Quantum Topology Optimization via Quantum Annealing

Zisheng Ye, Xiaoping Qian, Wenxiao Pan

Abstract—We present a quantum annealing-based solution method for topology optimization (TO). In particular, we consider TO in a more general setting, i.e., applied to structures of continuum domains where designs are represented as distributed functions, referred to as continuum TO problems. According to the problem's properties and structure, we formulate appropriate sub-problems that can be solved on an annealing-based quantum computer. The methodology established can effectively tackle continuum TO problems formulated as mixed-integer nonlinear programs. To maintain the resulting sub-problems small enough to be solved on quantum computers currently accessible with small numbers of quits and limited connectivity, we further develop a splitting approach that splits the problem into two parts: the first part can be efficiently solved on classical computers, and the second part with a reduced number of variables is solved on a quantum computer. By such, a practical continuum TO problem of varying scales can be handled on the D-Wave quantum annealer. More specifically, we concern the minimum compliance, a canonical TO problem that seeks an optimal distribution of materials to minimize the compliance with desired material usage. The superior performance of the developed methodology is assessed and compared with the state-of-the-art heuristic classical methods, in terms of both solution quality and computational efficiency. The present work hence provides a promising new avenue of applying quantum computing to practical designs of topology for various applications.

Index Terms—Mixed-integer nonlinear program; Quadratic unconstrained binary optimization; Quantum annealing; Topology optimization

I. Introduction

Topology optimization (TO) is a computational design method that aims to find optimal distribution of material to improve part performance under governing physical equilibrium equations [1]. It originates in structural mechanics [2] and has been applied to structures of continuum domains where designs are represented as distributed functions and to discrete truss structures where designs are of finite dimensions and represented via discrete parameters. In this paper, we focus on design problems of distributed parameter systems a.k.a. continuum problems. Continuum TO has received growing attention in a variety of fields, including multi-scale structures [3], fluid mechanics [4]–[7], electromagnetics [8], [9], photonics [10], [11], quantum devices [12], and coupled multiphysics problems [13]–[16]. Topologically optimized designs often exhibit complex free-form shapes, and additive manufacturing can produce parts of complex shapes. The recent wide adoption of additive manufacturing technologies have

led to further development of TO for a host of applications [17], including imposing additive manufacturability constraints during the optimization process [18], [19].

As TO has been expanded from structural mechanics to general multiphysics, it is accompanied by quickly growing complexity and vast searching space, resulting in grand computational challenges [12], [17], [20]. Quantum computing (QC) is emerging as a new computing paradigm that could be superior to classical computing for a variety of problems, optimization being among them. In literature, only a few attempts have been made to apply quantum algorithms for solving TO problems. Maruo et al. concerned the TO of electromagnetic devices [21]. For that, a linear optimization problem was formulated, which can be directly cast into a quadratic unconstrained binary optimization (QUBO) formulation through the Biot-Savart's law. The formulated QUBO problem was then solved through simulated annealing. Their results have indicated that the proposed QUBO formulation can significantly reduce the number of iterations required for the optimization process, compared with the commonly used classical method, normalized Gaussian network [22]. Thereby, the overall computational cost can be significantly reduced, because much fewer forward evaluations of the objective functions are invoked, which usually dominate the computational cost. However, the resultant QUBO formulation was not implemented in quantum annealers. Thus, the additional overhead due to quantum embedding and the limitations of quantum hardware currently accessible were not addressed in the analysis of computational cost. Also, the benefit of quantum annealing to converge to the optimal or ground state with larger probability than simulated annealing was not utilized. The work by Sato et al. reported the TO for a simple, discrete truss structure with three edges [23]. The objective functions of all possible configurations composed by three edges were simultaneously evaluated through quantum entanglement. The optimal configuration with the minimum objective function value was determined by applying the variational quantum algorithms (VQAs) that are suitable for the noisy intermediate-scale quantum (NISQ) devices. The VQAs and forward evaluations of the objective function via finite element analysis (FEA) were integrated together when implemented on a quantum computer, eliminating the need of amplitude estimation for the solution of FEA. In those two works, the proposed quantum algorithms can only be applied to linear TO problems, where the field variables such as magnetic flux density and temperature linearly depend on the design variables. Given that, the TO can be greatly simplified when the objective function is just a linear [23] or quadratic

Z. Ye, X. Qian and W. Pan are with the Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA.
    Corresponding author: Wenxiao Pan (email: wpan9@wisc.edu).

[22] function of the field variables. In addition, the volume or material usage constraint is not strictly imposed by any equality or inequality constraint; as a result, one cannot set a target optimal volume or material usage for the optimization, which is not common in practical TO applications. Further, discrete truss structures only represent a small portion of applications of TO. More generally, TO considers structures of continuum domains where designs are represented as distributed functions.

In the present work, we concern how to apply quantum algorithms to *continuum*, *nonlinear* TO, beyond the discrete truss structures or the linear TO problems tackled in literature [22], [23]. In addition, the volume constraint is strictly imposed by an equality or inequality constraint, such that any target optimal volume can be practically achieved. To determine the optimal distribution of materials, continuum TO essentially seeks the numerical solution of a partial differential equation (PDE)-constrained optimization problem. To solve it, the solution domain is first discretized with discrete nodes or elements. The distributed design problem hence involves a large number of binary (0/1) optimization variables, typically one variable per discrete node or element. As the field variables nonlinearly depend on the design variables, it results in a large-scale, mixed integer nonlinear programming (MINLP) problem, subject to equality and/or inequality constraints. This type of problem is intrinsically NP-hard [24], namely unable to be solved in polynomial time, and hence is challenging to be handled using classical approaches [25]–[28].

There are two main implementations of quantum computers: quantum annealers and NISQ processors. Quantum annealers embrace quantum annealing techniques, and the NISQ processors are build with gate-based quantum circuits. Due to the noisy nature of quantum gates and the high overhead cost of noise reduction and error mitigation, the quantum circuit that can be reliably executed on NISQ devices must be limited to short. Thus, hybrid quantum-classical algorithms such as VAQs must be developed for solving optimization on NISQ devices, where only a short parameterized circuit is executed, and classical optimizers are leveraged to train the quantum circuit by optimizing the expectation value over the circuit's parameters [29]. However, in light of the noise-induced barren plateau [30]–[32], the training process calls for an exponentially scaled computational cost [30], and the classical optimization problems corresponding to the training of parameters are NP-hard [33]. Therefore, in the present work, we exploit quantum annealing for solving the TO problem of our interest. Quantum annealing has been demonstrated for its advantages and speedups in solving a variety of combinatorial optimization problems [34]–[38]. For a n-qubit system, it works in the binary discrete space, with the operator defined as a Hamiltonian that can correspond to the objective function of an optimization problem. Quantum annealing can effectively escape local minima by tunneling through the barriers when exploring the objective function's landscape and seeking the global minimum. Thus, it can efficiently solve large-scale nonlinear optimization problems over binary discrete spaces. The final state of the n-qubit system at the end of annealing could be the ground state of the Hamiltonian, with each

qubit a classical object standing for the solution to the binary optimization problem.

The currently accessible quantum computers that are designed to implement quantum annealing are the D-Wave systems [39]. To embed problems on the D-Wave quantum annealer, they need to be formulated in the form of QUBO. However, the optimization problem corresponding to the TO of our interest cannot be directly cast into QUBO formulations. Therefore, we first decompose the original problem into a sequence of mixed integer linear programs (MILPs) via the generalized Benders' decomposition (GBD). Next, in each MILP the continuous variables are represented by a series of auxiliary binary variables, such that each MILP can be mapped onto a binary optimization problem and then through the penalty method transformed into the QUBO formulation. Finally, from the solution of quantum annealing, we obtain the solution of each MILP, and in turn the ultimate optimal material layout after the sequence of MILPs are completely solved. If sufficient, fully connected quantum qubits are available [40], quantum advantages in computational efficiency can be straightforwardly obtained and demonstrated, owing to the efficiency of quantum annealing for solving QUBO. However, currently accessible quantum hardware only offers a small number of qubits with sparse connectivity. To solve practical TO problems in current and near-term quantum annealing devices, we further develop a splitting approach, by which a reduced problem that needs to be solved on a quantum annealer is separated from other sub-problems that can be efficiently handled by classical computers. The resultant, greatly reduced QUBO problem is then implemented on a D-Wave Advantage quantum processing unit (QPU). Both the solution quality and computational cost (including the cost for embedding the problem onto the quantum computer, the time of quantum annealing, and the number of optimization iterations) are analyzed and assessed systematically in a series of problems of variable sizes.

II. PRELIMINARY

A. A Continuum Topology Optimization Problem

Continuum TO optimizes the material layout $\rho(\mathbf{x})$ within a continuum design domain Ω. A continuum TO problem is usually constrained by a set of partial differential equations (PDEs) subject to boundary conditions (BCs) which describe the physical laws governing the design, as well as the target material usage or volume of material layout. In general, the problem can be stated as:

$$\min_{\rho} \quad \int_{\Omega} F[\mathbf{u}(\rho(\mathbf{x}))]\mathrm{d}\Omega$$

$$\text{s.t.} \quad G_0(\rho(\mathbf{x})) = \frac{\int_{\Omega} \rho(\mathbf{x})\mathrm{d}\Omega}{\int_{\Omega}\mathrm{d}\Omega} - V_T = 0$$

$$\mathcal{L}(\mathbf{u}(\rho(\mathbf{x}))) + \mathbf{b} = 0 \qquad \forall \mathbf{x} \in \Omega$$

$$\mathbf{u}(\rho(\mathbf{x})) = \mathbf{u}_{\Gamma}(\mathbf{x}) \qquad \forall \mathbf{x} \in \Gamma_D$$

$$\mathbf{n} \cdot \nabla \mathbf{u}(\rho(\mathbf{x})) = \mathbf{h}_{\Gamma}(\mathbf{x}) \qquad \forall \mathbf{x} \in \Gamma_N$$

$$G_j(\mathbf{u}(\rho(\mathbf{x})), \rho(\mathbf{x})) = 0, \quad j = 1, \ldots m$$

$$H_k(\mathbf{u}(\rho(\mathbf{x})), \rho(\mathbf{x})) \leq 0, \quad k = 1, \ldots, n$$

$$\tag{1}$$

where $\mathbf{u}$ denotes the field variable defined in the design domain Ω; ρ is the design variable; $\mathcal{L}$ denotes some differential operator; $\mathbf{b}$ is the source term; and $\Gamma = \partial\Omega$ represents the boundary of the design domain with Γ_D denoting the boundary where Dirichlet BC is imposed and Γ_N the boundary where Neumann BC is enforced. In Eq. (1), $G_0(\rho) = 0$ serves to constrain the volume of optimal material layout to the target value V_T. The following three lines after $G_0(\rho) = 0$ outline the governing PDE and BCs. $G_j(\mathbf{u}(\rho), \rho) = 0$ includes all equality constraints in addition to the volume constraint; and $H_k(\mathbf{u}(\rho), \rho) \leq 0$ comprises all inequality constraints imposed to the optimization problem. In practice, these constraints are usually related to certain manufacture limitations and/or material property requirements. For simplicity, we neglect them in the following discussion. However, they can be easily included in the proposed approach, following the way how we deal with the constraint $G_0(\rho) = 0$. The objective function F is convex with respect to the design variable ρ, and hence, if the differential operator $\mathcal{L}$ is positive definite, the optimization problem defined in Eq. (1) is convex. In case that the TO of interest leads to a non-convex optimization problem, a sequential approximation program can always be employed such that a series of local, convex problems are solved to update the design variable locally in a sequence [41].

To solve the continuum TO problem as in Eq. (1), the design domain Ω is usually discretized, and the design and field variables are represented in discrete settings. For example, Ω can be discretized with a uniform mesh, and FEA is used for the numerical discretization of the governing PDEs and BCs. Taking the minimum compliance as an example, the discretized TO problem can be expressed as:

$$\min_{\mathbf{u}, \boldsymbol{\rho}} \quad \mathbf{f}^{\mathsf{T}}\mathbf{u}$$
$$\text{s.t.} \quad \mathbf{K}(\boldsymbol{\rho})\mathbf{u} = \mathbf{f}$$
$$\sum_{i=1}^{n_\rho} \frac{\rho_i}{n_\rho} = V_T \quad (2)$$
$$\mathbf{u} \in \mathbb{R}^{n_u}, \boldsymbol{\rho} \in \{0, 1\}^{n_\rho}$$

where $\mathbf{f}$ is a known external load exerted on the material; the material is subject to linear elasticity; the superscript T denotes transpose; $\mathbf{u}$ is the discretized displacement field defined on the nodes of the mesh; and ρ_i is the discretized design variable defined on each mesh element i. In Eq. (2),

$$\mathbf{K}(\boldsymbol{\rho}) = \mathbf{K}_0 + \sum_{i=1}^{n_\rho} \rho_i \mathbf{K}_i(E, \nu) , \quad (3)$$

where $\mathbf{K}_i \in \mathbb{R}^{n_u \times n_u}, i = 1, 2, \ldots n_\rho$ is the predefined element stiffness matrix; E is the Young's modulus; ν is the Poisson ratio; and $\mathbf{K}_0$ is a symmetric positive definite matrix:

$$\mathbf{K}_0 = \varepsilon \sum_{i=1}^{n_\rho} \mathbf{K}_i , \quad (4)$$

with ε a small number, e.g. $\varepsilon = 10^{-9}$, such that it would not affect the resulting optimal material layout and the values estimated for the field variable on the solid elements with $\rho_i = 1$. The equality constraint $\sum_{i=1}^{n_\rho} \frac{\rho_i}{n_\rho} = V_T$ in Eq. (2)

is referred to as the volume constraint that drives the resulting material layout toward the target volume or material usage. The inclusion of $\mathbf{K}_0$ in $\mathbf{K}$ ensures that for any $\boldsymbol{\rho}$, $\mathbf{K}(\boldsymbol{\rho})$ is a symmetric positive definite matrix.

The challenge for solving the optimization problem defined in Eq. (2) lies in the fact that while $\mathbf{u} \in \mathbb{R}^{n_u}$ is a continuous variable, $\boldsymbol{\rho} \in \{0, 1\}^{n_\rho}$ is a binary variable, and the filed variable $\mathbf{u}$ nonlinearly depends on the design variable $\boldsymbol{\rho}$, resulting in a MINLP problem. To tackle the challenge, existing classical approaches in TO fall into either of the two categories.

In the first category, the most widely used method is the Solid Isotropic Material with Penalization (SIMP) method [25]. Its strategy is to relax the binary variable into a continuous variable such that the MINLP problem can be converted into a continuous optimization problem. Specifically, the SIMP method approximates the binary design variable with a high-order polynomial function of a continuous variable that smooths out the discontinuity of the binary variable [25]. Hence, the density ρ_i continuously varies from 0 to 1 for each element. From the design perspective, such an interpolated representation (with "gray" density) does not allow easy imposition of design-dependent loading. To recover the desired binary (black/white) representation of material layout in the optimal design, additional numerical treatment is required [42], which is not always mathematically justifiable. Also, the relaxation from binary to continuous breaks the convexity of the original problem (2) and makes the optimization hard to converge [42]. Further, the resultant continuous optimization problem is not suitable for quantum annealing to solve. Therefore, we do not follow the SIMP method herein, but use its solution as the baseline for comparison with the proposed quantum annealing solution, as discussed in Section IV-C.

The second category of methods separate the binary and continuous variables into different sub-problems, by iteratively determining one of them with the other fixed in a sub-problem. The representative methods include the Discrete Variable Topology Optimization via Canonical Relaxation Algorithm (DVTOCRA) [26], the Topology Optimization of Binary Structures (TOBS) method [27], and the GBD method [28]. The DVTOCRA method employs a sequential linear/quadratic approximation to separate the binary and continuous variables into different sub-problems [26]. The sub-problem associated with the binary variable is a constrained, quadratic integer programming problem. Due to its NP-hardness, the canonical relaxation algorithm is applied, by which new continuous variables are introduced to replace the binary variable through an approximation function (different from that used in the SIMP method), and the binary optimization programming problem is transformed into a convex, continuous optimization problem in terms of the newly introduced continuous variables. Similarly to the SIMP method, the resultant continuous optimization problems are not suitable for quantum annealing to solve. The TOBS method utilizes sequential linear programming to separate the binary and continuous variables into different sub-problems [27]. The resultant sub-problem involving the binary variable is a linear integer programming problem. With the uniform mesh discretization and the exclusion of non-

volumetric constraints (i.e., $G_j = 0$ and $H_k \leq 0$ in Eq. (1)), the corresponding integer programming problem is not NP-hard. Hence, the TOBS method does not serve as an ideal candidate for investigating the potential advantages of quantum annealing in TO.

The GBD method follows a different route. It decomposes the original problem into a sequence of MILPs. In contrast to the other methods that have no guarantee of convergence, the GBD method provides a deterministic optimality criterion to warrant convergence within a finite number of iterations [24]. It is also anticipated to converge faster than the SIMP or TOBS method, because the GBD formulation permits to use all the material layouts generated from previous iterations in each new iteration, while other methods like the SIMP or TOBS can only take into account the material layout yield from the last iteration. The GBD method was first introduced to TO by Muñoz et al. [28] for solving discrete TO problems. In the present paper, we extend it to address a continuum TO problem, as discussed in Section II-B. In the sub-problem related to the binary design variable after decomposition, additional constraints (associated with the PDEs), other than the volumetric constraint, are introduced and can bring NP-hardness to the optimization problem. Such a sub-problem is well suited to be accelerated by quantum annealing, and hence we focus on the GBD method in the present paper to investigate and demonstrate how QC, particularly via quantum annealing, can be leveraged for solving a continuum, nonlinear TO problem as stated in Eq. (2).

B. Generalized Benders' Decomposition

The field variable $\mathbf{u}$ and the design variable $\boldsymbol{\rho}$ are separated into different sub-problems via decomposition and are iteratively updated until reaching the convergence.

At the k-th iteration, we have a sequence of feasible integer solutions $\boldsymbol{\rho}^j$ that satisfy the volume constraint with *any* desired volume V, i.e., $\sum_{i=1}^{n_\rho} \rho_i^j = V$ for $\forall j$, and have non-singular stiffness matrices $\mathbf{K}(\boldsymbol{\rho}^j)$, where j loops over all previous iterations until the current iteration k. We first form the sub-problem with respect to $\mathbf{u}$, i.e., the so-called primal problem, which is given by:

$$\min_{\mathbf{u}} \quad \mathbf{f}^\mathsf{T}\mathbf{u}$$
$$\text{s.t.} \quad \mathbf{K}(\boldsymbol{\rho}^k)\mathbf{u} = \mathbf{f} \tag{5}$$
$$\mathbf{u} \in \mathbb{R}^{n_u}$$

This primal problem can be easily solved for $\mathbf{u}^k$ by using a linear system solver, as:

$$\mathbf{u}^k = \mathbf{K}^{-1}(\boldsymbol{\rho}^k)\mathbf{f} . \tag{6}$$

The primal problem is a restriction to the original problem (2), and any $\mathbf{f}^\mathsf{T}\mathbf{u}^j$ serves as an upper bound to the problem (2). We denote the lowest upper bound as U such that $U = \min_j(\mathbf{f}^\mathsf{T}\mathbf{u}^j), j = 1, \ldots, k$.

Next, we derive the sub-problem with respect to $\boldsymbol{\rho}$, referred to as the master problem. To proceed, we first use a Lagrange multiplier $\boldsymbol{\lambda}$ to move the equality constraint to the objective function in problem (2), such that the resulting problem is only subject to the constraints involving the binary variables, as:

$$\min_{\mathbf{u},\boldsymbol{\rho}} \quad \mathbf{f}^\mathsf{T}\mathbf{u} + \boldsymbol{\lambda}^\mathsf{T}\mathbf{K}(\boldsymbol{\rho})\mathbf{u}$$
$$\text{s.t.} \quad \sum_{i=1}^{n_\rho} \frac{\rho_i}{n_\rho} = V \tag{7}$$
$$\mathbf{u} \in \mathbb{R}^{n_u}, \boldsymbol{\rho} \in \{0,1\}^{n_\rho}$$

The Lagrange multiplier $\boldsymbol{\lambda}$ satisfies:

$$\boldsymbol{\lambda}^\mathsf{T}\mathbf{K}(\boldsymbol{\rho}) = -\mathbf{f}^\mathsf{T} . \tag{8}$$

Then, we introduce an auxiliary variable η and replace the objective function in Eq. (7) with an inequality constraint, as:

$$\min_{\boldsymbol{\rho},\mathbf{u},\eta} \quad \eta$$
$$\text{s.t.} \quad \mathbf{f}^\mathsf{T}\mathbf{u} + \boldsymbol{\lambda}^\mathsf{T}\mathbf{K}(\boldsymbol{\rho})\mathbf{u} \leq \eta$$
$$\sum_{i=1}^{n_\rho} \frac{\rho_i}{n_\rho} = V \tag{9}$$
$$\mathbf{u} \in \mathbb{R}^{n_u}, \boldsymbol{\rho} \in \{0,1\}^{n_\rho}$$

Applying the first-order Taylor expansion at $(\mathbf{u}^k, \boldsymbol{\rho}^k)$ and due to the convexity of $\mathbf{f}^\mathsf{T}\mathbf{u} + \boldsymbol{\lambda}^\mathsf{T}\mathbf{K}(\boldsymbol{\rho})\mathbf{u}$, the inequality constraint in Eq. (9) can be relaxed as:

$$\mathbf{f}^\mathsf{T}\mathbf{u}^k + \sum_{i=1}^{n_\rho} (\boldsymbol{\lambda}^k)^\mathsf{T}\mathbf{K}_i\mathbf{u}^k(\rho_i - \rho_i^k) \leq \mathbf{f}^\mathsf{T}\mathbf{u} \leq \eta \tag{10}$$

By doing so, the problem (9) can be relaxed with multiple cuts as:

$$\min_{\boldsymbol{\rho},\eta} \quad \eta$$
$$\text{s.t.} \quad \mathbf{f}^\mathsf{T}\mathbf{u}^j + \sum_{i=1}^{n_\rho} (\boldsymbol{\lambda}^j)^\mathsf{T}\mathbf{K}_i\mathbf{u}^j(\rho_i - \rho_i^j) \leq \eta$$
$$j = 1, \ldots, k \tag{11}$$
$$\sum_{i=1}^{n_\rho} \frac{\rho_i}{n_\rho} = V$$
$$\boldsymbol{\rho} \in \{0,1\}^{n_\rho}$$

Eq. (11) is the derived master problem for the original TO problem (2). The term $(\boldsymbol{\lambda}^j)^\mathsf{T}\mathbf{K}_i\mathbf{u}^j$ denotes the so-called sensitivity in TO. Note that $\boldsymbol{\lambda}^j = -\mathbf{u}^j$, due to the symmetry of $\mathbf{K}$, and hence the sensitivity is just $-(\mathbf{u}^j)^\mathsf{T}\mathbf{K}_i\mathbf{u}^j$. The optimal solution of Eq. (11) is denoted as $\boldsymbol{\rho}^{k+1}$ and added into the sequence of $\boldsymbol{\rho}^j$. The optimum of Eq. (11), denoted as η^k, serves as the lower bound of the problem (2). The iterations continue until the upper and lower bounds meet such that $(U - \eta^k)/U < \xi$, where ξ is a predefined tolerance for convergence.

Since the continuum TO is essentially a PDE-constrained optimization problem (as in (1)), the solution quality also depends on the accuracy of the numerical approximation of the differential operator (e.g., gradient) in the PDE, which becomes especially challenging at the interface between solid ($\rho_i = 1$) and void ($\rho_i = 0$) elements. The FEA with uniform meshing can lead to the so-called "checkerboard" artifact, as illustrated in Fig. 1. To remedy for that, filtering is required

(a) Material layout with "checkerboard" artifact

(b) Material layout after filtering
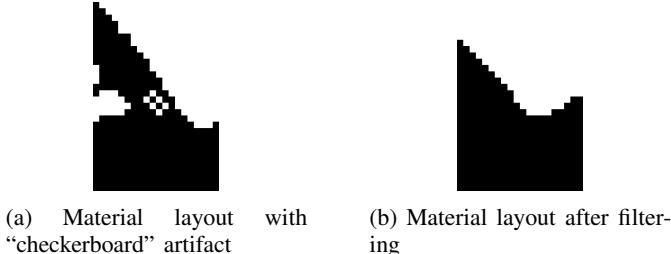
Fig. 1: The effect of filtering on eliminating the checkerboard artifact. The two sub-figures report the resulting optimal material layouts with the same problem setup, using the same optimization procedure, and corresponding to the same region of the solution domain. (a) The optimal material layout yield without filtering exhibits a checkerboard pattern. (b) After applying filtering, the checkerboard artifact is effectively eliminated.

for the sensitivity, following the literature [27], [43], as:

$$
\widetilde{w}_i^j = \begin{cases} \dfrac{\sum_{l \in \mathcal{N}_i^r} h_{i,l}^r \mathbf{u}^j \mathbf{K}_l \mathbf{u}^j}{\sum_{l \in \mathcal{N}_i^r} h_{i,l}^r}, & \rho_i^j = 1 \\[2ex] \varepsilon \dfrac{\sum_{l \in \mathcal{N}_i^r} h_{i,l}^r \mathbf{u}^j \mathbf{K}_l \mathbf{u}^j}{\sum_{l \in \mathcal{N}_i^r} h_{i,l}^r}, & \rho_i^j = 0 \end{cases} \tag{12}
$$

where $h_{i,l}^r = \max(0, r - \|\mathbf{x}_i - \mathbf{x}_l\|_2)$. Therefore, the master problem with filtering is rewritten as:

$$
\begin{aligned}
\min_{\boldsymbol{\rho}, \eta} \quad & \eta \\
\text{s.t.} \quad & \mathbf{f}^{\mathsf{T}} \mathbf{u}^j - \sum_{i=1}^{n_\rho} \widetilde{w}_i^j (\rho_i - \rho_i^j) \leq \eta \\
& \qquad\qquad\qquad\qquad j = 1, \dots, k \\
& \sum_{i=1}^{n_\rho} \frac{\rho_i}{n_\rho} = V \\
& \boldsymbol{\rho} \in \{0,1\}^{n_\rho}
\end{aligned} \tag{13}
$$

As the iterations proceed, the number of inequality constraints ($j = 1, \dots, k$) included in the master problem (13) increases. However, those inequality constraints are not independent, and we do not need to include them all when solving the master problem. To accelerate the solution process, we further introduce the Pareto optimal cuts [28], as defined by:

$$
\mathcal{P}(k) = \{j | \mathbf{f}^{\mathsf{T}} \mathbf{u}^j \leq \mathbf{f}^{\mathsf{T}} \mathbf{u}^k, \forall j = 1, \dots k\}, \tag{14}
$$

which is based on the Pareto dominance relationship and selects only the previous cuts with the objective function values no greater than that of the new cut generated in the current iteration. Such selection of optimal cuts has been numerically proven to improve the rate of convergence to the optimal solution [28]. Thus, the master problem further with

the Pareto optimal cuts can be written as:

$$
\begin{aligned}
\min_{\boldsymbol{\rho}, \eta} \quad & \eta \\
\text{s.t.} \quad & \mathbf{f}^{\mathsf{T}} \mathbf{u}^j - \sum_{i=1}^{n_\rho} \widetilde{w}_i^j (\rho_i - \rho_i^j) \leq \eta, \quad \forall j \in \mathcal{P}(k) \\
& \sum_{i=1}^{n_\rho} \frac{\rho_i}{n_\rho} = V \\
& \boldsymbol{\rho} \in \{0,1\}^{n_\rho}
\end{aligned} \tag{15}
$$

If $|P(k)| = 1$, the master problem in Eq. (15) can be simplified as:

$$
\begin{aligned}
\min_{\boldsymbol{\rho}} \quad & \mathbf{f}^{\mathsf{T}} \mathbf{u}^j - \sum_{i=1}^{n_\rho} \widetilde{w}_i^j (\rho_i - \rho_i^j), \quad j \in \mathcal{P}(k) \\
\text{s.t.} \quad & \sum_{i=1}^{n_\rho} \frac{\rho_i}{n_\rho} = V \\
& \boldsymbol{\rho} \in \{0,1\}^{n_\rho}
\end{aligned} \tag{16}
$$

with $\eta^k = \mathbf{f}^{\mathsf{T}} \mathbf{u}^j - \sum_{i=1}^{n_\rho} \widetilde{w}_i^j (\rho_i^{k+1} - \rho_i^j)$.

In summary, for any given volume constraint V and the initial feasible integer solution $\boldsymbol{\rho}^1$, the iterative procedure based on the GBD method to solve the TO problem (2) is outlined in Algorithm 1.

Algorithm 1 ToGBDSub($V, \boldsymbol{\rho}^1$)

Input: Volume fraction V, initial material layout $\boldsymbol{\rho}^1$
Output: Optimal material layout $\boldsymbol{\rho}^*$
Utility: Find an optimal material layout with a given volume fraction V and initial material layout $\boldsymbol{\rho}^1$

1: **for** $k = 1, \dots$ **do**
2:      Initialize the upper bound U as $U = +\infty$
3:      Employ the linear system solver to obtain $\mathbf{u}^k = \mathbf{K}^{-1}(\boldsymbol{\rho}^k)\mathbf{f}$
4:      **if** $\mathbf{f}^{\mathsf{T}} \mathbf{u}^k < U$ **then**
5:          $U \leftarrow \mathbf{f}^{\mathsf{T}} \mathbf{u}^k$
6:          $\boldsymbol{\rho}^* \leftarrow \boldsymbol{\rho}^k$
7:      **end if**
8:      Generate the Pareto optimal cuts $\mathcal{P}(k)$ according to (14)
9:      **if** $|\mathcal{P}(k)| = 1$ **then**
10:         Solve the master problem (16) for the minimizer η^k and $\boldsymbol{\rho}^{k+1}$
11:      **else**
12:         Solve the master problem (15) for the minimizer η^k and $\boldsymbol{\rho}^{k+1}$
13:      **end if**
14:      **if** $(U - \eta^k)/U < \xi$ **then**
15:         **break**
16:      **end if**
17: **end for**
     Return: $\boldsymbol{\rho}^*$

In the following, we explain how to obtain $\boldsymbol{\rho}^1$ to initiate the above iterative procedure, which is a feasible binary solution satisfying the volume constraint $\sum_{i=1}^{n_\rho} \frac{\rho_i}{n_\rho} = V$ and has a non-

singular stiffness matrix $\mathbf{K}(\boldsymbol{\rho}^1)$. First, we consider solving the problem:

$$\min_{\boldsymbol{\rho}} \quad \mathbf{f}^\mathsf{T}\mathbf{u}^0 - \sum_{i=1}^{n_\rho} \widetilde{w}_i^0 (\rho_i - \rho_i^0)$$
$$\text{s.t.} \quad \sum_{i=1}^{n_\rho} \frac{\rho_i}{n_\rho} = V \qquad (17)$$
$$\boldsymbol{\rho} \in \{0,1\}^{n_\rho}$$

where $\mathbf{u}^0 = \mathbf{K}^{-1}(\boldsymbol{\rho}^0)\mathbf{f}$; $\boldsymbol{\rho}^0$ can be any material layout that gives a non-singular stiffness matrix $\mathbf{K}(\boldsymbol{\rho}^0)$, e.g. with all $\rho_i^0 = 1$, $i = 1, \ldots, n_\rho$. The problem (17) is formed by applying the first-order Taylor expansion to the problem (2) at $(\mathbf{u}^0, \boldsymbol{\rho}^0)$ and with filtering. Hence, it can be regarded as the linear approximation of the original problem. To ensure sufficient accuracy for this approximation, the volume constraint imposed cannot be too far from $\sum_{i=1}^{n_\rho} \frac{\rho_i^0}{n_\rho}$. However, our ultimate goal is to satisfy the target volume V_T, and the value of V_T can generally be far from 1, e.g., $V_T = 0.5$. Thus, we adopt an asymptotic process, namely letting the volume constraint gradually approach the target value V_T. In particular, we define a sequence of different volume values $V_m = 1 - m\Delta V$ with $m = 1, 2, \ldots, M$ and $\Delta V = (1 - V_T)/M$. Limiting ΔV sufficiently small, we follow an iterative procedure: in the iteration step m, solve the problem (17) with $\boldsymbol{\rho}^0$ and letting $V = V_m$, and denote its solution as $\boldsymbol{\rho}^1$; invoke Algorithm 1 with the input of $\boldsymbol{\rho}^1$ to find the optimal solution with respect to the volume constraint $V = V_m$ and let this optimal solution be the new $\boldsymbol{\rho}^0$ for the next iteration step $m + 1$; and when $m = 1$, $\boldsymbol{\rho}^0 = \mathbf{1}$. The proposed procedure is summarized in Algorithm 2.

Algorithm 2 ToGbd(V_T, ΔV)

Input: Target volume fraction V_T, incremental change of volume ΔV, number of iterations $M = (1 - V_T)/\Delta V$
Output: Optimal material layout $\boldsymbol{\rho}^*$

1: Initialize the design variables as $\boldsymbol{\rho}^0 = \mathbf{1}$
2: **for** $m = 1, \cdots M$ **do**
3:     Employ the linear system solver to obtain $\mathbf{u}^0 = \mathbf{K}^{-1}(\boldsymbol{\rho}^0)\mathbf{f}$
4:     Solve the problem (17) to obtain $\boldsymbol{\rho}^1$
5:     $\boldsymbol{\rho}^* \leftarrow \text{ToGbdSub}(V_m, \boldsymbol{\rho}^1)$
6:     Prepare for the next iteration with $\boldsymbol{\rho}^0 \leftarrow \boldsymbol{\rho}^*$
7: **end for**
    Return: $\boldsymbol{\rho}^*$

III. Quantum Algorithm

A. From Topology Optimization to QUBO

As discussed above, the original MINLP problem, as in Eq. (2), is relaxed into a series of MILP problems, as in Eq. (15). In this section, we establish the conversion of MILP into QUBO, such that the master problem (15) can be solved on quantum annealers.

First, a slack variable α^j is introduced into each cut in the problem (15) to transform inequality constraints into equality constraints, as:

$$\min_{\boldsymbol{\rho}, \eta, \boldsymbol{\alpha}} \quad \eta$$
$$\text{s.t.} \quad \mathbf{f}^\mathsf{T}\mathbf{u}^j - \sum_{i=1}^{n_\rho} \widetilde{w}_i^j (\rho_i - \rho_i^j) + \alpha^j = \eta, \quad \forall j \in \mathcal{P}(k)$$
$$\sum_{i=1}^{n_\rho} \frac{\rho_i}{n_\rho} = V$$
$$\boldsymbol{\rho} \in \{0,1\}^{n_\rho}$$
$$\alpha^j \geq 0, \quad \forall j \in \mathcal{P}(k)$$
$$(18)$$

Next, the continuous variables η and α^j are replaced with a series of binary variables. To do so, the upper and lower bounds for η and α^j need to be specified first. Note the facts that η is the lower bound of the problem (2), and the GBD has generated an upper bound U for (2). Thus, U can be regarded as the upper bound for η. Further, due to the positive definite property of the differential operator $\mathcal{L}$, the compliance defined as $\mathbf{f}^\mathsf{T}\mathbf{u}$ must be non-negative. The term, $\mathbf{f}^\mathsf{T}\mathbf{u}^j - \sum_{i=1}^{n_\rho} \widetilde{w}_i^j (\rho_i - \rho_i^j)$, is the linear approximation of the original problem (2) at $(\mathbf{u}^j, \boldsymbol{\rho}^j)$, which is close to $\mathbf{f}^\mathsf{T}\mathbf{u}^j$ when $\boldsymbol{\rho}$ satisfies the volume constraint. Hence, $\mathbf{f}^\mathsf{T}\mathbf{u}^j - \sum_{i=1}^{n_\rho} \widetilde{w}_i^j (\rho_i - \rho_i^j) \geq 0$ holds, from which we obtain: $\eta = \mathbf{f}^\mathsf{T}\mathbf{u}^j - \sum_{i=1}^{n_\rho} \widetilde{w}_i^j (\rho_i - \rho_i^j) + \alpha^j \geq \alpha^j \geq 0$. Thus, the lower bounds for both η and α^j can be set zero. From $\alpha^j \leq \eta \leq U$, we can set the upper bound for α^j as U as well. Given their upper and lower bounds specified, the continuous variables η and α^j can be approximated by a series of binary variables, as:

$$\eta \approx \widetilde{\eta}(\mathbf{e}) = U\left(1 - \frac{1}{2^{n_\eta}}\right) e_0 + \sum_{i=1}^{n_\eta} \frac{U}{2^i} e_i$$
$$\alpha^j \approx \widetilde{\alpha}^j(\mathbf{a}^j) = U\left(1 - \frac{1}{2^{n_{\alpha^j}}}\right) a_0^j + \sum_{i=1}^{n_{\alpha^j}} \frac{U}{2^i} a_i^j,$$

where

$$\mathbf{e} \in \{0,1\}^{n_\eta + 1}, \quad \mathbf{a}^j \in \{0,1\}^{n_{\alpha^j} + 1}.$$

The accuracy of this approximation is controlled by n_η and n_{α^j}. Their values can be chosen according to the desired accuracy and/or the number of accessible qubits in quantum annealers. Thus, we obtain a binary programming problem as:

$$\min_{\boldsymbol{\rho}, \mathbf{e}, \mathbf{a}^j} \quad \widetilde{\eta}(\mathbf{e})$$
$$\text{s.t.} \quad \mathbf{f}^\mathsf{T}\mathbf{u}^j - \sum_{i=1}^{n_\rho} \widetilde{w}_i^j (\rho_i - \rho_i^j) + \widetilde{\alpha}^j(\mathbf{a}^j)$$
$$= \widetilde{\eta}(\mathbf{e}), \quad \forall j \in \mathcal{P}(k) \qquad (19)$$
$$\sum_{i=1}^{n_\rho} \frac{\rho_i}{n_\rho} = V$$
$$\boldsymbol{\rho} \in \{0,1\}^{n_\rho}.$$

By such, the constraints can be readily moved into the objective function through the penalty method to obtain the

following QUBO formulation:

$$
\widetilde{\eta}(\mathbf{e}) + \sum_{j \in \mathcal{P}(k)} A^j \left[W^j - \left(\sum_{i=1}^{n_\rho} \widetilde{w}_i^j \rho_i \right) + \widetilde{\alpha}^j(\mathbf{a}^j) - \widetilde{\eta}(\mathbf{e}) \right]^2
$$
$$
+ B \left(\sum_{i=1}^{n_\rho} \frac{\rho_i}{n_\rho} - V \right)^2 ,
$$
(20)

where

$$
W^j = \mathbf{f}^\mathsf{T} \mathbf{u}^j + \sum_{i=1}^{n_\rho} \widetilde{w}_i^j \rho_i^j .
$$

As such, we have cast the MILP problem (15) into QUBO.

Finally, we determine the penalty factors. Theoretically, the penalty factors in (20) should be gradually increased until the optimal solution converges. However, due to the limited machine precision of currently accessible quantum computers (e.g. up to 10^{-6} [44]), the penalty factors cannot be arbitrarily large in practice. On the other hand, the penalty factors scaled with U, the upper bound of η, can effectively direct the quantum annealer to find a solution near the optimal. From our numerical experiments, setting the penalty factors equal to the upper bound U leads to a satisfactory performance, as a trade-off between the machine precision attainable in the hardware used and the magnitude of the penalty factors required for accuracy, i.e., $A^j = U$, $B = U$. It is worth to mention that any inexact solution for the problem (15), smaller than the exact solution due to the violation of the constraints, will not lead to an abnormal termination of the GBD iterations in Algorithm 1, but just result in more iteration steps before reaching the convergence such that $(U - \eta^k)/U < \xi$.

B. A Splitting Approach for Problem Reduction

Solving the QUBO problem in (20) requires $n_{qubit}^l = n_\rho + n_\eta + \sum_{j \in \mathcal{P}(k)} n_{\alpha^j} + |\mathcal{P}(k)| + 1$ logical qubits to embed the problem, which can be a large number. Furthermore, the quadratic terms $[W^j - (\sum_{i=1}^{n_\rho} \widetilde{w}_i^j \rho_i) + \widetilde{\alpha}^j - \widetilde{\eta}]^2$ and $(\sum_{i=1}^{n_\rho} \frac{\rho_i}{n_\rho} - V)^2$ require all-to-all connections, since each evolves all binary variables simultaneously. However, due to the limited connectivity of qubits on the currently available quantum computers [45], all-to-all connections would require much more physical qubits than logical qubits to embed the QUBO problem, making the hardware accessibility even more challenging. Consequently, a practical TO problem formulated as (20) cannot be handled by the current or near-term quantum computers.

To resolve this issue, we further develop a splitting approach in this section. We note that a binary programming problem like (16) or (17) can be efficiently solved by a classical optimizer, as discussed below. In fact, by taking the continuous relaxation of the design variable as $\bar{\boldsymbol{\rho}}$ ($0 \leq \bar{\rho}_i \leq 1$, $i = 1, \ldots, n_\rho$), the binary programming problem like (16) or (17) can be relaxed into a (continuous) linear programming. If we consider the volume constraint $\sum_{i=1}^{n_\rho} \bar{\rho}_i = n_\rho V$ as a cut onto the n_ρ-dimensional cube of the relaxed design variable

$\bar{\boldsymbol{\rho}}$, where $n_\rho V$ is an integer, the feasible set for the linear programming can be represented by a polyhedron P as:

$$
P = \left\{ (\bar{\rho}_1, \ldots, \bar{\rho}_{n_\rho}) \left| \sum_{i=1}^{n_\rho} \bar{\rho}_i = n_\rho V, \quad 0 \leq \bar{\rho}_i \leq 1, \forall i \right. \right\},
$$

where each element of $\bar{\boldsymbol{\rho}}$ must be binary, i.e., $\bar{\rho}_i = 0$ or 1 $\forall i$, at each vertex. Thus, the linear programming's optimal solution, i.e., a vertex of the polyhedron P [46], must also be the optimal solution of the original binary programming problem. Therefore, the problem (16) or (17) can be exactly solved with the complexity of $\mathcal{O}(n_\rho)$ and hence be handled efficiently on classical computers. In contrast, although the problem (19) is also a binary programming problem, the additional $|\mathcal{P}(k)|$ constraints or cuts, other than the volume constraint, make it generally impossible to find the feasible set of the relaxed linear programming as a polyhedron with vertices taking binary values, and hence it still suffers from the NP-hardness.

By leveraging this observation, we propose the following procedure to split the problem (15) into two parts. The first part consists of several sub-problems like (16), which will be solved on classical computers. The second part is a problem similar to (15) but with greatly reduced variables, which will be solved on a quantum computer. By taking only one of the $|\mathcal{P}(k)|$ inequality constraints as well as the volume constraint in (15), we can form $|\mathcal{P}(k)|$ sub-problems like (16) as:

$$
\min_{\boldsymbol{\rho}} \quad \mathbf{f}^\mathsf{T} \mathbf{u}^j - \sum_{i=1}^{n_\rho} \widetilde{w}_i^j (\rho_i - \rho_i^j), \quad \forall j \in \mathcal{P}(k)
$$
$$
\text{s.t.} \quad \sum_{i=1}^{n_\rho} \frac{\rho_i}{n_\rho} = V
$$
$$
\boldsymbol{\rho} \in \{0,1\}^{n_\rho} .
$$
(21)

The solution of (21) is denoted as $\widetilde{\boldsymbol{\rho}}^j$. Note that when $|\mathcal{P}(k)| > 1$, the material layouts obtained in different iterations (e.g., $j_1, j_2 \in \mathcal{P}(k)$) are only different in finite numbers of nodes, i.e., most elements of the design variables can be the same. We denote the index set of those elements as $\mathcal{I}$ such that

$$
\widetilde{\rho}_i^{j_1} = \widetilde{\rho}_i^{j_2}, \quad \forall i \in \mathcal{I} \subseteq \{1, \ldots, n_\rho\}, \quad \forall j_1, j_2 \in \mathcal{P}(k) . \quad (22)
$$

And the complement set of $\mathcal{I}$ is denoted as $\mathcal{I}^\mathcal{C} = \{1, \ldots, n_\rho\} \setminus \mathcal{I}$, which contains the index of elements that have different values for the design variable in different iterations, i.e., $\widetilde{\rho}_i^{j_1} \neq \widetilde{\rho}_i^{j_2}, \forall i \in \mathcal{I}^C, \exists j_1, j_2 \in \mathcal{P}(k)$.

Thus, a reduced problem of (15) can be formed as:

$$
\min_{\boldsymbol{\rho}, \eta} \quad \eta
$$
$$
\text{s.t.} \quad R^j - \sum_{i \in \mathcal{I}^C} \widetilde{w}_i^j (\rho_i - \rho_i^j) \leq \eta, \quad \forall j \in \mathcal{P}(k)
$$
$$
\sum_{i \in \mathcal{I}^C} \frac{\rho_i}{n_\rho} = V - \sum_{i \in \mathcal{I}} \frac{\widetilde{\rho}_i}{n_\rho}
$$
$$
\boldsymbol{\rho} \in \{0,1\}^{|\mathcal{I}^\mathcal{C}|} ,
$$
(23)

where

$$
R^j = \mathbf{f}^\mathsf{T} \mathbf{u}^j - \sum_{i \in \mathcal{I}} \widetilde{w}_i^j (\widetilde{\rho}_i^j - \rho_i^j) . \quad (24)
$$

Compared with the original problem (15), while the number of constraints is identical, the number of design variables has been reduced, from n_ρ to $|\mathcal{I}^\mathcal{C}|$. Although it is difficult to estimate the exact value of $|\mathcal{I}^\mathcal{C}|$ in advance, we anticipate $|\mathcal{I}^\mathcal{C}| \ll n_\rho$ as discussed above and also demonstrated by the numerical experiments in Section IV-B.

By introducing slack variables α^j and approximating the continuous variables η and α^j as series of binary variables, as described in Section III-A, the reduced master problem (23) can be transformed into a binary programming problem as:

$$
\begin{aligned}
\min_{\boldsymbol{\rho}, \mathbf{e}, \mathbf{a}_j, \mathbf{b}} \quad & \widetilde{\eta}(\mathbf{e}) \\
\text{s.t.} \quad & R^j - \sum_{i \in \mathcal{I}^\mathcal{C}} \widetilde{w}_i^j (\rho_i - \rho_i^j) + \widetilde{\alpha}^j(\mathbf{a}^j) \\
& \qquad = \widetilde{\eta}(\mathbf{e}), \quad j \in \mathcal{P}(k) \\
& \sum_{i \in \mathcal{I}^c} \frac{\rho_i}{n_\rho} = \widetilde{V} \\
& \boldsymbol{\rho} \in \{0,1\}^{|\mathcal{I}^\mathcal{C}|}, \mathbf{e} \in \{0,1\}^{n_\eta+1} \\
& \mathbf{a}^j \in \{0,1\}^{n_{\alpha j}+1} ,
\end{aligned} \tag{25}
$$

where

$$
\begin{aligned}
\widetilde{\eta}(\mathbf{e}) =& U \left(1 - \frac{1}{2^{n_\eta}} \right) e_0 + \sum_{i=1}^{n_\eta} \frac{U}{2^i} e_i , \\
\widetilde{\alpha}^j(\mathbf{a}^j) =& U \left(1 - \frac{1}{2^{n_{\alpha j}}} \right) a_{j0} + \sum_{i=1}^{n_{\alpha j}} \frac{U}{2^i} a_i^j , \\
\widetilde{V} =& V - \sum_{i \in \mathcal{A}} \frac{\widetilde{\rho}_i}{n_\rho} .
\end{aligned}
$$

And further through the penalty method, it can be written into the QUBO formulation as:

$$
\begin{aligned}
\widetilde{\eta}(\mathbf{e}) +& \sum_{j \in \mathcal{P}(k)} A^j \left[R^j - \left(\sum_{i \in \mathcal{I}^\mathcal{C}} \widetilde{w}_i^j \rho_i \right) + \widetilde{\alpha}_j(\mathbf{a}_j) - \widetilde{\eta}(\mathbf{e}) \right]^2 \\
&+ B \left[\sum_{i \in \mathcal{I}^c} \frac{\rho_i}{n_\rho} - \widetilde{V} \right]^2 .
\end{aligned} \tag{26}
$$

As $|\mathcal{I}^\mathcal{C}| \ll n_\rho$, the QUBO problem in (26) can now be handled by the current or near-term quantum computers with a small number of qubits and limited connectivity.

The proposed splitting approach is summarized in Algorithm 3.

IV. NUMERICAL RESULTS

A. Solving a Toy Problem for Validation

To validate that the method discussed in Section III-A permits a quantum computer to effectively solve the MILP problem like (15), we consider the following mixed-integer

---

Algorithm 3 ToGbdSubSplitting$(V, \boldsymbol{\rho}^1)$

Input: Volume fraction V, initial material layout $\boldsymbol{\rho}^1$
Output: Optimal material layout $\boldsymbol{\rho}^*$
Utility: Find an optimal material layout with a given volume fraction V and initial material layout $\boldsymbol{\rho}^1$

1: **for** $k = 1, \dots$ **do**
2:      Initialize the upper bound U as $U = +\infty$
3:      Employ the linear solver to obtain $\mathbf{u}^k = \mathbf{K}^{-1}(\boldsymbol{\rho}^k)\mathbf{f}$
4:      **if** $\mathbf{f}^\intercal \mathbf{u}^k < U$ **then**
5:          $U \leftarrow \mathbf{f}^\intercal \mathbf{u}^k$
6:          $\boldsymbol{\rho}^* \leftarrow \boldsymbol{\rho}^k$
7:      **end if**
8:      Generate the Pareto optimal cuts $\mathcal{P}(k)$ according to (14)

9:      **if** $|\mathcal{P}(k)| = 1$ **then**
10:          Solve the master problem (16) for the minimizer η^k and $\boldsymbol{\rho}^{k+1}$ on a classical computer
11:      **else**
12:          Solve the sub-problem (21) for the minimizer $\widetilde{\boldsymbol{\rho}}^j$ for each element in $\mathcal{P}(k)$ on a classical computer
13:          Generate the index sets $\mathcal{I}$ and $\mathcal{I}^\mathcal{C}$ according to (22)
14:          Solve the reduced QUBO problem (26) for the minimizer η^k and $\boldsymbol{\rho}^{k+1}$ on a quantum annealer
15:      **end if**
16:      **if** $(U - \eta^k)/U < \xi$ **then**
17:          **break**
18:      **end if**
19: **end for**
    Return: $\boldsymbol{\rho}^*$

---

programming problem with linear inequality and equality constraints:

$$
\begin{aligned}
\min_{u,v,w,t} \quad & v + w + t + (u - 2)^2 \\
\text{s. t.} \quad & v + 2w + t + u \leq 3 \\
& v + w + t \geq 1 \\
& v + w = 1 \\
& v, w, t \in \{0,1\}, u \in \mathbb{R} .
\end{aligned} \tag{27}
$$

This problem contains one continuous and three binary variables and one equality and two inequality constrains. Its unique optimal solution is known to be: $u = 2, v = 1, w = t = 0$.

According to the proposed methodology, we first transform this problem into a binary programming problem, and then rewrite it in the formulation of QUBO. By noting that $0 \leq u \leq 3$, we can introduce a binary approximation $\widetilde{u}$ for the continuous variable u as:

$$
\widetilde{u}(\mathbf{e}) = 3 \left(1 - \frac{1}{2^{n_u}} \right) e_0 + \sum_{i=1}^{n_u} \frac{3}{2^i} e_i .
$$

Next, we introduce two slack variables α^1 and α^2 to convert the two inequality constraints into equality constraints. By noting that $0 \leq \alpha^1 \leq 3$, $0 \leq \alpha^2 \leq 2$, and $\alpha^2 \in \mathbb{Z}$ since the second inequality constraint only contains integers, we can

obtain their binary representations $\widetilde{\alpha}^1$ and $\widetilde{\alpha}^2$, respectively, and thereby rewrite the MILP as a binary programming problem:

$$\min_{\mathbf{e},\mathbf{a}^1,\mathbf{a}^2,v,w,t} \quad v + w + t + (\widetilde{u} - 2)^2$$
$$\text{s. t.} \quad v + 2w + t + \widetilde{u} + \widetilde{\alpha}^1 = 3$$
$$- v - w - t + \widetilde{\alpha}^2 = -1$$
$$v + w = 1 \tag{28}$$
$$v, w, t \in \{0,1\}$$
$$\mathbf{e} \in \{0,1\}^{n_u+1}$$
$$\mathbf{a}^1 \in \{0,1\}^{n_{\alpha^1}+1}, \mathbf{a}^2 \in \{0,1\}^2 ,$$

where

$$\widetilde{\alpha}^1(\mathbf{a}^1) = 3\left(1 - \frac{1}{2^{n_{\alpha^1}}}\right)a_0^1 + \sum_{i=1}^{n_{\alpha^1}} \frac{3}{2^i} a_i^1$$
$$\widetilde{\alpha}^2(\mathbf{a}^2) = a_0^2 + a_1^2 .$$

Finally, through the penalty method, it can be written into the QUBO as:

$$v + w + t + [\widetilde{u} - 2]^2 + A\left[v + 2w + t + \widetilde{u} - 3 + \widetilde{\alpha}^1\right]^2$$
$$+ A\left[-v - w - t + 1 + \widetilde{\alpha}^2\right]^2 + A\left[v + w - 1\right]^2 , \tag{29}$$

where the penalty factor is set as $A = 4$ in this problem.

The problem (29) was directly submitted to the D-Wave Advantage QPU. The annealing time was set with $20\mu s$, and the sampling was repeated for 1000 times. The final state reached at the end of each annealing was recorded, and the state with the lowest energy among all 1000 samplings was regarded as the ground state of the Hamiltonian defined in (29) and hence the solution to the binary programming problem (28). The continuous variables u and α^1 as well as the integer variable α^2 of the original problem (27) were then recovered from their binary approximations. The final results for the values of different variables are summarized in Table I. To confirm that the results have converged with respect to the parameter settings in the binary approximations, we compare two different groups of values for n_u and n_{α^1}. The results agree with the theoretical solution to the original problem (27), and hence the accuracy of the obtained QC-based solution is validated. An additional note about the continuous variables is made as follows: if we treat the original problem (27) as a continuous optimization problem and solve it for the continuous variables, with all the binary variables fixed to the solution obtained from solving (29), the accuracy can be even better than recovering them from their binary approximations.

B. Solving the Minimal Compliance Problem

In this subsection, we concern a canonical TO problem, the minimum compliance, as formulated in (2). The problem setup follows that in [25]. More specifically, we consider a rectangular material domain with the external loading of a unit point force $\mathbf{F}$ exerted at the top left corner, as illustrated in Figure 2. The solution domain, or the design space, is discretized by a uniform quadrilateral mesh with different resolutions. In Eq. (3), the Young's modulus is $E = 1.0$, and the Poisson ratio is $\nu = 0.3$. The target volume fraction is $V_T = 0.5$, and the incremental change of volume is $\Delta V = \frac{1}{24}$.

The convergence criterion tolerance required in Algorithm 3 is $\xi = 5 \times 10^{-4}$. All values are in reduced units.
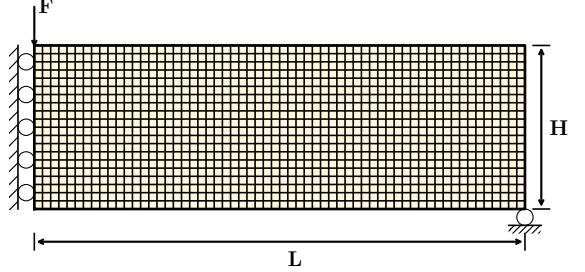

Fig. 2: Schematic of the problem setup for the minimum compliance. The yellow region denotes the design domain discretized by a uniform quadrilateral mesh with the resolution of 60×20, where $L = 60$, $H = 20$, and the external load is a unit point force, $\mathbf{F} = (0, -1)$, exerted at the top leftmost node.

To solve this TO problem, we followed the proposed methodology, consisting of the GBD, conversion of the MILP into QUBO, and the splitting approach. More specifically, Algorithm 2 was executed, but with ToGBDSuB in Line 5 replaced by ToGBDSuBSPLITTING defined in Algorithm 3 to integrate the proposed splitting approach. While the binary programming problems (16), (17) and (21) were solved by invoking the classical MILP solver Gurobi [47], the binary programming problem (25) was solved on the quantum annealer provided by D-Wave. From the validation test in Section IV-A and considering the accuracy of the current quantum devices and the required number of qubits, the parameters for the binary approximations were set as: $n_\eta = n_{\alpha^j} = 10$ in (25). Once the solution of the design variable $\boldsymbol{\rho}^{k+1}$ is determined, the problem (15) can be treated as a continuous optimization problem, from which we can solve for the continuous variable η^k required by Algorithm 3. Alternatively, η^k can be recovered from its binary approximation. We chose the former approach herein for its slightly better accuracy.

To implement the problem (25) on the quantum annealer provided by D-Wave, we examined two different ways. In the first way, we directly embedded the QUBO problem (26) formulated from (25) in the QPU, which is named as "*GBD-Splitting-Direct*". In the second way, we solved the problem (25) by taking advantage of the hybrid solver provided by D-Wave, particularly the constrained quadratic model (CQM), which is hence referred to as "*GBD-Splitting-CQM*". We compare the results in terms of solution quality and wall time, as summarized in Table II, where the discretization resolution is chosen as 60×20; the parameter for filtering, as described in Eq. (12), is set as $r = 2$; T denotes the total wall time spent for executing Algorithm 2 to find the optimal material layout; $\mathbf{f}^\intercal\mathbf{u}$ denotes the objective function's value obtained corresponding to the optimal material layout; and N denotes the total number of binary programming problems invoked by Algorithm 2 and 3. Details about each implementation and our main findings are provided as below.

TABLE I: The results for solving the toy problem (27).

$(n_u, n_{\alpha 1})$	u	v	w	t	α_1	α_2
(5, 5)	1.96875	1	0	0	0	0
(10, 10)	2.00098	1	0	0	0	0

TABLE II: Comparison between different implementations for the reduced binary master problem (25) on the quantum annealer provided by D-Wave, where T denotes the total wall time spent for executing Algorithm 2.

Resolution	n_ρ	r	GBD-Splitting-Direct			GBD-Splitting-CQM		
			T (s)	$\mathbf{f^\intercal u}$	N	T (s)	$\mathbf{f^\intercal u}$	N
60×20	1200	2	234.12	186.3727	80	111.97	178.9243	74

In *GBD-Splitting-Direct*, each sampling is set with $20\mu s$ annealing time, and the sampling is repeated for 1000 times. The final states reached at the end of annealing were recorded, and the state with the lowest energy among 1000 samplings was regarded as the ground state of the Hamiltonian defined in (26) and hence the solution to the binary programming problem (25). From our statistics, totally 80 binary programming problems were invoked, 11 among which are the problem (26) and were solved by the quantum annealer. For each, we carefully examined the number of the resultant reduced variables ($—\mathcal{I}^C—$), the number of physical qubits required for embedding the problem (26), and the solution time spent on the QPU, as summarized in Table III, where the 11 problems are organized according to their sequence of being solved. With the discretization resolution of 60×20, the number of design variables is $n_\rho = 1200$. From the value of $|\mathcal{I}^C|$ reported in Table III, we can see that the splitting approach proposed in Section III-B has greatly reduced the size of the problem to be solved on a quantum computer. Note that the total *logical* qubits needed to represent the problem (26) should also include those required to represent the binary approximations of η and α^j. Thus, the number of logical qubits required in total is: $n_q^r = |\mathcal{I}^C| + n_\eta + 1 + |\mathcal{P}(k)| + \sum_{j \in \mathcal{P}(k)} n_{\alpha^j}$. By reducing $|\mathcal{I}^C|$, n_q^r can be reduced accordingly. If the *physical* qubits have all-to-all connections, the number of physical qubits required would be consistent with n_q^r. However, due to the sparse connectivity provided by the current quantum annealing machines, the number of physical qubits required for embedding the problem, denoted as n_q^e, is in fact much larger than n_q^r [48], as reported in Table III. Also, with the increase of n_q^r, n_q^e can grow very fast [48], although fluctuating due to inhomogeneous connections between qubits. As a result, most of the time spent on the QPU is dominated by the embedding overhead, as indicated by the values of T_{QUBO} in Table III, noting that the total annealing time for 1000 repetitions of sampling is only 20 ms. The current D-Wave Advantage system permits access to no more than 5000 qubits. To constrain n_q^e not beyond 5000, $|\mathcal{I}^C|$ has to be no more than a few hundreds. Thus, even with the splitting approach, the number of design variables in the original problem must be limited to moderate, which makes solving the minimum compliance problem with finer discretization resolutions challenging.

To tackle this challenge, we pursued the second way of implementation, i.e., *GBD-Splitting-CQM*, where the problem (25) was submitted, and the CQM, provided by D-Wave, was called for embedding and solving the problem. All default setups were used when employing the CQM hybrid solver in all the numerical tests. As indicated by the wall time T in Table II, the implementation through CQM greatly saved the entire solving time. The reason for that is the CQM can further reduce the size of the problem embedded on the QPU and in turn save the time spent for embedding, knowing the fact that embedding dominates the time consumed on the quantum devices. In addition, the implementation through CQM results in fewer binary programming problems invoked and a lower value for the objective function. Recalling the discussion about the inexact solution in Section III-B that any inexact solution to the problem (15) can potentially increase the number of GBD iterations, fewer binary programming problem invoked suggest that the solution found by the CQM is closer to the exact solution. Possible reasons for that include: the CQM further reduces the problem embedded on the QPU and makes the resultant optimization problem easier to be accurately solved; in *GBD-Splitting-Direct*, the values chosen for the penalty factors may not be optimal, and the probability of recovering the global optimal solution is highly dependent on the embedding and annealing schedule.

Based on the above comparison and findings, when we scaled up the minimum compliance problem with increasing numbers of design variables (with finer discretization resolutions), we employed the implementation of *GBD-Splitting-CQM*, owing to its better performance in terms of the solution quality, wall time, and capacity to embed larger problems. In particular, we considered two different shapes of material domains: $L : H = 3 : 1$ and $L : H = 2 : 1$, and the discretization resolution varies from 120×40 to 480×240, leading to the number of design variables (n_ρ) varying from 4800 to 115200. The results about the obtained minimum values of the objective function, the associated computing time, and the corresponding optimal material layouts are presented in Table IV and Figure 3 and 4. To verify the QC-based solutions and to assess the benefits of QC for TO, we further compare with the state-of-the-art classical solvers, as discussed in the next subsection.

C. Comparison with Classical Solvers

The classical solvers considered include the SIMP method [25], the TOBS method [27], and the DVTOPCRA method [26]. For the SIMP method, we followed the implementation in [25] with the penalty factor $p = 3$ and the maximum number of iterations set as 1000. We consider the sensitivity filter controlled by a radius size as in (12) along with Heaviside

TABLE III: Statistics about the 11 QUBO problems (26) solved in the implementation of *GBD-Splitting-Direct*, where T_{QUBO} denotes the time spent for embedding and annealing on the QPU.

$|\mathcal{I}^C|$	16	20	34	22	23	27	23	14	32	18	33
n_q^r	49	53	67	55	56	60	56	47	65	51	66
n_q^e	198	234	512	260	559	498	432	172	423	212	616
T_{QUBO} (s)	6.20	7.11	16.63	8.57	29.05	21.01	32.71	4.20	13.23	10.52	33.99

projection [49]. The convergence criterion is controlled by the change of design variables as $\max_i |\Delta\rho_i| < 0.01$. For the TOBS method, the setup was the same as provided in [27], except that the MILP solver was changed to Gurobi [47] with default setups to permit a fair comparison with our proposed solution strategy, where Gurobi is used for solving the binary programming problems (16), (17) and (21). For the DVTOCRA method, the implementation exactly followed [50] for the minimum compliance problem.

The results obtained by *GBD-Splitting-CQM* and by the three classical methods are compared in Table IV and Figure 3 and 4, for different shapes of material domains with different discretization resolutions. In Table IV, T denotes the total wall time spent by each method to find the optimal material layout, including solving all sub-optimization-problems and FEM analysis in (6); $\mathbf{f}^\top\mathbf{u}$ denotes the objective function's value obtained corresponding to the optimal material layout. For all the three classical methods, N denotes the number of iterations; in each iteration, one FEM analysis (6) is performed, and one sub-optimization-problem is solved. In *GBD-Splitting-CQM*, N represents the total number of binary programming problems invoked by Algorithm 2 and 3; one FEM analysis (6) is performed prior to invoking each binary programming problem, as stipulated by Algorithm 2. Therefore, comparing the values of N is essentially comparing among all methods how many times the FEM analysis is conducted, which dominates the computational cost in large-scale TO problems.

By comparison, the proposed solution strategy exhibits the following advantages. First, it turns out requiring the least number of iterations to reach convergence in almost all the test cases. This can be related to the multi-cuts generated by the GBD iterations and the guarantee of finite iterations by the convexity of the problem. Whereas, the other classical methods solve non-convex sub-problems and in turn have no guarantee for convergence within finite iterations. As the discretization resolution increases, the SIMP method with Heaviside projection cannot meet the convergence criterion within the set maximum number of iterations, and the other two methods, TOBS and DVTOPCRA, tend to require more iterations to reach convergence, as shown in Table IV. Second, the minimum values of the objective function found by *GBD-Splitting-CQM* are generally lower than those by the other methods, especially SIMP and DVTOPCRA. Lastly, in terms of the wall time, *GBD-Splitting-CQM* exhibits only a slow growth in computing time and hence a good scaling performance, in contrast to the other three methods. Thus, for solving larger-scale problems, e.g., with the discretization resolutions of 480×160 and 480×240, *GBD-Splitting-CQM* is more efficient than the SIMP-Heaviside and DVTOPCRA methods

and comparable with the TOBS method. Several factors can contribute to that. First of all, the splitting approach proposed in Section III-B restricts the growth of the size of the problem to be embedded and solved on a quantum computer and in turn inhibits the growth of time required for embedding. Next, as the discretization resolution increases, the FEM analysis can be more expensive and in turn dominate the computational cost. As a result, the fewer FEM analysis conducted, the less computing time required. Thus, for higher discretization resolutions, e.g., 480×160 and 480×240, *GBD-Splitting-CQM* is more efficient, as it calls for the least numbers of iterations and hence the least times of FEM analysis. Finally, comparing with the TOBS method, the problem (16) contains less constraints than the MILP problem invoked in the TOBS method and in turn costs less solution time in each iteration.

The resultant optimal material layouts obtained by different methods are shown and compared in Figure 3 and 4, where two different shapes of the design space with the finest discretization resolution are presented. Unlike the nonconvergent SIMP designs with gray regions as shown in Figure 3(b) and 4(b), the *GBD-Splitting-CQM* designs have sharp contrast with clear 0/1 in density. We hence demonstrate the advantage of *GBD-Splitting-CQM* in generating optimal material layouts with clear boundaries, compared against the designs by the method like SIMP that relaxes the binary design variable into a continuous variable. In addition, when compared with the designs yield from the TOBS method, the designs optimized by *GBD-Splitting-CQM* also exhibit better minimal length control. Furthermore, the layouts generated by the DVTOPCRA method display crooked internal structures, which can be unfavorable with respect to yield strength.

## V. CONCLUSION AND DISCUSSION

We have established a quantum annealing-based solution method for solving continuum, nonlinear TO problems. Its key ingredients include the GBD, conversion of MILP into QUBO, a splitting approach, as well as the implementation on the quantum computer through the hybrid solver, CQM, provided by D-Wave. Through solving a canonical TO problem, the minimum compliance, we have assessed its accuracy and efficiency. By comparing with the state-of-the-art classical methods commonly used in the field of TO, we have demonstrated the advantages of our proposed QC-based methodology, with respect to both solution quality and computational efficiency. Considering the run-time penalty for embedding due to the hardware limitations of current quantum annealing machines [48] and given their future improvements in both the number of qubits and the long-range couplers between qubits, the advantage of the proposed QC-based solution method in terms of computing time can be remarkable. We have also shown its

TABLE IV: Comparison between different methods for solving the minimum compliance problem.

Resolution	n_ρ	r	GBD-Splitting-CQM			SIMP-Heaviside			TOBS			DVTOPCRA		
			T (s)	$\mathbf{f}^\intercal\mathbf{u}$	N	T (s)	$\mathbf{f}^\intercal\mathbf{u}$	N	T (s)	$\mathbf{f}^\intercal\mathbf{u}$	N	T (s)	$\mathbf{f}^\intercal\mathbf{u}$	N
120×40	4800	2	134.18	183.6182	74	19.84	188.1759	433	6.42	187.4646	103	7.24	188.1150	139
120×60	7200	2	61.51	75.1399	77	37.58	77.3887	450	7.80	76.0099	83	11.03	76.7379	133
240×80	19200	4	123.34	183.4272	73	132.44	188.8870	530	31.82	185.7086	116	39.82	191.8400	180
240×120	28800	4	88.21	78.2722	89	255.84	78.8675	665	39.00	77.2155	81	51.50	79.6199	145
480×160	76800	8	193.90	185.1250	89	1861.68	190.7774	1000	198.14	185.1229	134	259.85	191.6287	241
480×240	115200	8	261.34	79.3158	90	2929.59	81.1982	1000	259.45	79.1765	95	522.30	81.8421	329
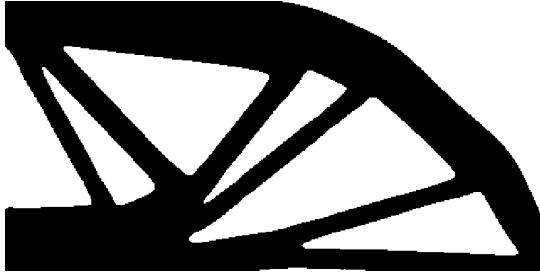
(a) GBD-Splitting-CQM
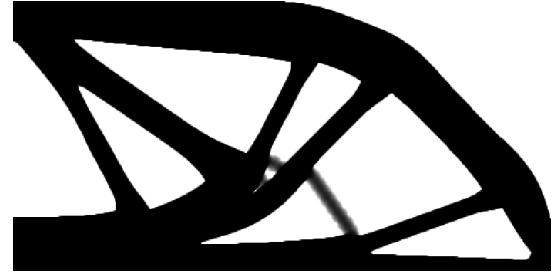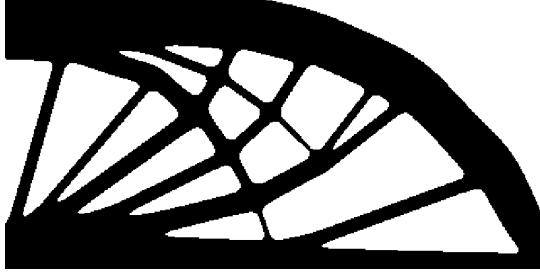
(b) SIMP-Heaviside

(c) TOBS

(d) DVTOPCRA

Fig. 3: The resultant optimal material layouts obtained by different methods, with the design space of ($L = 60$, $H = 30$) and the discretization resolution of 480×240.
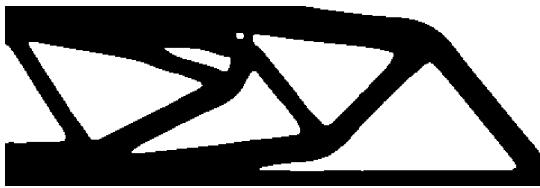
(a) GBD-Splitting-CQM

(b) SIMP-Heaviside

(c) TOBS

(d) DVTOPCRA

Fig. 4: The resultant optimal material layouts obtained by different methods, with the design space of ($L = 60$, $H = 20$) and the discretization resolution of 480×160.

computational efficiency for TO in the sense that the number of FEA runs has been reduced significantly when compared with the classical SIMP method. Future work would exploit this computational efficiency advantages for larger-scale TO problems where FEA cost dominates the optimization run.

This work also presents a new application paradigm for QC and expands the application horizon of QC to include TO, enabling more efficient designs of topology for broad applications.

With increasing complexity, the continuum TO problems can become more challenging. For example, involving the constraints like G_j and H_k in (1) introduces non-volumetric constraints; non-uniform meshes necessitated by discretizing complicated design spaces can lead to non-identical coefficients in the volume constraint associated in the problem (15). The GBD has provided a general enough framework to handle those complexity. Moreover, with non-identical coefficients in the volume constraint, the resultant binary programming problem (16) can be NP-hard and cannot be efficiently solved on classical computers. Whereas, the proposed quantum annealing-based solution method, as for the problem (23), provides a promising approach for solving NP-hard binary programming problems. In addition, due to the similarities between the problem (16) and the sub-problems generated by the TOBS method [27], the proposed solution strategy can also be extended to the TOBS method for dealing with the TO problems subject to complex constraints [51]. In light of these perspectives, introducing QC, particularly quantum annealing, to TO may lead to broad and significant impacts.

The splitting approach developed in the present work has been shown effective for greatly reducing the size of the problem to be embedded in quantum computers, while keeping the solution quality for the optimal material layout. Other approaches, e.g., the multilevel hybrid framework introduced for generic combinatorial optimization [52] and the hierarchical mesh refinement approach [53], can be explored in the future and integrated into the GBD framework to further reduce the cost for solving the problems like (15) on quantum computers.

Acknowledgement

References

[1] M. P. Bendsoe and O. Sigmund, *Topology optimization: theory, methods, and applications*. Springer Science & Business Media, 2003.

[2] M. P. Bendsøe and N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," *Computer methods in applied mechanics and engineering*, vol. 71, no. 2, pp. 197–224, 1988.

[3] J. Wu, O. Sigmund, and J. P. Groen, "Topology optimization of multi-scale structures: a review," *Structural and Multidisciplinary Optimization*, vol. 63, no. 3, pp. 1455–1480, 2021.

[4] T. Borrvall and J. Petersson, "Topology optimization of fluids in Stokes flow," *International journal for numerical methods in fluids*, vol. 41, no. 1, pp. 77–107, 2003.

[5] A. Gersborg-Hansen, O. Sigmund, and R. B. Haber, "Topology optimization of channel flow problems," *Structural and multidisciplinary optimization*, vol. 30, no. 3, pp. 181–192, 2005.

[6] S. Kreissl, G. Pingen, and K. Maute, "Topology optimization for unsteady flow," *International Journal for Numerical Methods in Engineering*, vol. 87, no. 13, pp. 1229–1253, 2011.

[7] E. Moscatelli, L. F. N. de Sá, S. Ranjbarzadeh, R. Picelli, R. dos Santos Gioria, and E. C. N. Silva, "Hybrid geometry trimming algorithm based on integer linear programming for fluid flow topology optimization," *Computers & Fluids*, vol. 244, p. 105561, 2022.

[8] G. Kiziltas, D. Psychoudakis, J. L. Volakis, and N. Kikuchi, "Topology design optimization of dielectric substrates for bandwidth improvement of a patch antenna," *IEEE Transactions on Antennas and Propagation*, vol. 51, no. 10, pp. 2732–2743, 2003.

[9] T. Nomura, K. Sato, K. Taguchi, T. Kashiwa, and S. Nishiwaki, "Structural topology optimization for the design of broadband dielectric resonator antennas using the finite difference time domain technique," *International Journal for Numerical Methods in Engineering*, vol. 71, no. 11, pp. 1261–1296, 2007.

[10] S. Molesky, Z. Lin, A. Y. Piggott, W. Jin, J. Vucković, and A. W. Rodriguez, "Inverse design in nanophotonics," *Nature Photonics*, vol. 12, no. 11, pp. 659–670, 2018.

[11] R. E. Christiansen and O. Sigmund, "Inverse design in photonics by topology optimization: tutorial," *Journal of the Optical Society of America B*, vol. 38, no. 2, pp. 496–509, 2021.

[12] O. Yesilyurt, Z. A. Kudyshev, A. Boltasseva, V. M. Shalaev, and A. V. Kildishev, "Efficient topology-optimized couplers for on-chip single-photon sources," *ACS Photonics*, vol. 8, no. 10, pp. 3061–3068, 2021.

[13] E. M. Dede, "Multiphysics topology optimization of heat transfer and fluid flow systems," in *proceedings of the COMSOL Users Conference*, vol. 715, 2009.

[14] S. Kreissl, G. Pingen, A. Evgrafov, and K. Maute, "Topology optimization of flexible micro-fluidic devices," *Structural and Multidisciplinary Optimization*, vol. 42, no. 4, pp. 495–516, 2010.

[15] S. Sun, P. Liebersbach, and X. Qian, "3D topology optimization of heat sinks for liquid cooling," *Applied Thermal Engineering*, vol. 178, p. 115540, 2020.

[16] P. Kumar, J. S. Frouws, and M. Langelaar, "Topology optimization of fluidic pressure-loaded structures and compliant mechanisms using the Darcy method," *Structural and Multidisciplinary Optimization*, vol. 61, no. 4, pp. 1637–1655, 2020.

[17] J. Liu, A. T. Gaynor, S. Chen, Z. Kang, K. Suresh, A. Takezawa, L. Li, J. Kato, J. Tang, C. C. Wang *et al.*, "Current and future trends in topology optimization for additive manufacturing," *Structural and Multidisciplinary Optimization*, vol. 57, no. 6, pp. 2457–2483, 2018.

[18] X. Qian, "Undercut and overhang angle control in topology optimization: a density gradient based integral approach," *International Journal for Numerical Methods in Engineering*, vol. 111, no. 3, pp. 247–272, 2017.

[19] F. Mezzadri, V. Bouriakov, and X. Qian, "Topology optimization of self-supporting support structures for additive manufacturing," *Additive Manufacturing*, vol. 21, pp. 666–682, 2018.

[20] S. Mukherjee, D. Lu, B. Raghavan, P. Breitkopf, S. Dutta, M. Xiao, and W. Zhang, "Accelerating large-scale topology optimization: state-of-the-art and challenges," *Archives of Computational Methods in Engineering*, vol. 28, no. 7, pp. 4549–4571, 2021.

[21] A. Maruo, T. Soeda, and H. Igarashi, "Topology optimization of electromagnetic devices using digital annealer," *IEEE Transactions on Magnetics*, vol. 58, no. 9, pp. 1–4, 2022.

[22] T. Sato, K. Watanabe, and H. Igarashi, "Multimaterial topology optimization of electric machines based on normalized Gaussian network," *IEEE Transactions on Magnetics*, vol. 51, no. 3, pp. 1–4, 2015.

[23] Y. Sato, R. Kondo, S. Koide, and S. Kajita, "Quantum topology optimization of ground structures using noisy intermediate-scale quantum devices," 2022. [Online]. Available: https://arxiv.org/abs/2207.09181

[24] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan, "Mixed-integer nonlinear optimization," *Acta Numerica*, vol. 22, p. 1–131, 2013.

[25] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund, "Efficient topology optimization in MATLAB using 88 lines of code," *Structural and Multidisciplinary Optimization*, vol. 43, no. 1, pp. 1–16, 2011.

[26] Y. Liang and G. Cheng, "Topology optimization via sequential integer programming and Canonical relaxation algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 348, pp. 64–96, 2019.

[27] R. Picelli, R. Sivapuram, and Y. M. Xie, "A 101-line MATLAB code for topology optimization using binary variables and integer programming,"

Structural and Multidisciplinary Optimization, vol. 63, no. 2, pp. 935–954, 2021.

[28] E. Muñoz and M. Stolpe, "Generalized Benders' decomposition for topology optimization problems," *Journal of Global Optimization*, vol. 51, no. 1, pp. 149–183, 2011.

[29] X. Liu, A. Angone, R. Shaydulin, I. Safro, Y. Alexeev, and L. Cincio, "Layer VQE: A variational approach for combinatorial optimization on noisy quantum computers," *IEEE Transactions on Quantum Engineering*, vol. 3, pp. 1–20, 2022.

[30] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles, "Noise-induced barren plateaus in variational quantum algorithms," *Nature Communications*, vol. 12, no. 1, 2021.

[31] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature Communications*, vol. 9, no. 1, p. 4812, 2018.

[32] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, "Connecting ansatz expressibility to gradient magnitudes and barren plateaus," *PRX Quantum*, vol. 3, no. 1, 2022.

[33] L. Bittel and M. Kliesch, "Training variational quantum algorithms is NP-hard," *Physical Review Letters*, vol. 127, no. 12, p. 120502, 2021.

[34] A. Ajagekar, T. Humble, and F. You, "Quantum computing based hybrid solution strategies for large-scale discrete-continuous optimization problems," *Computers & Chemical Engineering*, vol. 132, p. 106630, 2020.

[35] T. Albash and D. A. Lidar, "Adiabatic quantum computation," *Reviews of Modern Physics*, vol. 90, no. 1, p. 015002, 2018.

[36] A. Ajagekar, K. Al Hamoud, and F. You, "Hybrid classical-quantum optimization techniques for solving mixed-integer programming problems in production scheduling," *IEEE Transactions on Quantum Engineering*, vol. 3, pp. 1–16, 2022.

[37] A. Guillaume, E. Y. Goh, M. D. Johnston, B. D. Wilson, A. Ramanan, F. Tibble, and B. Lackey, "Deep space network scheduling using quantum annealing," *IEEE Transactions on Quantum Engineering*, vol. 3, pp. 1–13, 2022.

[38] K. A. H. Kelany, N. Dimopoulos, C. P. Adolphs, and A. Baniasadi, "Quantum annealing methods and experimental evaluation to the phase-unwrapping problem in synthetic aperture radar imaging," *IEEE Transactions on Quantum Engineering*, vol. 3, pp. 1–20, 2022.

[39] "D-Wave system documentation," 2022. [Online]. Available: https://docs.dwavesys.com/docs/latest/index.html

[40] P. L. McMahon, A. Marandi, Y. Haribara, R. Hamerly, C. Langrock, S. Tamate, T. Inagaki, H. Takesue, S. Utsunomiya, K. Aihara *et al.*, "A fully programmable 100-spin coherent Ising machine with all-to-all connections," *Science*, vol. 354, no. 6312, pp. 614–617, 2016.

[41] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.

[42] O. Sigmund and K. Maute, "Topology optimization approaches," *Structural and Multidisciplinary Optimization*, vol. 48, no. 6, pp. 1031–1055, 2013.

[43] K. Sun, Y. Liang, and G. Cheng, "Sensitivity analysis of discrete variable topology optimization," *Structural and Multidisciplinary Optimization*, vol. 65, no. 8, pp. 1–18, 2022.

[44] "D-Wave QPU Datasheet," 2022. [Online]. Available: https://docs.dwavesys.com/docs/latest/doc_qpu.html

[45] A. Lucas, "Hard combinatorial problems and minor embeddings on lattice graphs," *Quantum Information Processing*, vol. 18, no. 7, pp. 1–38, 2019.

[46] D. Bertsimas and J. N. Tsitsiklis, *Introduction to linear optimization*. Athena Scientific Belmont, MA, 1997, vol. 6.

[47] Gurobi Optimization, LLC, "Gurobi optimizer reference manual," 2022. [Online]. Available: https://www.gurobi.com

[48] M. S. Könz, W. Lechner, H. G. Katzgraber, and M. Troyer, "Embedding overhead scaling of optimization problems in quantum annealing," *PRX Quantum*, vol. 2, p. 040322, 2021.

[49] O. Sigmund, "Morphology-based black and white filters for topology optimization," *Structural and Multidisciplinary Optimization*, vol. 33, no. 4, pp. 401–424, 2007.

[50] Y. Liang and G. Cheng, "Further elaborations on topology optimization via sequential integer programming and Canonical relaxation algorithm and 128-line MATLAB code," *Structural and Multidisciplinary Optimization*, vol. 61, no. 1, pp. 411–431, 2020.

[51] R. Sivapuram, R. Picelli, and Y. M. Xie, "Topology optimization of binary microstructures involving various non-volume constraints," *Computational Materials Science*, vol. 154, pp. 405–425, 2018.

[52] H. Ushijima-Mwesigwa, R. Shaydulin, C. F. Negre, S. M. Mniszewski, Y. Alexeev, and I. Safro, "Multilevel combinatorial optimization across quantum architectures," *ACM Transactions on Quantum Computing*, vol. 2, no. 1, pp. 1–29, 2021.

[53] K. Svanberg and M. Werme, "Sequential integer programming methods for stress constrained topology optimization," *Structural and Multidisciplinary Optimization*, vol. 34, no. 4, pp. 277–299, 2007.