

Binary Optimal Control Of Single-Flux-Quantum Pulse Sequences

Ryan H. Vogt^a, N. Anders Petersson^a

^a*Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94550, USA.*

Abstract

We introduce a binary, relaxed gradient, trust-region method for optimizing pulse sequences for single flux quanta (SFQ) control of a quantum computer. The pulse sequences are optimized with the goal of realizing unitary gate transformations. Each pulse has a fixed amplitude and duration. We model this process as a binary optimal control problem, constrained by Schrödinger's equation, where the binary variables indicate whether each pulse is on or off. We introduce a first-order trust-region method, which takes advantage of a relaxed gradient to determine an optimal pulse sequence that minimizes the gate infidelity, while also suppressing leakage to higher energy levels. The proposed algorithm has a computational complexity of $\mathcal{O}(p \log(p))$, where p is the number of pulses in the sequence. We present numerical results for the H and X gates, where the optimized pulse sequences give gate fidelity's better than 99.9%, in ≈ 25 trust-region iterations.

Keywords: ODE-constrained optimization, integer optimal control, quantum mechanics, nonlinear programming

AMS: 65K10, 49M37, 34L40, 90C10

1. Introduction

Over the past decades the development of algorithms for a quantum computer, which differ from algorithms for classical computers, and need particular special treatment, brought a great practical interest to quantum computers [1, 2]. Several high impact problems such as combinatorial optimization [3], quantum cryptography[4], and large dimensional linear systems [5] are a few notable examples that motivate the interest in developing quantum computing technologies.

Quantum algorithms become attractive when a quantum computer has a large amount of qubits, however, state of the art quantum computers currently only have on the order of hundreds of qubits; and these qubits do not support error correction. Many quantum algorithms demand at least one million to one-hundred million qubits [6] to be an attractive alternative to classical computing. A popular approach to building a quantum computer is taking advantage of superconducting electronic circuits, which is referred to as superconducting quantum computing. While the superconducting circuits reside inside a dilution refrigerator at milli-Kelvin temperatures, conventional qubit control pulses consisting of modulations of a microwave carrier tone, are generated outside the fridge at room temperature. As a result, the control waveform that is delivered to the qubit is the convolution of the applied waveform with the transfer function of the wiring into the refrigerator. This transfer function is known to depend on the frequency of the carrier wave, but is hard to characterize precisely. An additional challenge from having separate control lines for each qubit is the significant heat load on the milli-Kelvin environment inside the dilution fridge, see [7] for further discussions.

An alternative approach to controlling an array of qubits is by using a classical co-processor that resides inside the fridge. This approach to coherent control involves irradiation of the qubit with trains of quantized flux pulses derived from the single-flux-quantum (SFQ) digital logic family [8]. Here, classical

*Corresponding author

URL: vogt9@llnl.gov (Ryan H. Vogt), petersson1@llnl.gov (N. Anders Petersson)

bits of information are stored as the presence or absence of a phase slip across a Josephson junction in a given clock cycle, where each phase slip corresponds to a voltage pulse whose time integral equals $\Phi_0 = h/2e$, i.e., the superconducting flux quantum. These circuits have gained the attention of the quantum computing community for several reasons: (i) SFQ circuits are a promising candidate for scalable quantum computing, i.e many qubit systems; and (ii) SFQ circuits can realize gate and measurement fidelity to high accuracy [6]. SFQ circuits offer one possible solution to realizing these many qubit systems. One limitation of the SFQ technology is that calculations are done by applying a sequence of pulses of fixed amplitude. This means to conduct computations on a SFQ quantum computer pulses must be administered in a particular manner to achieve desired unitary gate transformations.

The study of controlling quantum mechanical systems in an optimal manner, referred to as quantum optimal control theory, has increasingly become an important tool for developing optimal control strategies that yield a favorable outcome in a quantum mechanical system. The performance of a control strategy can be gauged by a preferred metric, such as realizing gates to high fidelity. We refer the reader to [9] and the references therein, for a review of the current state-of-the-art of quantum optimal control in different domains, such as atomic, molecular, and chemical physics, magnetic resonance, and quantum information and communication. We make special note that [10], and the references therein, discusses the state of the art approach for solving quantum optimal control problems using reinforcement machine learning. Reinforcement machine learning is attractive in the quantum computing setting because unlike supervised machine learning, there is no explicit need to have training data. Generating data, even for established quantum technologies, could still require expensive experiments to be conducted. Some quantum architectures that are of interest to researchers are still theoretical so data cannot be gathered in those instances. We include this reference so that i) the reader is aware of the effectiveness machine learning has had on the field of quantum optimal control for realizing robust high fidelity one and two qubit gates; and ii) while machine learning approaches for solving the SFQ problem is out of the scope of this paper, we wish to not dismiss the idea that machine learning approaches could yield effective control strategies for the SFQ problem. We favor leveraging the binary structure that appears in the problem to develop an specialized trust-region method for the SFQ architecture that yields quality control strategies quickly.

In this paper we introduce a binary optimal control problem to determine the manner in which to apply pulse sequences to the SFQ circuit so that gates are realized to high fidelity. In this work we focus on controlling only a single qubit. We model the control as a binary vector of fixed size, where each element of the binary vector represents the decision to apply a pulse, or no to apply a pulse, to the SFQ circuit. Note that the SFQ architecture only provides two control options in each time interval: either apply a pulse with predefined amplitude and duration, or not apply a pulse, i.e., a pulse with zero amplitude. Since we are deciding on a collection of fixed amplitude pulses to realize a gate, rather than a pulse in which the amplitude varies in time, the popular GRAPE [11] algorithm is not directly applicable. To solve the optimal control problems with binary variables there are several standard approaches: genetic algorithms [12, 13, 14], the branch and bound algorithm [15, 16], and annealing [17]. Genetic algorithms have already been utilized for finding pulse sequences in the SFQ problem [6], but a quality solution is not guaranteed. To the best of our knowledge annealing has not been applied to the SFQ problem, but like genetic algorithms, would not guarantee a quality solution. Branch and bound on the other hand does offer a notion of finding a quality solution of the binary optimal control problem. However in practice a large amount of binary variables will make the method computationally intractable. In this paper we introduce a first-order trust-region method that is computationally tractable, as exemplified by optimizing pulse sequences for realizing H and X gates.

The paper is outlined as follows: In Section 2 we introduce a truncated modal expansion for Schrödinger equation to model a quantum system of one qubit, and the Hamiltonian we use to model a SFQ quantum circuit using SFQ pulses. In Section 3 we introduce the optimal control problem for the SFQ quantum circuit; here, we define the objective function of the optimal control problem that is the sum of an gate infidelity term and a leakage suppression term. This objective function balances the interests of realizing gates to high infidelity, while also suppressing leakage to the guard states. In Section 4 we introduce a first-order trust-region method to solve the SFQ optimal control problem. In Section 5 we solve the

optimal control problem to realize two quantum gates to high fidelity: the H and X gate. In addition, we explore how the tip angle, which we define in the paper, impacts gate realization. In Section 6 we summarize our findings.

2. SFQ optimal control problem

Consider a truncated modal expansion of the Schrödinger equation with $N > 0$ states for realizing a gate transformation. The truncation of the expansion is justified by discouraging population of the states with the highest energy levels. The gate transformation is defined in the "essential" subspace corresponding to the $E > 0$ lowest energy levels and we let $G = N - E \geq 0$ denote the number of "guard" states. The evolution of the state vector satisfies $\psi(t; \boldsymbol{\alpha}) = U(t; \boldsymbol{\alpha})\psi(0; \boldsymbol{\alpha})$, where the $N \times N$ complex-valued solution operator matrix $U(t; \boldsymbol{\alpha})$ satisfies Schrödinger's equation in matrix form:

$$\frac{dU}{dt} + \frac{i}{\hbar}H(t; \boldsymbol{\alpha})U = 0, \quad t \in \Omega = [0, T], \quad U(0) = I_N. \quad (1)$$

Here, \hbar is the reduced Plank constant, $i = \sqrt{-1}$ is the imaginary unit, and $T > 0$ is the duration of the gate transformation. The $N \times N$ identity matrix is denoted I_N and represents the canonical basis for the N -dimensional state vector. The Hamiltonian matrix is $H(t; \boldsymbol{\alpha})$, in which the time-dependence is parameterized by the p -dimensional binary control pulse vector: $\boldsymbol{\alpha} \in \{0, 1\}^p$, $p \geq 1$. As a result, the solution operator matrix U depends implicitly on $\boldsymbol{\alpha}$. The Hamiltonian matrix is assumed to be of the form

$$H(t) = H_0 + H_{SFQ}(t),$$

where the system Hamiltonian is modeled by

$$\frac{H_0}{\hbar} = \omega a^\dagger a - \frac{\xi}{2} a^\dagger a^\dagger a a.$$

Here, a and a^\dagger are the lowering and raising operators. The constants ω_a and ξ represent the qubit's fundamental frequency and anharmonicity (self-Kerr coefficient) respectively. The control Hamiltonian is given by

$$H_{SFQ}(t) = \frac{C_c}{C} v(t) \hat{Q}, \quad \hat{Q} = 2e i(a - a^\dagger)$$

Here, C_c and C are the coupling and Josephson capacitance, respectively, and e is the electron charge constant. We divide time into pulse intervals of length $\tau_p > 0$, which we refer to as the SFQ time steps, with $t_k = k\tau_p$, for $k = 0, 1, \dots, p$, such that $T = p\tau_p$. We model $v(t)$, which represents the voltage of the pulse delivered to the system in the following manner:

$$v(t) = \sum_{k=1}^p \alpha_k \hat{B}(t - t_{k-1}), \quad t \in [t_{k-1}, t_k].$$

We use a quadratic B-Spline basis function [18, 19] $\hat{B}(t)$ to represent the pulse delivered at each SFQ time step. Here, $[\boldsymbol{\alpha}]_k = \alpha_k$ for $k = 1, \dots, p$ is a binary variable that indicates whether the pulse is on or off during the interval $t \in [t_{k-1}, t_k]$,

$$\alpha_k = \begin{cases} 1, & \text{pulse on} \\ 0, & \text{pulse off.} \end{cases}$$

If the pulse is on, the function $v(t)$ is scaled such that its strength corresponds to one magnetic flux quanta,

$$\int_{t_k}^{t_k + \tau_p} v(t) dt = \Phi_0.$$

It is convenient to scale the time-function such that

$$\tilde{v}(t) = \Phi_0 v(t).$$

Because the magnetic flux quanta is defined by $\Phi_0 = 2\pi\hbar/(2e)$,

$$\frac{C_c}{C} \Phi_0 v(t) 2e i(a - a^\dagger) = \hbar \beta \tilde{v}(t) i(a - a^\dagger), \quad \beta = 2\pi \frac{C_c}{C}.$$

Thus,

$$\frac{1}{\hbar} H_{SFQ}(t) = \beta \tilde{v}(t) i(a - a^\dagger).$$

$$\tilde{v}(t) = \frac{1}{\gamma} \sum_{k=1}^p \alpha_k \hat{B}(t - t_{k-1}), \quad t \in [t_{k-1}, t_k], \quad \gamma = \int_0^\delta \hat{B}(t) dt.$$

Here $0 < \delta \leq \tau_p$ is the pulse duration, which defines the support of $\hat{B}(t)$. By defining $\tilde{v}(t)$ in this manner we ensure that

$$\int_{t_{k-1}}^{t_k} \tilde{v}(t) dt = 1,$$

if the pulse is on, which corresponds to applying a pulse with strength β . We illustrate a single pulse in Figure 1 with SFQ time step $\tau_p = 2.5 \times 10^{-2}$ nanoseconds (ns) and pulse duration $\delta = 4 \times 10^{-3}$ ns.

For the case of $E = 2$ essential states in a system, the Bloch sphere provides a convenient geometric representation of single qubit transformations [20]. We illustrate the Bloch sphere in Figure 2. Starting from the ground state (represented by the North pole on the Bloch sphere) one SFQ pulse will perturb the state away from the North pole. In terms of spherical coordinates, the corresponding change in polar angle will be referred to as the tip angle θ , which is related to the pulse strength by $\beta = \theta/\pi\tau_p$.

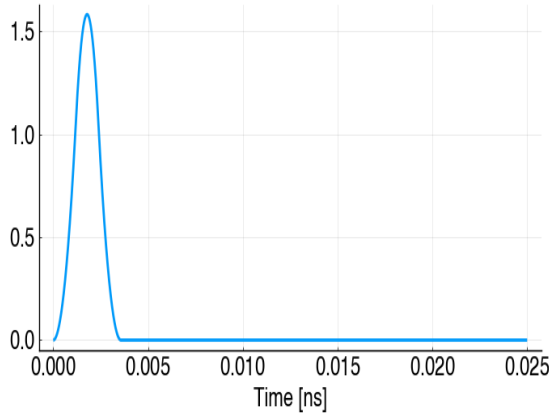


Figure 1: Illustration of a B-spline pulse function with SFQ time step $\tau_p = 2.5 \times 10^{-2}$ ns and pulse duration $\delta = 4 \times 10^{-3}$ ns.

3. Objective and gradient computation

In this section we introduce the objective function for our optimal control problem. We use this objective function to write a closed form expression for the gradient with respect to the binary control vector α .

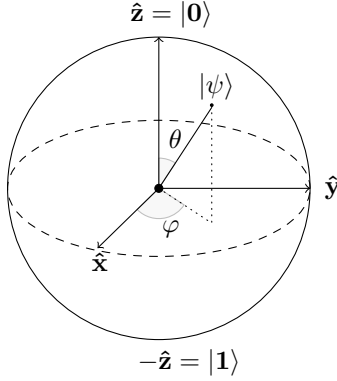


Figure 2: Illustration of the Bloch sphere.

In this work we denote the column vectors of the solution operator matrix by

$$U = [\psi_0 \quad \psi_1 \quad \cdots \quad \psi_{N-1}] \in \mathbb{C}^{N \times N}, \quad \psi_j = \begin{bmatrix} \psi_{0,j} \\ \psi_{1,j} \\ \vdots \\ \psi_{N-1,j} \end{bmatrix} \in \mathbb{C}^{N \times 1}.$$

The unitary target gate transformation V_E is only defined for the first E rows and columns of $U(T, \alpha)$. For notional simplicity we embed it in the blocked matrix V :

$$V = \begin{bmatrix} V_E & \mathbf{0}_{E \times G} \\ \mathbf{0}_{G \times E} & \mathbf{0}_{G \times G} \end{bmatrix} \in \mathbb{C}^{N \times N}.$$

The overlap between the solution operator matrix and the target gate matrix can be measured by the gate infidelity:

$$\mathcal{J}_1(U_T(\alpha)) = 1 - \frac{1}{E^2} |\langle U_T(\alpha), V \rangle_F|^2 \geq 0, \quad U_T(\alpha) := U(T, \alpha), \quad (2)$$

where $\langle \cdot, \cdot \rangle_F$ denotes the Fröbenius matrix scalar product (with associated norm $\|\cdot\|_F$). Because the G last rows and columns of V are zero, only the first E rows and columns of U_T matter for this part of the objective function. We refer to \mathcal{J}_1 as the infidelity term in the remainder of this paper. In this work we strive to construct pulse sequences that cause the infidelity to be below 10^{-3} . This is equivalent to requiring the gate fidelity to be larger than 99.9%.

An important observation we make is that α_k is either zero or one for $k = 1, \dots, p$. This means that we only need to solve Schrödinger's equation (1) for the duration of one SFQ time step in two cases: when the pulse is on, or when it is off. We denote by D_0 the solution corresponding to when the pulse is off, and by D_1 when the pulse is on. We can then represent the solution of the state equation at the final time, $t = T$, as

$$U(T, \alpha) = D_{\alpha_p} D_{\alpha_{p-1}} \cdots D_{\alpha_1},$$

which also implies

$$U_j = U(t_j) = \begin{cases} D_{\alpha_j} D_{\alpha_{j-1}} \cdots D_{\alpha_1}, & j \geq 1, \\ I, & j = 0. \end{cases} \quad (3)$$

This is a helpful observation because it means that, based on the unitary matrices D_0 and D_1 , the solution operator matrix can be evaluated by p matrix multiplications, where p is the number of SFQ pulses. Furthermore, the matrix D_0 can be calculated by matrix exponentiation. Hence, only the matrix

D_1 needs to be calculated through numerical time stepping. This only needs to be performed once, in a pre-processing stage of the algorithm.

The population of the guard states can be measured by evaluating the following expression:

$$\frac{1}{T} \int_0^T \sum_{j=0}^{E-1} \langle \psi_j(t, \boldsymbol{\alpha}), W \psi_j(t, \boldsymbol{\alpha}) \rangle_2 dt = \frac{1}{T} \int_0^T \langle UP, WUP \rangle_F, \quad (4)$$

where the rectangular matrix

$$P = \begin{bmatrix} I_E \\ 0_G \end{bmatrix} \in \mathbb{R}^{N \times E}$$

is used to single out the first E columns of the matrix that is applied to it. Further, $W \in \mathbb{R}^{N \times N} \geq 0$ is a diagonal positive semi-definite weight matrix where only the rows of W corresponding to the guard states have positive elements. For example, if $E = 2$ and $G = 2$, the weight matrix may be

$$W = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & w_1 & 0 \\ 0 & 0 & 0 & w_2 \end{bmatrix}, \quad w_1, w_2 \geq 0.$$

We approximate (4) with the trapezoidal rule, using τ_p as the time step, which results in the following expression:

$$\mathcal{J}_2 = \frac{1}{p} \left(\frac{1}{2} \langle U_0 P, W U_0 P \rangle_F + \sum_{j=1}^{p-1} \langle U_j P, W U_j P \rangle_F + \frac{1}{2} \langle U_p P, W U_p P \rangle_F \right). \quad (5)$$

Here, U_j is defined by (3). In the following, we refer to \mathcal{J}_2 as the leakage term.

For the quantum control problem with guard states, we formulate the optimization problem as

$$\min_{\boldsymbol{\alpha}} \mathcal{J}(\boldsymbol{\alpha}) := \mathcal{J}_1(U_T(\boldsymbol{\alpha})) + C_1 \mathcal{J}_2(U(\cdot, \boldsymbol{\alpha})), \quad C_1 > 0,$$

where $U(t, \boldsymbol{\alpha})$ is governed by (1). From numerical experimentation we find that $C_1 = 10^{-2}$ helps insure that both terms of \mathcal{J} are of the same order of magnitude during the optimization. We note that this optimization problem is a large dimensional nonlinear integer programming problem (NIP), since $\boldsymbol{\alpha}$ is a binary (integer) vector, $\mathcal{J}(\boldsymbol{\alpha})$ is nonconvex. NIP's are typically in the class of computations problems that are non-polynomial (NP), which is to say that there does not currently exist a polynomial time algorithm to solve them. This causes large dimension NIP's to be extremely hard to solve in practice. We discuss more about NIP's in Section 4, and propose a trust-region algorithm to solve the NIP that arises from the SFQ optimal control problem. An overview of the state-of-the-art theory, and numerical solutions for NIP's can be read in [21, 22, 23, 24].

Next, we discuss the gradient computation of the objective function. We note that,

$$\frac{\partial \mathcal{J}}{\partial \alpha_k} = \frac{\partial \mathcal{J}_1}{\partial \alpha_k} + C_1 \frac{\partial \mathcal{J}_2}{\partial \alpha_k}.$$

In this work we define $A_j = D_{\alpha_j}$ and $B_j = \partial D_{\alpha_j} / \partial \alpha_j$. The latter expression is defined in the sense of a relaxed gradient that is calculated by temporarily assuming that α_j is real, with $0 \leq \alpha_j \leq 1$. The continuous gradient can then be defined and evaluated at its upper and lower bounds, corresponding to the binary variable $\alpha_j \in \{0, 1\}$. In the following, the gradient of the objective function is to be understood in the sense of this relaxed gradient.

In Appendix Appendix A, we derive the expressions for the gradients of \mathcal{J}_1 and \mathcal{J}_2 . Here we only present the main results. To calculate the gradient of the infidelity term \mathcal{J}_1 , we first note that (2) can be written

$$\mathcal{J}_1 = 1 - \frac{1}{E^2} S_T \bar{S}_T, \quad S_T = \langle U_p P, V P \rangle_F, \quad (6)$$

where \bar{S}_T denotes the complex conjugate of S_T . The gradient can then be expressed as

$$\frac{d\mathcal{J}_1}{d\alpha_{p-q}} = -\frac{2}{E^2} \text{Re} \left(\bar{S}_T \langle B_{p-q} U_{p-q-1} P, \Lambda_{p-q} P \rangle_F \right), \quad q = 0, \dots, p-1,$$

where the discrete adjoint variable Λ_k satisfies

$$\Lambda_{p-q} = \begin{cases} \Lambda_p = V, & q = 0, \\ \Lambda_{p-q} = A_{p-q+1}^\dagger \Lambda_{p-q+1}, & q = 1, \dots, p-1. \end{cases}$$

The gradient of the leak term can be expressed as

$$\frac{d\mathcal{J}_2}{d\alpha_{p-q}} = \frac{2}{p} \text{Re} \left\langle B_{p-q} U_{p-q-1} P, \tilde{\Lambda}_{p-q} P \right\rangle_F, \quad q = 0, \dots, p-1.$$

In this case, the discrete adjoint variable is calculated according to

$$\tilde{\Lambda}_{p-q} = \begin{cases} \frac{1}{2} W U_p P, & q = 0, \\ W U_{p-q} P + A_{p-q+1}^\dagger \tilde{\Lambda}_{p-q+1}, & q = 1, 2, \dots, p-1. \end{cases}$$

4. Trust-region method

Trust-region methods are a popular approach for solving continuous nonlinear constrained optimization problems [25]. Given an initial feasible solution to the constrained optimization problem at iteration k , a first or second-order Taylor series approximation is written for an objective function $f(\mathbf{x})$ to be minimized i.e

$$f(\mathbf{x}) \approx \tilde{f}(\mathbf{x}) = f(\mathbf{x}_k) + \nabla_{\mathbf{x}}^T f(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k),$$

or

$$f(\mathbf{x}) \approx \tilde{f}(\mathbf{x}) = f(\mathbf{x}_k) + \nabla_{\mathbf{x}}^T f(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^T \nabla_{\mathbf{x}}^2 f(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k),$$

where $\nabla_{\mathbf{x}} f$ and $\nabla_{\mathbf{x}}^2 f$ are the gradient and hessian respectively. Next a trust-region subproblem is solved:

$$\begin{aligned} & \min_{\mathbf{x}} \tilde{f}(\mathbf{x}) \\ & \text{subject to} \\ & \|\mathbf{x} - \mathbf{x}_k\|_2 \leq \Delta_k \\ & g(\mathbf{x}) = 0 \\ & h(\mathbf{x}) \leq 0, \end{aligned}$$

where Δ_k is referred to as the trust-region radius. Once the solution \mathbf{x} is found, a condition is tested to evaluate if the truncated Taylor series approximation made is valid to the true objective function f constrained to the trust-region radius Δ_k . In the case that the assumption is valid, the trust-region radius is increased by some rule in order to attempt to globalize the search for a local minimum since the approximation of the objective function and the objective function are approximately the same locally. If the assumption is not valid, then the trust-region radius is decreased by some rule. This indicates that the approximation is not accurate to the true objective function within the trust-region radius. This process continues until the trust-region radius is small, indicating that a local minimum has been found, thus terminating the trust-region method.

In [26, 27] a trust-region method was introduced for optimal control problems with binary variables. The main idea of this algorithm is use the standard first-order trust-region method suggested in [25] for solving nonlinear optimization problems. In this algorithm the gradient is understood to be the relaxed gradient, derived from the continuous relaxation of the binary variable as explained above.

Algorithm 1: Steepest-Descent Trust-Region Algorithm.

Given an initial trust-region radius $\Delta_0 \geq 1$, and an initial parameter vector $\boldsymbol{\alpha}^{(0)} \in \{0, 1\}^p$.

Select an acceptance step parameter $\hat{\rho}$, for example $\hat{\rho} = 0.75$, and initialize $k = 0$.

Evaluate the objective function $\mathcal{J}^{(k)} = \mathcal{J}(\boldsymbol{\alpha}^{(k)})$ and the gradient $g^{(k)} = \nabla_{\boldsymbol{\alpha}} \mathcal{J}(\boldsymbol{\alpha}^{(k)})$.

while $\Delta_k \geq 1$ **do**

 Solve the trust-region (knapsack) sub-problem for $\hat{\boldsymbol{\alpha}}$:

$$\begin{aligned} \hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & g^{(k)T} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^{(k)}) + \mathcal{J}^{(k)} \\ & \|\boldsymbol{\alpha} - \boldsymbol{\alpha}^{(k)}\|_1 \leq \Delta_k \\ & \boldsymbol{\alpha} \in \{0, 1\}^p \end{aligned}$$

 Solve the state equation for the binary parameter vector $\hat{\boldsymbol{\alpha}}$ and evaluate the objective $\mathcal{J}(\hat{\boldsymbol{\alpha}})$.
 Compute the ratio of actual over predicted reduction:

$$\rho_k = \frac{\mathcal{J}^{(k)} - \mathcal{J}(\hat{\boldsymbol{\alpha}})}{-(g^{(k)})^T (\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^{(k)})}$$

if $\|g^{(k)}\|_2 = 0$ *or* $\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}^{(k)}$ **then**

$\boldsymbol{\alpha}^{(k)}$ is a local solution

 Set $\boldsymbol{\alpha}^{(k+1)} = \boldsymbol{\alpha}^{(k)}$

 Reduce to trust-region radius $\Delta_{k+1} = 0$

else if $\rho_k > \hat{\rho}$ **then**

 Accept the step: $\boldsymbol{\alpha}^{(k+1)} = \hat{\boldsymbol{\alpha}}$, and evaluate the gradient $g^{(k+1)} = \nabla_{\hat{\boldsymbol{\alpha}}} \mathcal{J}(\boldsymbol{\alpha}^{(k+1)})$

if $\|\boldsymbol{\alpha}^{(k+1)} - \boldsymbol{\alpha}^{(k)}\|_1 = \Delta_k$, **then**

 Increase the trust-region radius $\Delta_{k+1} = 2\Delta_k$

else if $\rho_k > 0$ **then**

 Accept the step $\boldsymbol{\alpha}^{(k+1)} = \hat{\boldsymbol{\alpha}}$, and evaluate the gradient $g^{(k+1)} = \nabla_{\hat{\boldsymbol{\alpha}}} \mathcal{J}(\boldsymbol{\alpha}^{(k+1)})$

 Keep trust-region radius unchanged $\Delta_{k+1} = \Delta_k$

else

 Reject the step, set $\boldsymbol{\alpha}^{(k+1)} = \boldsymbol{\alpha}^{(k)}$, and copy the gradient $g^{(k+1)} = g^{(k)}$

 Reduce the trust-region radius $\Delta_{k+1} = \text{floor}(\frac{\Delta_k}{2})$

 Set $k \leftarrow k + 1$

For completeness, we present the first-order trust-region method in Algorithm 1. We note that because $\alpha \in \{0, 1\}^p$, we may rewrite the trust-region sub-problem (Hamming distance) constraint as:

$$\|\alpha - \alpha^{(k)}\|_1 = \sum_{\substack{j=1 \\ \alpha_j^{(k)}=0}}^p \alpha_j + \sum_{\substack{j=1 \\ \alpha_j^{(k)}=1}}^p (1 - \alpha_j) \leq \Delta_k,$$

which is linear in α . Hence, the trust-region sub-problem is a knapsack problem with computational complexity $\mathcal{O}(p \log(p))$ [28]. In the context of this work, the computational complexity for evaluating the gradient is $\mathcal{O}(p)$. This means that the first-order trust-region method applied to the SFQ problem has computational complexity $\mathcal{O}(p \log(p))$.

The convergence of the binary trust-region is inherited from the fact that the standard continuous trust-region method converges monotonically, implying that the objective function value must remain the same or decrease after each trust-region iteration [25]. This is to say that at step k of the standard trust-region method:

$$\mathcal{J}(\alpha_{k+1}) \leq \mathcal{J}(\alpha_k).$$

In our context, a binary (local) optimal solution $\tilde{\alpha}$ satisfies the condition:

$$\mathcal{J}(\tilde{\alpha}) \leq \mathcal{J}(\alpha),$$

for all α satisfying

$$\|\tilde{\alpha} - \alpha\|_1 \leq 1.$$

This is to say no local neighbor in binary space under the Hamming measure produces a lower objective function value. In this work we assume the existence of at least one local minimizing solution $\tilde{\alpha}$ (even though it may not be the only local optimal solutions), which the trust-region is theoretically guaranteed to converge to [25], and indeed find such minimum numerically. A proof for the existence of such minimizing solutions is out of the scope of this paper.

As we proceed to show in Section 5, our algorithm finds quality local solutions even if the binary optimization space is of high dimension. We note that we chose a first-order trust-region method versus a second order trust-region method because then the trust-region problem would become a quadratic integer programming problem, which could be computationally intractable.

5. Numerical experiments

To evaluate the performance of our trust region algorithm, we consider optimizing pulse sequences for two standard quantum gates: H and X. For each of these gates we study the cases when the pulse strength β corresponds to the tip angles $\theta = \frac{\pi}{100}$ and $\theta = \frac{\pi}{300}$.

To model a typical superconducting transmon, we set the fundamental frequency, self-Kerr coefficient, SFQ time step, and pulse duration according to

$$\begin{aligned} \frac{\omega}{2\pi} &= 5.0 \text{ GHz}, & \frac{\xi}{2\pi} &= 0.25 \text{ GHz} & \tau_p &= 2.5 \times 10^{-2} \text{ ns}, \\ \delta &= 4 \times 10^{-3} \text{ ns}. \end{aligned}$$

We numerically calculate the unitary matrices D_0 and D_1 by the Strömer-Verlet method, which is a partitioned Runge-Kutta scheme [19]. To effectively eliminate time-stepping errors, we solve the state equation on the interval $[0, \tau_p]$ with 10,000 time steps. Truncating the modal expansion of Schrödinger's equation after N terms is only valid if the occupation of the highest energy level is sufficiently small. From numerical experimentation we find that retaining four energy levels is sufficient for meeting that requirement; in the following, we use 2 essential states ($E = 2$) and 2 guard states ($G = 2$). The essential states correspond to the number states $|0\rangle$ and $|1\rangle$, and the guard states are $|2\rangle$ and $|3\rangle$. The truncation of

the modal expansion is justified when the occupation of the $|3\rangle$ state is small throughout the duration of the pulse sequence. Furthermore, for all experiments, we take $w_1 = 0.1$ and $w_2 = 1$ in the weight matrix W . In our experiments we observe that larger values of w_1 and w_2 reduce the leakage to higher energy states, but leads to a larger infidelity. We note that the values for w_1 and w_2 we use here are only tuned by numerical experiments. They may be further improved through a separate optimization procedure. We start by fixing the duration of each gate to $T = 40$ ns, which correspond to $p = 1,600$ SFQ time steps. In each experiment we apply the trust-region method for 10 random binary initial guesses, and report the trust-region solution that resulted in the smallest objective function. In these experiments we set the acceptance step size to be $\hat{\rho} = 0.75$, and the initial trust-region radius to equal the number of SFQ pulses ($\Delta_0 = p$). In a separate set of numerical experiments, reported in Section 5.4, we explore the shortest gate duration that is needed to realize the gates to high fidelity.

5.1. H gate experiment

For an H gate, the target unitary matrix is

$$V_E = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Starting from the $|0\rangle$ or $|1\rangle$ states, an H gate is realized when the probabilities of being in either the $|0\rangle$ or $|1\rangle$ states become equal.

In Figure 3 we plot the populations for the essential states, the leakage from the essential to the guard states, and the corresponding pulse sequence, for tip angles $\frac{\pi}{300}$ and $\frac{\pi}{100}$. We observe that we are successful in realizing the H gate for both tip angles when the duration is $T = 40$ ns. In the case of tip angle $\theta = \frac{\pi}{300}$ we observe an infidelity that is less than 10^{-4} , corresponding to a gate fidelity larger than 99.99 %. For the tip angle $\theta = \frac{\pi}{100}$ we observe an infidelity less than 10^{-3} . In that case, the gate fidelity exceeds 99.9%. Thus both tip angles meet the gate fidelity threshold of 99.9 %.

A desired trait of our approach is that the leakage from the essential states to the guard states is suppressed. The area under the delivered pulse, β , can be physically interpreted as the energy invested into the system. Because $\beta \sim \theta$, a scaling in θ corresponds to an equivalent scaling in β . So the energy delivered by a pulse with tip angle $\theta = \frac{\pi}{100}$ is three times stronger than the energy delivered by a pulse with tip angle $\theta = \frac{\pi}{300}$. We observe in the leakage plots in Figure 3 that more leakage occurs for the larger tip angle, because that corresponds to administering a pulse with more energy. This indicates that there is practical impact of using higher energy pulses compared to lower energy pulses. This also motivates the introduction of the leakage term \mathcal{J}_2 to suppress transition to the guard states. To justify the truncation of the modal expansion, it is desired to see sufficiently small population of the $|3\rangle$ state. In this case the largest population of $|3\rangle$ is less than 10^{-2} for $\theta = \frac{\pi}{300}$, and less than 10^{-1} for $\theta = \frac{\pi}{100}$. This observation indicates that there is advantage to using weaker pulses, because it leads to low infidelity and smaller population of the $|3\rangle$ state. Furthermore, the stronger pulse strength cause more leakage to occur, which increases the leakage term (\mathcal{J}_2) relative to the infidelity term (\mathcal{J}_1) in the objective function. As a result, the infidelity becomes an order of magnitude larger compared to a weaker pulse. This suggests that stronger pulses may not be favored. In Section 5.4 we discuss how stronger pulses, while generating more leakage, allow the gate duration to be reduced without increasing the gate infidelity.

In Figure 3 we present the pulse sequences for each of the two tip angles. We plot these in a barcode format, where the presence of a line indicates $\alpha_k = 1$ for a given k and otherwise $\alpha_k = 0$. We observe that there is a structure that emerges from solving the optimal control problem, with packets of pulses being created in what seems to be a semi-periodic structure.

In Figure 4 we plot, for $\theta = \frac{\pi}{300}$ and $\frac{\pi}{100}$, the trust-region convergence history of the objective functions \mathcal{J}_1 , \mathcal{J}_2 , and $\mathcal{J}_1 + C_1\mathcal{J}_2$. These plots illustrate that the number of iterations required to find a local minimizing binary solution is modest. For $\theta = \frac{\pi}{300}$, we find a minimal solution in around 25 iterations. In the case of $\theta = \frac{\pi}{100}$, around 20 iterations are needed. Since the number of control pulses in this case are $p = 1600$, there are $2^{1600} \approx 4.44 \times 10^{481}$ feasible solutions to the binary optimal control problem. In spite of the enormous size of the solution space, our method quickly finds quality solutions for both tip angles, with significant reduction in the infidelity term. We also observe that the leakage

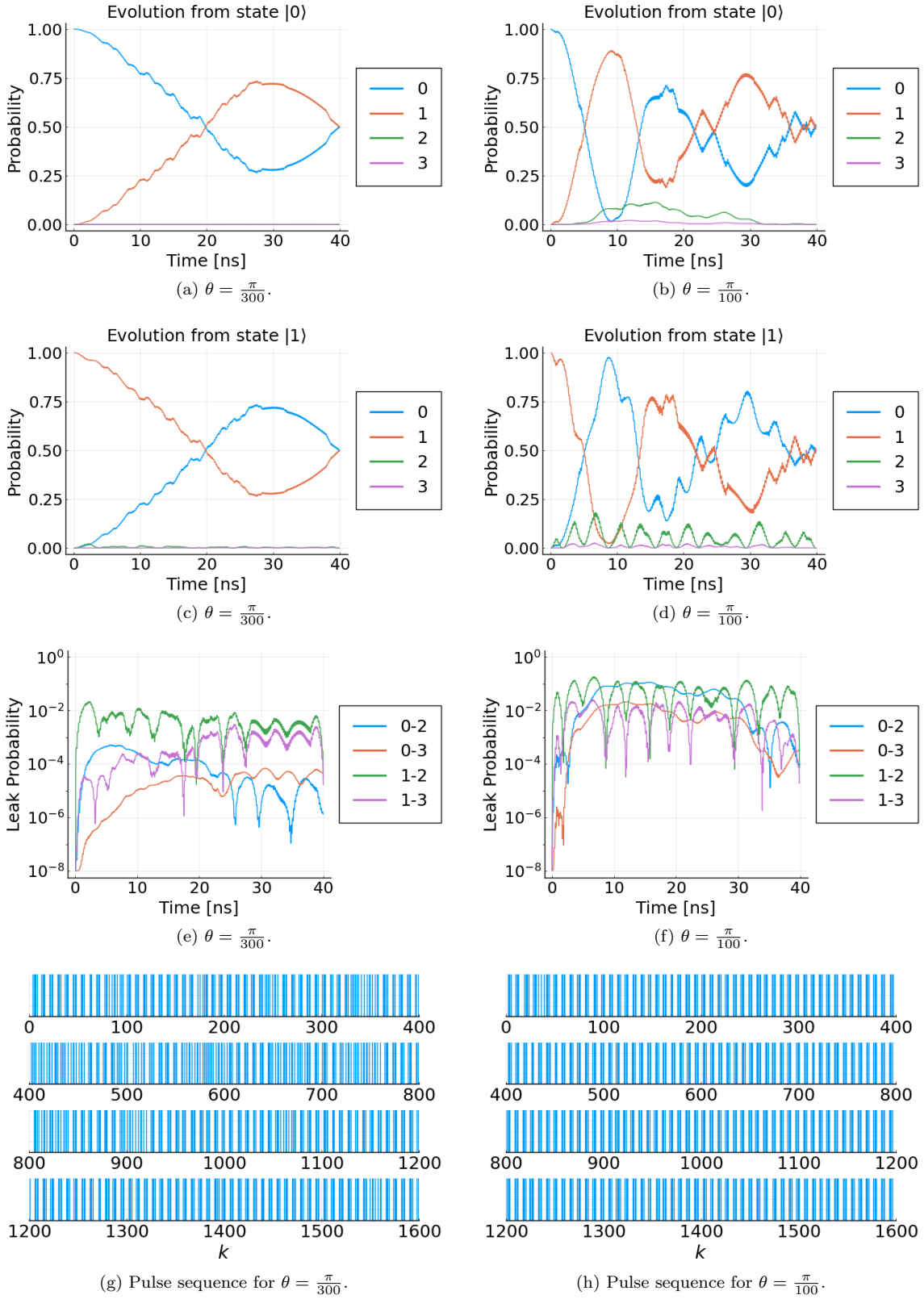


Figure 3: The evolution of essential and guard state populations during the H gate with tip angle $\theta = \frac{\pi}{300}, \frac{\pi}{100}$, and corresponding pulse sequences. In figures (e) and (f), the legend $a-b$ indicates the population of $|b\rangle$ corresponding to initial state $|a\rangle$.

is small in the initial guess and does not become larger in the optimized solution. Similar convergence histories are observed for the X gate as well.

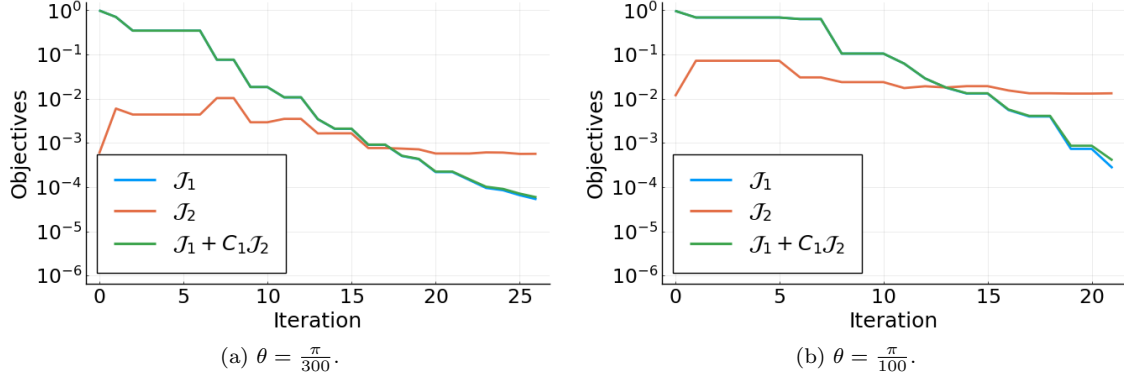


Figure 4: Convergence history for the trust-region method for realizing an H gate with 1600 control pulses, for tip angles $\theta = \frac{\pi}{300}$ (left) and $\frac{\pi}{100}$ (right).

5.2. X gate experiment

An X gate corresponds to the unitary transformation

$$V_E = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

To realize the X gate is equivalent to saying that the populations of the initial states are swapped, e.g. if the qubit starts in state $|0\rangle$, it will transition to state $|1\rangle$, and vice versa.

In Figure 5 we show the populations for the essential states, the leak probability from the essential to the guard states, and the pulse sequences associated with the tip angles $\theta = \frac{\pi}{300}$ and $\frac{\pi}{100}$. We observe close similarities between the H and X gates. First, we are able to realize the gates with infidelity that is less than 10^{-4} for tip angle $\theta = \frac{\pi}{300}$. In addition we observe an infidelity less than 10^{-3} for a tip angle of $\theta = \frac{\pi}{100}$. We observe more leakage for a stronger pulse than a weaker pulse. Similar to the H gate, there is a semi-periodic structure in the X gate pulse sequences.

5.3. Comparison with the branch and bound algorithm

The branch and bound (B&B) method is the primary algorithm leveraged for integer programming problems. To read more about the standard method we refer the reader to [29]. B&B only has theoretical guarantees of finding an optimal solution, even though it may take enormous effort, in the case of convex integer programming problems. In the case of nonconvex problems, like our SFQ optimal control problem, the method is merely a heuristic. To compare with our trust-region method, we applied the B&B approach to the H and X gate experiments. First, we executed the B&B algorithm using several different branching strategies: most infeasible branching, pseudo cost branching, and strong branching. For each branching strategy we allowed the B&B solution to search for a solution for the duration of one hour. In each run, B&B not only could not find an integer optimal solution, but the quality of the fidelity for both the H and X gate was only on the order of 10^{-1} . We conducted the experiment again, this time increasing the search time allotted from one hour to ten hours. The quality of the infidelity remained unchanged. These experiments indicate that the standard branch and bound approach is not successful at finding a quality solution; whereas our trust-region approaches finds solutions with infidelity less than 10^{-3} within one minute of computation. This isn't to say however that a specialized B&B scheme could not be developed for the SFQ application, however to develop such a scheme would require deep expertise in the theory of integer programming, and is still not guaranteed to solve the SFQ optimal control problem. While an interesting research direction, developing a specialized B&B for this application is out of the scope of the

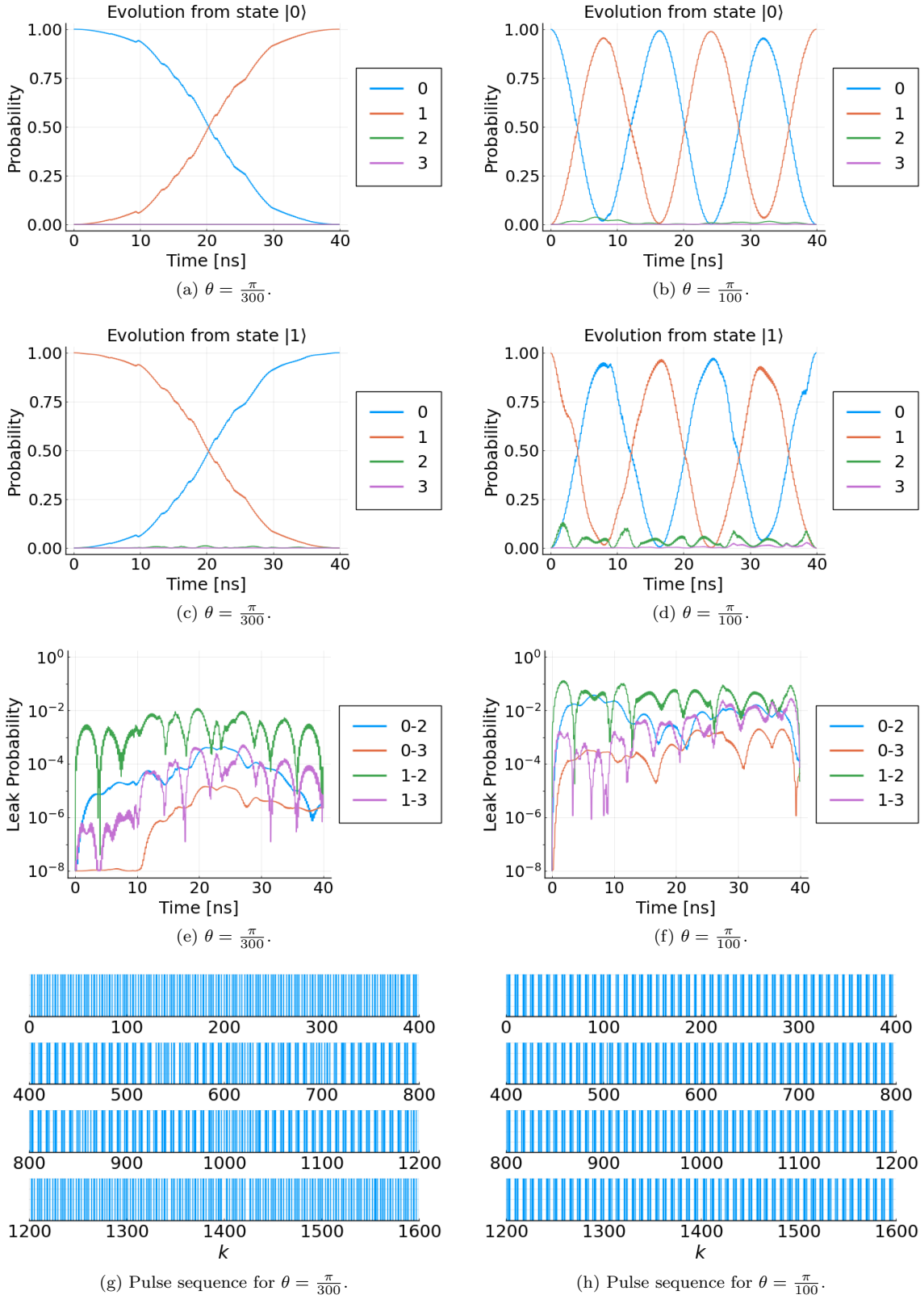


Figure 5: The evolution of essential and guard state populations during the X gate, with tip angles $\theta = \frac{\pi}{300}$ and $\frac{\pi}{100}$, and corresponding pulse sequences. In figures (e) and (f), the legend $a-b$ indicates the population of $|b\rangle$ corresponding to initial state $|a\rangle$.

paper. We refer the reader to convexification [30, 31, 32] and underestimators [33, 34, 35] used in order to develop specialized B&B schemes for nonconvex integer programming, to give insight how one may begin to think about developing a specialized B&B scheme for this application.

5.4. On minimizing the gate duration

We now examine the smallest gate duration that is required to achieve an acceptable gate infidelity for the H and X gates.

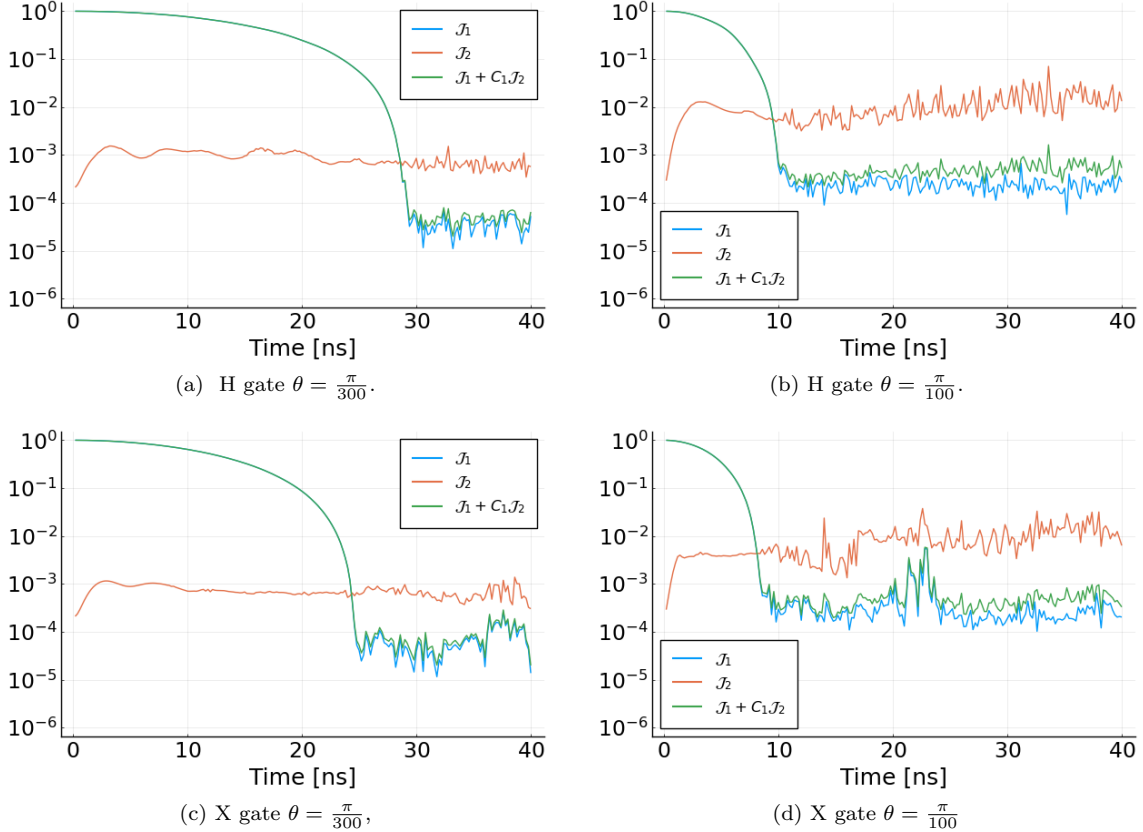


Figure 6: The infidelity, leakage, and total objective function on a \log_{10} scale for tip angle $\theta = \frac{\pi}{300}, \frac{\pi}{100}$ for the H and X gates.

In Figure 6 we study the behavior of the infidelity as a function of gate duration. We solve the optimal control problem for a gate duration of $[0, p\tau_p]$ for $p = 8, 16, 24, \dots, 1600$. For each value of p , we draw 10 random samples from $\{0, 1\}^p$ as initial guesses, and execute the trust-region on each sample. The trust-region solution with the lowest objective function value is shown in Figure 6. For all of the gates and tip angles considered here, we find solutions with an infidelity less than 10^{-3} for gate durations less than 40 ns. We observe that a pulse with three times the strength produces an infidelity that is less than 10^{-3} for a gate duration that is about three times shorter. In particular, for the H gate we realize a gate infidelity that is less than 10^{-3} for the tip angle $\theta = \frac{\pi}{300}$ in ≈ 28 ns, versus ≈ 10 ns for the tip angle $\theta = \frac{\pi}{100}$. For the X gate, we realize a gate infidelity that is less than 10^{-3} for the tip angle $\theta = \frac{\pi}{300}$ in ≈ 24 ns, versus ≈ 8 ns for tip angle $\theta = \frac{\pi}{100}$.

Our experiments indicate that high fidelity gates can be realized in shorter duration than 40 ns. The leakage to the $|3\rangle$ state largely depends on the amplitude of each pulse, i.e., less leakage is achieved by decreasing the tip angle. To find a solution that meets the needs for a small infidelity and a small leakage, we suggest an iterative approach where the number of pulses is gradually increased until the infidelity

becomes acceptable. If the leakage is too large, reduce the tip angle and repeat the calculation for a larger number of pulses.

6. Conclusions

In this paper we introduce a novel approach for optimizing pulse delivery to a SFQ quantum computer. We pose the problem as a binary optimal control problem, where the binary variables indicate whether each pulse is on or off, with a primary goal of minimizing the gate infidelity. A secondary goal of our formulation is to suppress leakage to the higher energy levels. To solve the optimal control problem we take advantage of a first order trust-region method. We devise an algorithm for calculating the objective function and its relaxed gradient in $\mathcal{O}(p)$ operations, where p is the number of SFQ time steps. Based on the linearity of the Hamming distance that appears in the trust-region sub-problem, the overall computational complexity of the algorithm becomes $\mathcal{O}(p \log p)$. Our numerical experiments indicate that our trust-region method needs ≈ 25 iterations to find local minima with gate fidelities better than an 99.9%. We also investigated the minimal gate duration that is required to realize a gate. The actual duration varies from gate to gate, but we have demonstrated that we are able to find pulse sequences for the H and X gate. Furthermore, our numerical experiments indicate pros and cons when it comes to using stronger versus weaker pulses. For a given gate, we observe that increasing the pulse amplitude by a factor of three allows the gate duration to be reduced by a factor of three. However, using stronger pulses results in more leakage, which is not desired. When using weaker pulses, we observe less leakage, but the gate duration must be increased to achieve an acceptable gate fidelity. We note that all behaviors we have observed for the H and X gates have also been observed in separate numerical experiments for the Y and Z gates. Those experiments were omitted to conserve space.

This paper illustrates that the reduced gradient trust region method is a very promising candidate for optimizing SFQ pulse sequences. Initial numerical experiments indicate that our approach will generalize in a straightforward way to SFQ control of two-qubit systems. While out of the scope of this work, in a future work we intend to introduce the final time duration T as an optimization variable; and penalize it such that we realize gates in the shortest duration possible for both single and many qubit systems.

Acknowledgments

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344; this is contribution LLNL-JRNL-823366.

Appendix A. Gradient computation

The gradient of U_j with respect to α_k satisfies

$$\frac{\partial U_j}{\partial \alpha_k} = \begin{cases} 0, & k > j, \\ B_k U_{k-1}, & k = j, \\ A_{k+1} B_k U_{k-1}, & k = j - 1, \\ A_j \cdots A_{k+1} B_k U_{k-1}, & k \leq j - 2, \\ A_j \cdots A_2 B_1, & k = 1. \end{cases} \quad (\text{A.1})$$

Recall that p corresponds to the total number of control pulses and that $S_T \in \mathbb{C}$. By differentiating (6) with respect to α_p and using (A.1), we get

$$\frac{d\mathcal{J}_1}{d\alpha_p} = -\frac{2}{E^2} \operatorname{Re} \left(\bar{S}_T \left\langle \frac{dU_p}{d\alpha_p} P, VP \right\rangle_F \right) = -\frac{2}{E^2} \operatorname{Re} \left(\bar{S}_T \langle B_p U_{p-1} P, VP \rangle_F \right).$$

For $k = p - 1$, we observe

$$\begin{aligned} \frac{d\mathcal{J}_1}{d\alpha_{p-1}} &= -\frac{2}{E^2} \operatorname{Re} \left(\bar{S}_T \left\langle \frac{dU_p}{d\alpha_{p-1}} P, VP \right\rangle_F \right) \\ &= -\frac{2}{E^2} \operatorname{Re} \left(\bar{S}_T \langle A_p B_{p-1} U_{p-2} P, VP \rangle_F \right) \\ &= -\frac{2}{E^2} \operatorname{Re} \left(\bar{S}_T \langle B_{p-1} U_{p-2} P, A_p^\dagger VP \rangle_F \right). \end{aligned}$$

For $k = p - 2$, we observe

$$\begin{aligned} \frac{d\mathcal{J}_1}{d\alpha_{p-2}} &= -\frac{2}{E^2} \operatorname{Re} \left(\bar{S}_T \left\langle \frac{dU_p}{d\alpha_{p-2}} P, VP \right\rangle_F \right) \\ &= -\frac{2}{E^2} \operatorname{Re} \left(\bar{S}_T \langle A_p A_{p-1} B_{p-2} U_{p-3} P, VP \rangle_F \right) \\ &= -\frac{2}{E^2} \operatorname{Re} \left(\bar{S}_T \langle B_{p-2} U_{p-3} P, A_{p-1}^\dagger A_p^\dagger VP \rangle_F \right). \end{aligned}$$

Let's define the discrete adjoint variable Λ_k according to

$$\begin{aligned} \Lambda_p &= V, \\ \Lambda_{p-1} &= A_p^\dagger V = A_p^\dagger \Lambda_p, \\ \Lambda_{p-2} &= A_{p-1}^\dagger A_p^\dagger V = A_{p-1}^\dagger \Lambda_{p-1}, \\ &\vdots \\ \Lambda_{p-q} &= A_{p-q+1}^\dagger \Lambda_{p-q+1} \end{aligned}$$

This means that all components of the gradient can be evaluated by backwards propagation of the adjoint variable,

$$\frac{d\mathcal{J}_1}{d\alpha_{p-q}} = -\frac{2}{E^2} \operatorname{Re} \left(\bar{S}_T \langle B_{p-q} U_{p-q-1} P, \Lambda_{p-q} P \rangle_F \right), \quad q = 0, 1, \dots, p-1.$$

Note that $U_0 = I$ does not depend on α . Because $W^\dagger = W$, the gradient of the leak term satisfies

$$\begin{aligned} \frac{d\mathcal{J}_2}{d\alpha_k} &= \frac{2}{p} \operatorname{Re} \left(\sum_{j=1}^{p-1} \left\langle \frac{dU_j}{d\alpha_k} P, WU_j P \right\rangle_F + \frac{1}{2} \left\langle \frac{dU_p}{d\alpha_k} P, WU_p P \right\rangle_F \right) \\ &= \frac{2}{p} \operatorname{Re} \left(\sum_{j=k}^{p-1} \left\langle \frac{dU_j}{d\alpha_k} P, WU_j P \right\rangle_F + \frac{1}{2} \left\langle \frac{dU_p}{d\alpha_k} P, WU_p P \right\rangle_F \right) \end{aligned}$$

For $k = p$,

$$\frac{d\mathcal{J}_2}{d\alpha_p} = \frac{2}{p} \operatorname{Re} \left\langle \frac{dU_p}{d\alpha_p} P, \frac{1}{2} WU_p P \right\rangle_F = \frac{2}{p} \operatorname{Re} \left\langle B_p U_{p-1} P, \frac{1}{2} WU_p P \right\rangle_F$$

For $k = p - 1$,

$$\begin{aligned}
\frac{d\mathcal{J}_2}{d\alpha_{p-1}} &= \frac{2}{p} \operatorname{Re} \left(\left\langle \frac{dU_{p-1}}{d\alpha_{p-1}} P, WU_{p-1} P \right\rangle_F + \left\langle \frac{dU_p}{d\alpha_{p-1}} P, \frac{1}{2} WU_p P \right\rangle_F \right) \\
&= \frac{2}{p} \operatorname{Re} \left(\langle B_{p-1} U_{p-2} P, WU_{p-1} P \rangle_F + \left\langle A_p B_{p-1} U_{p-2} P, \frac{1}{2} WU_p P \right\rangle_F \right) \\
&= \frac{2}{p} \operatorname{Re} \left(\left\langle B_{p-1} U_{p-2} P, WU_{p-1} P + \frac{1}{2} A_p^\dagger WU_p P \right\rangle_F \right)
\end{aligned}$$

For $k = p - 2$,

$$\begin{aligned}
\frac{d\mathcal{J}_2}{d\alpha_{p-2}} &= \frac{2}{p} \operatorname{Re} \left(\left\langle \frac{dU_{p-2}}{d\alpha_{p-2}} P, WU_{p-2} P \right\rangle_F + \left\langle \frac{dU_{p-1}}{d\alpha_{p-2}} P, WU_{p-1} P \right\rangle_F \right. \\
&\quad \left. + \left\langle \frac{dU_p}{d\alpha_{p-2}} P, \frac{1}{2} WU_p P \right\rangle_F \right) \\
&= \frac{2}{p} \operatorname{Re} \left(\langle B_{p-2} U_{p-3} P, WU_{p-2} P \rangle_F + \langle A_{p-1} B_{p-2} U_{p-3} P, WU_{p-1} P \rangle_F \right. \\
&\quad \left. + \left\langle A_D A_{p-1} B_{p-2} U_{p-3} P, \frac{1}{2} WU_p P \right\rangle_F \right) \\
&= \frac{2}{p} \operatorname{Re} \left\langle B_{p-2} U_{p-3} P, WU_{p-2} + A_{p-1}^\dagger WU_{p-1} + \frac{1}{2} A_{p-1}^\dagger A_p^\dagger WU_p \right\rangle_F
\end{aligned}$$

We now define

$$\begin{aligned}
\tilde{\Lambda}_p &= \frac{1}{2} WU_p P, \\
\tilde{\Lambda}_{p-1} &= WU_{p-1} P + \frac{1}{2} A_p^\dagger WU_p P = WU_{p-1} P + A_p^\dagger \tilde{\Lambda}_p, \\
\tilde{\Lambda}_{p-2} &= WU_{p-2} P + A_{p-1}^\dagger WU_{p-1} P + \frac{1}{2} A_{p-1}^\dagger A_p^\dagger WU_p P = WU_{p-2} P + A_{p-1}^\dagger \tilde{\Lambda}_{p-1}
\end{aligned}$$

One can verify the recursive formula,

$$\tilde{\Lambda}_{p-q} = \begin{cases} \frac{1}{2} WU_p P, & q = 0, \\ WU_{p-q} P + A_{p-q+1}^\dagger \tilde{\Lambda}_{p-q+1}, & q = 1, 2, \dots, p-1, \end{cases}$$

from which all components of the gradient of the objective function can be calculated. As a result, the contribution to the gradient from the leak term becomes

$$\frac{d\mathcal{J}_2}{d\alpha_{p-q}} = \frac{2}{p} \operatorname{Re} \left\langle B_{p-q} U_{p-q-1} P, \tilde{\Lambda}_{p-q} P \right\rangle_F, \quad q = 0, 1, \dots, p-1.$$

References

- [1] R. Cleve, A. Ekert, C. Macchiavello, M. Mosca, Quantum algorithms revisited, Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 454 (1969) (1998) 339–354.
- [2] A. Montanaro, Quantum algorithms: an overview, npj Quantum Information 2 (1) (2016) 1–8.
- [3] K.-H. Han, J.-H. Kim, Genetic quantum algorithm and its application to combinatorial optimization problem, in: Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512), Vol. 2, IEEE, 2000, pp. 1354–1360.
- [4] V. Mavroeidis, K. Vishi, M. D. Zych, A. Jøsang, The impact of quantum computing on present cryptography, arXiv preprint arXiv:1804.00200.
- [5] M. Nakahara, T. Ohmi, Quantum computing: from linear algebra to physical realizations, CRC press, 2008.
- [6] R. McDermott, M. Vavilov, B. Plourde, F. Wilhelm, P. Liebermann, O. Mukhanov, T. Ohki, Quantum–classical interface based on single flux quantum digital logic, Quantum science and technology 3 (2) (2018) 024004.

- [7] K. Li, R. McDermott, M. G. Vavilov, Hardware-efficient qubit control with single-flux-quantum pulse sequences, *Physical Review Applied* 12 (1) (2019) 014044.
- [8] K. K. Likharev, V. K. Semenov, Rsfq logic/memory family: A new josephson-junction technology for sub-terahertz-clock-frequency digital systems, *IEEE Transactions on Applied Superconductivity* 1 (1) (1991) 3–28.
- [9] S. J. Glaser, U. Boscain, T. Calarco, C. P. Koch, W. Köckenberger, R. Kosloff, I. Kuprov, B. Luy, S. Schirmer, T. Schulte-Herbrüggen, et al., Training schrödinger’s cat: quantum optimal control, *The European Physical Journal D* 69 (12) (2015) 1–24.
- [10] M. Y. Niu, S. Boixo, V. N. Smelyanskiy, H. Neven, Universal quantum control through deep reinforcement learning, *npj Quantum Information* 5 (1) (2019) 1–8.
- [11] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, S. J. Glaser, Optimal control of coupled spin dynamics: design of nmr pulse sequences by gradient ascent algorithms, *Journal of magnetic resonance* 172 (2) (2005) 296–305.
- [12] J. F. Gonçalves, M. G. Resende, Biased random-key genetic algorithms for combinatorial optimization, *Journal of Heuristics* 17 (5) (2011) 487–525.
- [13] H. Mühlenbein, M. Gorges-Schleuter, O. Krämer, Evolution algorithms in combinatorial optimization, *Parallel computing* 7 (1) (1988) 65–85.
- [14] A. Jaszkiwicz, Genetic local search for multi-objective combinatorial optimization, *European journal of operational research* 137 (1) (2002) 50–71.
- [15] R. Fletcher, S. Leyffer, Numerical experience with lower bounds for miqp branch-and-bound, *SIAM Journal on Optimization* 8 (2) (1998) 604–616.
- [16] S. Leyffer, Integrating sqp and branch-and-bound for mixed integer nonlinear programming, *Computational optimization and applications* 18 (3) (2001) 295–309.
- [17] S. Nahar, S. Sahni, E. Shragowitz, Simulated annealing and combinatorial optimization, in: *23rd ACM/IEEE Design Automation Conference, IEEE, 1986*, pp. 293–299.
- [18] W. J. Gordon, R. F. Riesenfeld, B-spline curves and surfaces, in: *Computer aided geometric design, Elsevier, 1974*, pp. 95–126.
- [19] N. A. Petersson, F. M. Garcia, A. E. Copeland, Y. L. Rydin, J. L. DuBois, Discrete adjoints for accurate numerical optimization with application to quantum control, *arXiv preprint arXiv:2001.01013*.
- [20] J. J. Sakurai, E. D. Commins, *Modern quantum mechanics, revised edition (1995)*.
- [21] M. W. Cooper, A survey of methods for pure nonlinear integer programming, *Management Science* 27 (3) (1981) 353–361.
- [22] R. Hemmecke, M. Köppe, J. Lee, R. Weismantel, Nonlinear integer programming, in: *50 Years of Integer Programming 1958-2008, Springer, 2010*, pp. 561–618.
- [23] J. Lee, S. Leyffer, *Mixed integer nonlinear programming, Vol. 154, Springer Science & Business Media, 2011*.
- [24] D. Li, X. Sun, et al., *Nonlinear integer programming, Vol. 84, Springer, 2006*.
- [25] J. Nocedal, S. J. Wright, *Numerical Optimization, Springer-Verlag, New York, 1999*.
- [26] R. H. Vogt, S. Leyffer, T. S. Munson, A mixed-integer pde-constrained optimization formulation for electromagnetic cloaking, *SIAM Journal on Scientific Computing* 44 (1) (2022) B29–B50.
- [27] R. H. Vogt, S. Strikwerda, Solving bang-bang problems using the immersed interface method and integer programming, *arXiv preprint arXiv:2104.08649*.
- [28] K. Dudziński, S. Walukiewicz, Exact methods for the knapsack problem and its generalizations, *European Journal of Operational Research* 28 (1) (1987) 3–21.
- [29] P. Bonami, J. Lee, S. Leyffer, A. Wächter, More branch-and-bound experiments in convex nonlinear integer programming, Preprint ANL/MCS-P1949-0911, Argonne National Laboratory, Mathematics and Computer Science Division.
- [30] D. P. Bertsekas, Convexification procedures and decomposition methods for nonconvex optimization problems, *Journal of Optimization Theory and Applications* 29 (2) (1979) 169–197.
- [31] Y. Mao, M. Szmuk, B. Açikmeşe, Successive convexification of non-convex optimal control problems and its convergence properties, in: *2016 IEEE 55th Conference on Decision and Control (CDC), IEEE, 2016*, pp. 3636–3641.
- [32] R. Pörn, I. Harjunkoski, T. Westerlund, Convexification of different classes of non-convex minlp problems, *Computers & chemical engineering* 23 (3) (1999) 439–448.
- [33] P. Kesavan, R. J. Allgor, E. P. Gatzke, P. I. Barton, Outer approximation algorithms for separable nonconvex mixed-integer nonlinear programs, *Mathematical Programming* 100 (3) (2004) 517–535.
- [34] G. P. McCormick, Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems, *Mathematical programming* 10 (1) (1976) 147–175.
- [35] H. S. Ryoo, N. V. Sahinidis, Global optimization of nonconvex nlp and minlp with applications in process design, *Computers & Chemical Engineering* 19 (5) (1995) 551–566.