy algorithms perform poorly and so does PSO. PSO variants were superior in its rate of convergence but most runs are nonetheless trapped. Fortunately the DE algorithm performs extraordinarily well and dramatically outperforms the other algorithms we test here.

These plots are indicative only. Careful comparisons are summarized in Table III for the qutrit case and in Table IV for median, best-case, and worst-case performance as well as for the percentage of runs \(\psi^3\) that exceed \(L^i\) over \(R = 40\) runs for evolutionary algorithms and over \(R = 80\) runs for greedy algorithms.
For these qutrit and CNOT gates, all the runs get trapped at poor fidelities for the greedy-algorithm case, and the various PSO algorithms are also poor. In contrast the DE performance is vastly superior for the qutrit phase gate and significantly better for the CNOT gate under the extreme conditions of $T = 3.2$ and $K = 4$. Naturally the greedy and PSO algorithms can be improved by increasing $K$, and this strategy is common in the quantum control literature, but our aim is to constrain $T$ and limit the number of control parameters $K$, and DE is the superior tool for doing so in that it works when the greedy and PSO algorithms fail.

In conclusion we have shown that evolutionary algorithms are an essential alternative to greedy algorithms for hard quantum control problems with strong constraints. Greedy algorithms are widely used in this field as fitness landscapes are regarded as well-behaved [22], and traps, although possible, appear to be negligible if $T$ can be long and the number of control parameters $K$ can be increased without paying a significant price. In such cases greedy algorithms work because most local optima are globally optimal or close enough. However, when resources are limited, even straightforward control problems for simple systems can become hard due to a proliferation of traps in the landscape and non-convexity, thereby causing greedy algorithms to fail.

We have considered two quantum gates relevant to quantum information and used drift and control Hamiltonians that illustrate our point. These examples show that differential evolution is effective for hard quantum control problems. The superiority of differential evolution over greedy algorithms is unsurprising because the fitness landscape is no longer well behaved for hard quantum control. On the other hand, the superiority of differential evolution over PSO and its variants is due to the greater efficacy of DE for optimization over higher-dimensional search spaces, which is the case for hard quantum control. Finally we emphasize that, when greedy algorithms work, the quantum control strategy should be to employ current practice and use the best available greedy algorithm. When greedy algorithms fail, though, evolutionary algorithms could work and differential evolution is the best amongst these according to our investigation. This is particularly relevant when exploring quantum speed limits numerically. In view of our results, quantum speed limits found using greedy algorithms reflect the limitations of these algorithms rather than intrinsic speeds limits for quantum control.

EZ acknowledges a Murray Fraser Memorial Graduate Scholarship and Eyes High International Doctoral Scholarship and support from NSERC. SS and BCS acknowledge valuable discussions with A. Pechen and Y. R. Sanders during the early stages of this work. This project was initiated at a Kavli Institute for Theoretical Physics Workshop and thus supported in part by the National Science Foundation under Grant no. NSF PHY11-25915.

| $L$ | DE | PSO | PSO1 | PSO2 | PSO3 | Newton | Simplex |
|-----|----|-----|------|------|------|--------|---------|
| median | -5.6 | -1.2 | -2.2 | -2.1 | -1.0 | -1.0 |
| best-case | -8.3 | -1.2 | -3.6 | -3.3 | -1.4 | -1.4 |
| worst-case | -2.2 | -0.7 | -1.3 | -1.3 | -0.8 | -0.8 |
| $\phi^1$ | 72.5 | 0 | 12.5 | 7.5 | 0 | 0 | 0 |

TABLE I: Median, best-case, worst-case, and $\phi^1$ for logarithmic-infidelity $L$ for the qutrit phase gate with $T = 2.5\pi$, $K = 10$, and $R = 80$ for greedy algorithms and $R = 40$ for evolutionary algorithms.

| $L$ | DE | PSO | PSO1 | PSO2 | PSO3 | Newton | Simplex |
|-----|----|-----|------|------|------|--------|---------|
| median | -2.9 | -1.8 | -2.4 | -2.3 | -0.7 | -2.4 | -1.45 |
| best | -5.5 | -2.9 | -4.2 | -4.7 | -1.0 | -3.9 | -2.4 |
| worst | -2.0 | -1.3 | -1.9 | -1.3 | -0.6 | -2.0 | -0.8 |
| $\phi^1$ | 15.0 | 0 | 2.5 | 10.0 | 0 | 0 | 0 |

TABLE II: Median, best-case, worst-case, and $\phi^1$ for logarithmic-infidelity $L$ for the CNOT gate with $T = 3.2$, $K = 4$, and $R = 80$ for greedy algorithms and $R = 40$ for evolutionary algorithms.
