A SUBGRADIENT APPROACH FOR CONSTRAINED BINARY PROGRAMMING VIA QUANTUM ADIABATIC EVOLUTION

SAHAR KARIMI AND POOYA RONAGH

ABSTRACT. In an earlier work [14], Ronagh et al. propose a method for solving the Lagrangian dual of a constrained binary quadratic programming problem via quantum adiabatic evolution using an outer approximation method. This should be an efficient prescription for solving the Lagrangian dual problem in the presence of an ideally noise-free quantum adiabatic system. Current implementations of quantum annealing systems demand methods that are efficient at handling possible sources of noise. In this paper we consider a subgradient method for finding an optimal primal-dual pair for the Lagrangian dual of a polynomially constrained binary polynomial programming problem. We then study the quadratic stable set (QSS) problem as a case study. We see that this method applied to the QSS problem can be viewed as an instance-dependent penalty-term approach that avoids large penalty coefficients. Finally, we report our experimental results of using the D-Wave 2X quantum annealer and conclude that our approach helps this quantum processor to succeed more often in solving these problems compared to the usual penalty-term approaches.

1. INTRODUCTION

In this paper, we consider solving a polynomially constrained binary polynomial programming (CBPP) problem stated formally as

$$\begin{align*}
\text{max} & \quad f(x), \\
\text{s.t.} & \quad g_i(x) = 0 \quad \text{for} \quad i = 1, \ldots, m, \\
& \quad h_j(x) \leq 0 \quad \text{for} \quad j = 1, \ldots, p, \\
& \quad x \in \mathbb{B}^n,
\end{align*}$$

(CBPP)

where \( f, g_i \) for \( i = 1, \ldots, m \), and \( h_j \) for \( j = 1, \ldots, p \) are a finite number of polynomials on \( \mathbb{B}^n = \{0,1\}^n \).

Note that since for any three binary variables \( x, y, z \in \mathbb{B} \), the locus

$$\text{argmax} \quad 2z(x + y) - 3z - xy$$

is identical to the vanishing locus of \( xy - z = 0 \) on \( \mathbb{B}^3 \), we may without loss of generality assume that all binary polynomial programming problems have at most a quadratic objective and are

---

(Sahar Karimi, Pooya Ronagh) 1QB INFORMATION TECHNOLOGIES, SUITE 458, 550 BURRARD STREET, VANCOUVER, BC, V6C 2B5, CANADA.

E-mail addresses: sahar.karimi@1qbit.com, pooya.ronagh@1qbit.com.

Date: June 1, 2016.

Key words and phrases. Adiabatic quantum computation, Constrained integer programming, Lagrangian duality, Gradient descent, Subgradient method.
constrained:

\[
\begin{align*}
\text{max} & \quad f(x), \\
\text{s.t.} & \quad g_i(x) = 0 \quad \text{for} \quad i = 1, \ldots, m, \\
& \quad h_j(x) \leq 0 \quad \text{for} \quad j = 1, \ldots, p, \\
& \quad x \in \mathbb{B}^n,
\end{align*}
\]

where $f$, and all $g_i$, and $h_j$ are of degree at most two. Note that any quadratic function of the form $x^t Ax + b^t x + c$ on $x \in \mathbb{B}^n$ can be written as $x^t (A + \text{diag}(b)) x + c$ by the fact that $x_i^2 = x_i$.

A subclass of CBQP is the quadratic stable set (QSS) problem. Let $G = (V, E)$ be a graph with vertex set $V$ and edge set $E$. $S \subseteq V$ is a stable set of $G$ if the subset of edges with both endpoints in $S$ is empty. Let $W$ define a matrix of weights between each pair of vertices, and $A$ stand for the adjacency matrix of graph $G$. The following is a formal presentation of the QSS problem:

\[
\begin{align*}
\text{max} & \quad x^t W x, \\
\text{s.t.} & \quad x^t Ax = 0, \\
& \quad x \in \mathbb{B}^n.
\end{align*}
\]

Unlike its well known linear counterpart, i.e., the stable set problem (also known as the maximum independent set problem), the QSS problem is more contemporary and has been addressed less in the literature (cf. [6], [12], and references therein).

A common technique for solving (QSS) is penalizing the independence constraint in the objective function and solving the resulting unconstrained problem instead. It is easy to argue that for a sufficiently large penalty coefficient, the unconstrained problem attains the optimal solution of (QSS). However, when aiming to solve the unconstrained problem using a quantum annealer (e.g., the D-Wave 2X), we wish to limit the value of the penalty coefficient to reduce the error and increase the chance of attaining the optimal solution. The above explanation shows that such large penalty coefficients correspond to dual variables for (QSS). We will see that the smallest values of optimal dual variables are generally much smaller than theoretically derived penalty coefficients. This is suggestive of substituting the penalty coefficients method by a subgradient method for iteratively solving the Lagrangian relaxations of (QSS) until a primal-dual pair is achieved.

In this paper we develop a method for solving the Lagrangian dual of a CBPP problem using a subgradient descent approach and quantum annealing. Once applied to (QSS) this method terminates in strong duality, hence solving (QSS) to optimality without a need for a branch-and-bound scheme. In fact, the results presented in this paper apply to a more general form of the QSS problem, in which $A$ is not necessarily an adjacency matrix and entries of $A$ can take any non-negative values, i.e., a problem similar to formulation (QSS) in which $A \geq 0$. We refer to this variant of the QSS problem as the generalized quadratic stable set (GQSS) problem.

The paper is organized as follows. In Section 2, we review the quantum adiabatic approach to solving unconstrained binary quadratic programming (UBQP) problems, and the subgradient descent method for solving the Lagrangian dual problem. In Sections 3 and 4 we focus on the QSS
problem; we present the theoretical bounds on the penalty coefficients as well as the iterative methods for solving the QSS problem via solving its Lagrangian dual. Our experiments are described and numerical results are reported in Section 5. In Section 6, we discuss the application of this method in branch-and-bound frameworks and in multi-constrained integer programming problems. Finally, in Section 7, we state our concluding remarks.

2. Overview

2.1. The quantum adiabatic approach to solving UBQP problems. We refer the reader to [14] for a short introduction on quantum adiabatic computation. For a more extensive study, we refer the reader to [4, 5] for the proposal of a quantum adiabatic algorithm by Farhi et al., and to [15] for an exposition on its computational aspects.

These references suggest that practical quantum hardware can yield a significant quantum speedup in certain integer programming problems. In particular, quantum annealers manufactured by D-Wave Systems Inc. solve a spin glass model problem where couplings connect pairs of quantum bits [9]. These annealers solve UBQP problems,

\[(UBQP) \quad \max_{x \in \mathbb{B}^n} x^t Q x,\]

where nonzero coefficients of the quadratic form create a subgraph of a sparse graph structure known as the Chimera graph [2]. Using graph minor embedding [3] and degree reduction techniques [8], one might assume that these machines can solve any unconstrained binary polynomial programming problem.

\[(UBPP) \quad \max_{x \in \mathbb{B}^n} f(x),\]

where \(f(x)\) is any polynomial in real coefficients.

Our goal is to broaden the scope of application of quantum annealers by designing algorithms for solving constrained binary programming problems that work in conjunction with such integer programming oracles. In [14] we propose a method for solving CBQP problems using a branch-and-bound framework in which the bounding strategy is solving the Lagrangian dual of the primal problem by successive application of quantum adiabatic evolution. The method described in [14] shows a fast rate of convergence to solution of the dual problem in every node of the branch-and-bound tree and provides a tight bound that drastically reduces the number of nodes traversed by the algorithm.

However, it is important to mention that quantum annealers are coupled to an environment, and this significantly affects their performance. Albash et. al. propose a noise model for D-Wave devices [1]. This model includes the control noise of the local field and couplings of the chip, as well as the effect of the cross-talk between qubits that are not coupled. In [1], it is concluded that despite the thermal excitations and small value of the ratio of the single-qubit decoherence time to the annealing time, an open-system, quantum-dynamical description of the D-Wave device that starts from a quantized energy-level structure is well justified. The design of benchmark instances
that can detect quantum speedup or any quantum advantage of a quantum annealer in comparison to state-of-the-art classical algorithms is studied by Katzgraber et. al. [10]. Zhu et. al. [16] show that increasing the classical energy gap beyond the intrinsic noise level of the machine can improve the success of the D-Wave Two quantum annealer, at the cost of producing considerably easier benchmark instances. We refer the reader to [11] for the practicality and best practices in using D-Wave devices.

In this paper, we suggest a method for solving the Lagrangian dual of a BPP problem that has a slower rate of convergence but shows more success in practical applications of noisy quantum annealers than that of [14]. Finally, we propose that our suggested methods are easily applicable to future quantum annealers that will provide higher-degree interactions of quantum bits.

2.2. The Lagrangian dual problem. The Lagrangian Dual (LD) of (CBPP) is

\[
\text{(LD)} \quad \min_{\lambda \in \mathbb{R}^m, \mu \in \mathbb{R}^p} d(\lambda, \mu),
\]

where \(d(\lambda, \mu)\) is evaluated via the Lagrangian relaxation

\[
(L_{\lambda,\mu}) \quad \quad d(\lambda, \mu) = \max_{x \in B^n} L(x, \lambda, \mu) = f(x) + \lambda^t g(x) + \mu^t h(x).
\]

Here \(g(x) = (g_1(x), \ldots, g_m(x))^t\) and \(h(x) = (h_1(x), \ldots, h_p(x))^t\) are the multivariable functions \(g : \mathbb{R}^n \to \mathbb{R}^m\) and \(h : \mathbb{R}^n \to \mathbb{R}^p\) with polynomial entries. The function \(d(\lambda, \mu)\) is the maximum of a finite set of linear functions of \(\lambda\) and \(\mu\) and hence it is convex and piecewise linear.

**Lemma 1** (Weak Duality). The optimal value of (LD) is an upper bound for the optimal value of (CBPP).

**Proof.** Straightforward calculations show that given any fixed choice of \(\lambda \in \mathbb{R}^m\) and \(\mu \in \mathbb{R}^p\),

\[
v := \max_{x \in B^n} \{ f(x) : g(x) = 0, h(x) \leq 0 \} \\
\leq \max_{x \in B^n} \{ f(x) : \lambda^t g(x) = 0, \mu^t h(x) \geq 0 \} \\
\leq \max_{x \in B^n} \{ f(x) + \lambda^t g(x) + \mu^t h(x) \}
\]

\[\square\]

2.3. Subgradient method. Given any fixed tuple of Lagrange multipliers \((\lambda, \mu)\), the value of \(d(\lambda, \mu)\) is the solution to a UBPP problem. This is the key fact in the method explained here.

To solve (LD) an iterative approach may be employed. At \(k\)-th iteration of the algorithm, \(d(\lambda, \mu)\) is evaluated using a quantum annealing process and a primal solution \(x_k^*\) is attained. Note that \(d(\lambda, \mu)\) is a convex function and the subgradient is a descent direction for \(L\) because of which we can use a subgradient descent to the local (and hence global) minimum of \(L\). The update rule for
the multipliers in the \( k \)-th iteration of the algorithm will follow [13]:
\[
\begin{align*}
\lambda^{k+1} &= \lambda^k + s^k \frac{\nabla_{\lambda} L(x^*_k)}{\| \nabla_{\lambda} L(x^*_k) \|} \\
\mu^{k+1} &= P^- \left( \mu^k + s^k \frac{\nabla_{\mu} L(x^*_k)}{\| \nabla_{\mu} L(x^*_k) \|} \right),
\end{align*}
\]
where the projection \( P^- \) keeps a \( p \)-dimensional vector \( \mu \) in the negative orthant:
\[
P^-(\mu) = (\min(0, \mu_1), \ldots, \min(0, \mu_p)).
\]
The algorithm is essentially [13] Procedure 3.1, where Step 1 is performed by quantum annealing. Note that the performance of the algorithm depends on the step sizes \( (s^k)_{k=1,2,\ldots} \).

**Algorithm 1** (Quantum Gradient Descent).

initialize: Lagrange multipliers \((\lambda^0, \mu^0)\) and \( k = 0 \)

until termination do

solve \((L_{\lambda, \mu})\) using a quantum annealing device

find \((\lambda^{k+1}, \mu^{k+1})\) using the update rule (2)

\( k \leftarrow k + 1 \)

In what follows, we will apply this method to solving the Lagrangian dual of (QSS). We will suggest experimental step sizes for the case of (QSS) in subsequent sections.

3. The Quadratic Stable Set Problem

In this section we investigate solving techniques for the generalized quadratic stable set problem defined as

\[
\begin{align*}
\max & \quad x^t W x, \\
\text{s.t.} & \quad x^t A x = 0, \\
& \quad x \in \mathbb{B}^n,
\end{align*}
\]
where \( W \in \mathbb{R}^{n \times n} \) and \( A \succeq 0 \). If we have several constraints of the form \( x^t A^{(i)} x = 0 \) for \( i \in I \) in (GQSS), we can combine them into a single constraint of the form \( x^t (\sum_{i \in I} A^{(i)}) x = 0 \), hence reducing it to the form mentioned above. Moreover, without loss of generality, we may assume that all of the matrices in the quadratic terms in (GQSS) are symmetric since \( x^t Q x = \frac{1}{2} x^t (Q + Q^t) x \) for any \( Q \in \mathbb{R}^{n \times n} \). Finally, notice that for any \( i, j \) where \( A_{ij} \neq 0 \), we have \( x_i x_j = 0 \); therefore, we may preprocess (GQSS) such that \( W \bullet A = 0 \) by setting \( W_{ij} \) corresponding to nonzero \( A_{ij} \) equal to zero, where \( \bullet \) denotes the Hadamard (or entry-wise) product. The discussion above is summarized below in a list of assumptions on (GQSS):

**Assumptions on (GQSS):**

1. Problem (GQSS) has only a single constraint \( x^t A x = 0 \), where \( A \succeq 0 \).
2. Matrices \( A \) and \( W \) are symmetric.
3. \( W \bullet A = 0 \).
3.1. **Penalty methods for GQSS problems.** It is obvious that for sufficiently large \( \lambda \), the UBQP problem

\[(GQSS_{\lambda}) \quad L(\lambda) := \max_{x \in \mathbb{B}^n} x^t Wx - \lambda (x^t Ax) \]
solves problem \([GQSS]\). To use a quantum annealer to solve this problem it is important to choose the smallest possible \( \lambda \) since quantum annealers are analogue processing units with intrinsic noise. Therefore, the presence of very large or very small coefficients in the objective function decreases the success of these devices in finding optimal solutions to an optimization problem.

Let \( \mathcal{N}(i) := \{ j \neq i : a_{ij} \neq 0 \} \) and \( W^+ \) be the matrix containing non-negative entries of \( W \). Note that by Assumption 3, \( W_{ik} = 0 \) (hence \( W^+_{ik} = 0 \)) for all \( k \in \mathcal{N}(i) \).

**Proposition 1.** Let \( \lambda_i = \frac{W^+_{ii} + \sum_{j \in \mathcal{N}(i) \neq i} W^+_{ij}}{\min_{j:A_{ij} \neq 0} A_{ij}} \). For any \( \lambda > \bar{\lambda} := \max_i \lambda_i \), formulation \((GQSS_{\lambda})\) solves \((GQSS)\).

**Proof.** Our proof is by contradiction. Suppose \( x^* \) is the optimal solution of \((GQSS_{\lambda})\) but it is not optimal for \((GQSS)\); thus \((x^*)^t Ax^* \neq 0\), meaning that \( \exists p, q, \) such that \( A_{pq}x_p^*x_q^* > 0 \), i.e., \( A_{pq} \neq 0 \) and \( x_p^*x_q^* = 1 \). We argue that by setting \( x_p^* = 0 \), we can improve the objective value; hence we reach the contradiction that \( x^* \) was optimal for \((GQSS_{\lambda})\). Let \( L(\lambda, x) := x^t Wx - \lambda (x^t Ax) \), and \( \tilde{x}^* \) be the vector attained by setting \( x_p^* = 0 \). It is easy to confirm that

\[
L(\lambda, \tilde{x}^*) = L(\lambda, x^*) + 2 \left( -\frac{W_{pp}^+}{2} - \sum_{j \notin \mathcal{N}(p) \neq p} W^+_{pj}x_j \right) \\
+ 2\lambda \left( A_{pq} + \sum_{j \notin \mathcal{N}(p) \neq q} A_{pj}x_j \right) \\
\geq L(\lambda, x^*) + 2 \left( -\frac{W_{pp}^+}{2} - \sum_{j \notin \mathcal{N}(p) \neq p} W^+_{pj}x_j \right) + 2\lambda A_{pq} \\
> L(\lambda, x^*),
\]

where the first inequality is satisfied by the fact that \( -W_{ij} \geq 0 \) for \( ij \) not appearing in \( W^+ \) and \( 2\lambda \sum_{j \notin \mathcal{N}(p) \neq q} A_{pj}x_j \geq 0 \); and the second inequality is a result of our choice of \( \lambda \), i.e.,

\[
2\lambda A_{pq} > W_{pp}^+ + 2 \sum_{j \notin \mathcal{N}(p) \neq p} W^+_{pj}.
\]

**Remark.** One may easily confirm that in the absence of Assumption 3, \( \lambda_i \) in Proposition \([\square]\) can be modified so that the conclusion stays valid.

\[
\lambda_i = \frac{W^+_{ii} + \sum_{j \notin \mathcal{N}(i) \neq i} W^+_{ij} + \max_{j \in \mathcal{N}(i)} W^+_{ij}}{\min_{j:A_{ij} \neq 0} A_{ij}}
\]

**Remark.** The bound derived in Proposition \([\square]\) is tight. Consider the following graph:
In the above graph, solid and dotted lines stand for existing and non-existing edges, respectively; $A \in \mathbb{R}^{n \times n}$ is a binary matrix with entries 1 corresponding to existing edges and 0 otherwise; and weights matrix $W$ is zero everywhere except for pairs $\{1, 2\}$ and $\{3, 4\}$, on which the weight is $\omega$. Note that if $\lambda \leq \omega$, then $x = (1, 1, 1, 1, 0)$ would be optimal for (GQSS), while it is infeasible for (GQSS).

In the hope of decreasing the penalty coefficient and improving the chance of observing the optimal solution, we use a separate $\lambda_{ij}$ for each nonzero entry of $A$, i.e., solving (GQSS$_\Lambda$)

$$L(\Lambda) := \max_{x \in \mathbb{B}^n} x^t W x - x^t (\Lambda \cdot A) x,$$

instead of (GQSS), where $\Lambda \in \mathbb{R}^{n \times n}$. We assume that $\Lambda$ is symmetric, similar to $A$. The following proposition, which is analogous to Proposition 1, shows how we can guarantee solving (GQSS) via (GQSS$_\Lambda$) by the proper choice of $\Lambda$.

**Proposition 2.** Let $\lambda_i = \frac{w_i^+}{2} + \sum_{j \notin N(i) \neq i} w_{ij}^+$, where $N(i)$ is as defined before. The problem (GQSS$_\Lambda$) solves (GQSS) for matrix $\Lambda = [\lambda_{ij}]$, where $\lambda_{ij} = \lambda_{ji} > \frac{\max\{\lambda_i, \lambda_j\}}{A_{ij}}$ for those $i$ and $j$ where $A_{ij} \neq 0$, and $\lambda_{ij} = 0$ otherwise.

The proof of the above proposition is very similar to the proof of Proposition 1 and so it is omitted here to avoid repetition. Note that the biggest entry of $\Lambda$ is equal to $\tilde{\lambda}$ from Proposition 1.

Even though the bounds of Propositions 1 and 2 are tight, when using the D-Wave 2X to solve (GQSS$_\Lambda$), we observed that these bounds fail to achieve the solution in many of our test cases. In the next section, we see that the subgradient method for solving the Lagrangian dual of (GQSS) can be viewed as a way of finding smaller penalty coefficients that are instance dependent and improve the performance of the D-Wave 2X in solving (GQSS).
4. Solved \textbf{(GQSS)} with quantum subgradient descent

Note that by $A \geq 0$, we have $x^tAx \geq 0$ for all $x \in \mathbb{B}^n$. Therefore, we can substitute our equality constraint $x^tAx = 0$ with an inequality constraint as in the alternative formulation

$$\text{max } x^tWx ,$$

$$\text{s.t. } x^tAx \leq 0 ,$$

$$x \in \mathbb{B}^n ,$$

with Lagrangian dual

$$\text{min } \lambda \geq 0 \text{ max } x^tWx - \lambda (x^tAx).$$

**Proposition 3** (Strong duality). An optimal dual solution of (D\textit{GQSS}) corresponds to an optimal primal solution of (GQSS).

**Proof.** In fact, a linear programming formulation of (D\textit{GQSS}) is

$$\text{min } \delta ,$$

$$\text{s.t. } \delta \geq x^tWx - \lambda (x^tAx) \forall x \in \mathbb{B}^n ,$$

$$\lambda \geq 0, \delta \in \mathbb{R}.$$

For every $x \in \mathbb{B}^n$, every cut $y = x^tWx - \lambda (x^tAx)$ has a non-positive slope. Since the constraint $x^tAx \leq 0$ is feasible (for example at $x = 0$) this dual problem is bounded and at its optimal solution the mentioned constraint is turned on. This proves strong duality. \[ \square \]

Recall that the performance of Algorithm 1 depends heavily on the step-size schedule. In this section, we are looking at a few iterative schemes to find a primal-dual solution $(x^*, \lambda^*)$ that solves (D\textit{GQSS}). Note that since $x^tAx$ is always positive, the direction of the negative of the subgradient vectors is always in the positive direction of the $\lambda$ axis. Therefore, for solving (D\textit{GQSS}), we need only an increasing sequence $\{\lambda_k\}$ of Lagrange multipliers. The subgradient method then always terminates in strong duality with a zero subgradient.

**Remark.** In view of this Lagrangian duality, the results of the previous section can be interpreted as formulae for sufficiently large Lagrange multipliers for which strong duality occurs. However, given specific instances of (GQSS), the smallest Lagrange multipliers at which an optimal primal-dual pair $(x^*, \lambda^*)$ is observed may be much smaller than what these formulae suggest.

4.1. Newtonian method. This method is based on the update rule

$$\lambda^k = (x^{k-1})^tWx^{k-1} \frac{(x^{k-1})^tA(x^{k-1})}{(x^{k-1})^tAx^{k-1}} ,$$

for all $k \geq 1$ and starting from $\lambda^0 = 0$.

**Algorithm 2.**

initialize: $k = 0$, $\lambda^0 = 0$, and $x^0 = \arg\max_{x \in \mathbb{B}^n} x^tWx$

\textbf{while} $(x^k)^tAx^k \neq 0$ \textbf{do}


\[ k = k + 1 \]
\[ \text{find } \lambda^k \text{ using updating rule (4)} \]
\[ \text{let } x^k := \arg \max_{x \in B^n} x^t W x - \lambda^k(x^t A x) \]

**Proposition 4.** The sequence \( \{\lambda^k\} \) generated by Algorithm 2 is an increasing sequence.

**Proof.** Our proof is by induction. Note that \( x^1 W x = 0 \) for \( x = 0 \). Therefore, \( (x^1)^t W x^1 \geq 0 \); as a result, \( \lambda^1 \geq 0 = \lambda^0 \). By optimality of \( x^{k+1} \), we have

\[
(x^{k+1})^t W x^{k+1} - \lambda^k (x^{k+1})^t A x^{k+1} \geq (x^k)^t W x^k - \lambda^k (x^k)^t A x^k = 0;
\]

therefore,

\[
\frac{(x^{k+1})^t W x^{k+1}}{(x^{k+1})^t A x^{k+1}} - \lambda^k = \lambda^{k+1} - \lambda^k \geq 0.
\]

Notice that a feasible solution \( x_f \) to (GQSS) provides a lower bound \( x_f^t W x_f \) for the Lagrangian dual of the problem. We can, therefore, modify the Newtonian method by replacing Equation (4) with

\[
\lambda^k = \frac{(x^{k-1})^t W x^{k-1} - (x_f)^t W x_f}{(x^{k-1})^t A x^{k-1}}.
\]

This suggests a modification of Algorithm 2 that we refer to as the *modified Newtonian method*. At iteration \( k \), after obtaining a solution corresponding to \( \lambda^k \), \( x_i \)'s are greedily set to zero until we reach a feasible solution, and the best feasible solution attained so far is updated accordingly. The best feasible solution is then used in (5) for finding the next \( \lambda \).

The Newtonian method is already an improvement over the theoretical bounds of Propositions 1 and 2 and can attain the solution in some of the instances that the D-Wave 2X did not see the solution for the theoretical bounds. The method occasionally takes large steps, especially towards the end of the algorithm, and exceeds the error threshold. To prevent this behaviour, in the remainder of this section, we suggest incrementing \( \lambda \) with more-controlled step sizes.

**4.2. Incremental method.** In the incremental method, the updating rule for \( \lambda \) is as follows:

\[
\lambda^{k+1} = \lambda^k + \delta s,
\]

where \( \delta \leq 1 \) is a given constant. \( \delta = 1 \) gives a fixed step-size update, and \( \delta < 1 \) is a geometric update in which the step size shrinks as the algorithm proceeds.

In utilizing a noisy quantum annealer, it is recommended to use the following termination criterion: the number of feasible solutions we wish to see before termination is given to the algorithm; in Algorithm 3, it is FeasCnt. After the termination, we pick the best observed feasible solution.

**Algorithm 3.**

- **given:** FeasCnt, \( \delta \leq 1 \), \( \lambda^p \), and \( s_\lambda \)
- **initialize:** \( \lambda^0 = \lambda^p \) and \( \text{cnt} = 0 \).
for $k = 1, 2, \ldots$ do
\begin{align*}
\lambda^k &= \lambda^{k-1} + s_{\lambda} \\
s_{\lambda} &= \delta s_{\lambda} \\
&\text{let } x^k := \arg \max_{x \in B^n} x^t W x - \lambda^k (x^t A x) \\
&\text{if } (x^k)^t A x^k = 0 \\
&\quad \text{cnt} = \text{cnt} + 1 \\
&\text{if } \text{cnt} \geq \text{FeasCnt} \\
&\quad \text{terminate}
\end{align*}

The incremental method avoids the problem of taking long jumps, unlike the Newtonian method. The shortcoming is that it is highly dependent on $s_{\lambda}$ and may take many iterations if $\lambda_p$ and/or $s_{\lambda}$ are too small. In what follows, we propose a hybrid technique that combines the advantages of both the Newtonian and incremental methods.

4.3. Hybrid method. In the hybrid method, the step sizes are proportionate to the length of the subgradients. Suppose that at iteration $k$ we have $\lambda^k$ and
\begin{equation}
(7) \quad x^k := \arg \max_{x \in B^n} x^t W x - \lambda^k (x^t A x).
\end{equation}
Then,
\begin{equation}
(8) \quad L(\lambda) \geq (x^k)^t W x^k - \lambda (x^k)^t A x^k,
\end{equation}
and the gradient of the right-hand side, $\delta_k = (x^k)^t A x^k$, is a subgradient for $L$. In our hybrid scheme, we set
\begin{equation}
(9) \quad \lambda^{k+1} = \lambda^k + s^k_{\lambda}, \quad \text{for } s^k_{\lambda} = \tilde{\alpha} \delta^k.
\end{equation}

Similar to the incremental method, in the hybrid method we wish to collect a number of feasible solutions before termination. Because $\delta^k$ vanishes when we see a feasible solution, we switched to the incremental method with $\lambda_p = \lambda^k$ and a given $s_{\lambda}$ after reaching the first feasible solution. The complete algorithm is given below.

Algorithm 4.
\begin{itemize}
  \item given: $s_{\lambda}$, FeasCnt
  \item initialize: $k = 0, \lambda^0 = 0, x^0 = \arg \max_{x \in B^n} x^t W x$, and $\tilde{\alpha}$
  \item while $(x^k)^t A x^k \neq 0$ do
    \begin{align*}
    k &\gets k + 1 \\
    \delta^{k-1} &= (x^{k-1})^t A x^{k-1} \\
    \lambda^k &= \lambda^{k-1} + \tilde{\alpha} \delta^{k-1} \\
    x^k &= \arg \max_{x \in B^n} x^t W x - \lambda^k (x^t A x)
    \end{align*}
  \item go to Algorithm 3 with $\lambda_p := \lambda^k$, $s_{\lambda}$, $\delta = 1$, and FeasCnt.
\end{itemize}
In this section, we report our experimental results of solving (GQSS) using both the penalty methods of Section 3.1 and the subgradient methods of Section 4. The test instances of (GQSS) were generated randomly, with matrices $W$ taking integer entries between $-5$ and $5$, and $A$ generated as sparse binary matrices with a sparsity of $0.4$.

In our experiments, all algorithms were implemented using Matlab 2014b. The code was run under OS X El Capitan operating a 2.5 GHz Intel Core i5 processor, with 16 GB of memory. All queries to the D-Wave 2X were handled by D-Wave Sapi 2.2.1. In every query, the sapiSolveIsing function was used with solutions post-processed in the optimization mode. The ferromagnetic chain strengths were set to $-1$ after scaling the objective function such that all coefficients were between $-1$ and $1$. The chain strengths were iteratively incremented by the geometric series with a ratio of $0.8$. This iteration was terminated after sapiUnembedAnswer collected 10% of a set number of numreads in discard mode.

Let us motivate our discussion by first experimentally showing the inconvenience caused by large penalty coefficients. The impact of $\lambda$ on the chance of observing the optimal solution from the quantum annealer is depicted in Figure 1 for two random instances. In the plots of Figure 1, the range of $\lambda$, i.e., from $0$ to the theoretical bound for $\lambda$ derived by Proposition 4, is divided into 100 steps, and the bound of Proposition 4 is depicted with a dashed line. For each value of $\lambda$ for which the quantum annealer succeeded in reaching the optimal solution, there is a '+' mark showing the number of times the optimal solution is observed among 1000 reads. Note that in both cases, the values of $\lambda$ for which the optimal solution is observed are smaller than the theoretical bound; and, the smaller $\lambda$ is, the higher is the chance of observing the optimal solution.

Our test cases belong to two categories of embedded and Chimera-native instances. In the embedded instances, the preprocessed matrix $W$ is dense, so these instances need an embedding before attempting to solve them. The native instances refer to test cases where matrices $W$ and $A$ both respect the sparsity pattern of the chip. Therefore, the problem can be solved by the trivial embedding of mapping each variable to a single qubit. We have tested the theoretical bounds of
Propositions 1 and 2 as well as the four iterative schemes of Section 4: the Newtonian method, the modified Newtonian method, the incremental method, and the hybrid method.

The Newtonian and modified Newtonian methods were terminated as soon as the first feasible solution was obtained, i.e., when $x^{k+1}$ was feasible. The incremental method was initialized with $\lambda_0 = 0$, $\delta = 1$, and $s_\lambda = 1$ (i.e., we used fixed step-size increments), and we collected five feasible solutions before termination (i.e., FeasCnt = 5). For the hybrid method, we found $\hat{\alpha}$ such that $\lambda^1$ coincides with $\lambda^1$ from the Newtonian method, i.e.,

$$\hat{\alpha} \sigma^0 = \frac{(x^0)^t W x^0}{(x^0)^t A x^0};$$

which concludes

$$\hat{\alpha}(x^0)^t A x^0 = \frac{(x^0)^t W x^0}{(x^0)^t A x^0} \Rightarrow \hat{\alpha} = \frac{(x^0)^t W x^0}{((x^0)^t A x^0)^2},$$

where $x^0 = \arg\max_{x \in \mathbb{R}^n} x^t W x$. To avoid taking extremely small step sizes, we bound our step-size scheme by 0.05, so in our experiment we took $\hat{\alpha} = \max\left(\frac{(x^0)^t W x^0}{((x^0)^t A x^0)^2}, 0.05\right)$. Similar to the incremental method, we collected five feasible points before termination, so FeasCnt = 5. In our experiment, we set $s_\lambda = 0.5$.

Tables 1 and 2 summarize the results for embedded and native instances, respectively. In these tables, $n$ is the dimension of the problem. For embedded instances, opt is the optimal objective value of the problem found using Gurobi Optimizer 5.6 [7]. For native instances, the optimal objective value is not available due to the large size of these instances. Therefore, best is the best objective value observed during our experiment. The results for theoretical bounds attained in Propositions 1 and 2 are presented in columns prop1 and prop2, respectively. new, m-new, incr, and hyb refer to the Newtonian, modified Newtonian, incremental, and hybrid methods, respectively. For each method, the best objective value obj and the corresponding Lagrangian multiplier $\lambda$ are reported. In prop2, $\lambda$ refers to the range of the entries in the matrix of Lagrangian multipliers. For each instance, an asterisk indicates the best solution achieved.

The tables show that the iterative methods outperformed the theoretical bounds for this problem, especially on general instances. The hybrid method, in particular, performed the best and could achieve the optimal solution in most of our general test cases. Table 2 shows how the theoretical bound derived in Proposition 2, the matrix form of the penalty value, is an improvement over the bound in Proposition 1, the single penalty value. While the latter could never reach the best observed solution, the matrix form of the penalty values could return the best solution in some instances. Another important observation from the results from Table 2 is that the modified Newtonian method overcomes the overshooting problem of the step-size schedule in the Newtonian method. Since the size (and hence the objective value) of the native instances is large, and $x^t A x$ gets close to 0 as the algorithms proceed, the ratio $x^t W x$ can jump to large values. While this happens for the Newtonian method, the modified variant of this method avoids this problem because the best objective value observed for a feasible solution $x_f$ is also improved, so $x^t W x - x_f W x_f$ stays minimal.
In [14], the authors propose an adaptation of the outer approximation method for solving the Lagrangian dual of a CBQP problem. This method assumes the existence of a UBQP oracle that can either find the optimal solution of the UBQP problem or a bound for its value with some certainty. In practice, this means a quantum annealing device that has an error correction mechanism, or error analysis processes, attached (cf. [14, §8]).

In comparison to the subgradient descent method of this paper, the outer approximation method has a faster convergence to the solution of the Lagrangian dual problem. However, when using noisy quantum annealers, the subgradient descent method is more successful since it can avoid the need for optimization of the Lagrangian relaxation for large Lagrange multipliers. In the method in [14], a box constraint on the Lagrange multipliers is imposed in order to keep the multipliers relatively small. However, the upper bound of the Lagrange multipliers cannot be of the same order of magnitude as the coefficients of the primal problem because in that case the Lagrangian dual bound obtained is not tight enough.
Table 2. Computation experiment for Chimera-native instances of $(\text{GQSS})$.

| n   | best | $\lambda$ | $\lambda$ | $\lambda$ | $\lambda$ | incr | $\lambda$ | hyb |
|-----|------|-----------|-----------|-----------|-----------|------|-----------|-----|
| 1100 | 2784 | 20 2726   | 1-20 2784*| 69.7      | 2582      | 13.3 | 2740      | 10 2778 |
| 1100 | 2896 | 19 2838   | 1-19 2866 | 81.5      | 2650      | 10   | 2878*     | 7   2896 |
| 1100 | 2818 | 19 2748   | 1-19 2814 | 124.8     | 2546      | 9    | 2814      | 9   2818 |
| 1100 | 2748 | 20 2666   | 1-20 2748*| 54.8      | 2610      | 8.2  | 2746      | 8   2744 |
| 1100 | 2890 | 22 2782   | 1-22 2890*| 102.6     | 2658      | 11   | 2802*     | 9   2884 |
| 1100 | 2680 | 19 2609   | 1-19 2666 | 139.9     | 2458      | 8.6  | 2666      | 8   2670 |
| 1100 | 2842 | 18 2758   | 1-18 2838 | 108.2     | 2672      | 8.1  | 2834      | 8   2840 |
| 1100 | 3054 | 21 2954   | 1-21 3050 | 78.7      | 2824      | 15   | 2990      | 10  3038 |
| 1100 | 3002 | 23 2892   | 1-23 2998 | 173.4     | 2768      | 9.7  | 2986      | 8   3002 |
| 1100 | 2674 | 18 2596   | 1-18 2660 | 89        | 2458      | 8.5  | 2660      | 8   2662 |
| 1100 | 2894 | 21 2806   | 1-21 2876 | 126.7     | 2720      | 8.7  | 2894*     | 9   2876 |
| 1100 | 2874 | 20 2794   | 1-20 2874*| 166.7     | 2650      | 9.5  | 2862      | 9   2872 |
| 1100 | 2938 | 20 2876   | 1-20 2938*| 141       | 2714      | 10.8 | 2936      | 9   2928 |
| 1100 | 2776 | 21 2660   | 1-21 2776*| 79.5      | 2568      | 14   | 2730      | 10  2752 |
| 1100 | 2712 | 19 2636   | 1-19 2712*| 175.7     | 2540      | 10   | 2692      | 11  2700 |
| 1100 | 2876 | 19 2788   | 1-19 2868 | 485       | 2622      | 7    | 2866      | 8   2868 |
| 1100 | 2714 | 23 2588   | 1-23 2714*| 103.5     | 2510      | 11.7 | 2676      | 10  2690 |
| 1100 | 2814 | 20 2762   | 1-20 2806 | 209.7     | 2632      | 11.2 | 2802      | 11  2812 |
| 1100 | 2838 | 18 2766   | 1-18 2834 | 243.5     | 2588      | 9    | 2834      | 8   2836 |
| 1100 | 2808 | 19 2728   | 1-19 2808*| 100.9     | 2636      | 10.5 | 2790      | 10  2800 |
| 1100 | 2810 | 22 2746   | 1-22 2806 | 100.0     | 2636      | 10   | 2804      | 11  2798 |
| 1100 | 2790 | 21 2718   | 1-21 2786 | 76.4      | 2600      | 11   | 2764      | 10  2790*|
| 1100 | 2934 | 22 2834   | 1-22 2928 | 82.9      | 2736      | 8.1  | 2926      | 9   2918 |
| 1100 | 2838 | 21 2766   | 1-21 2818 | 148.3     | 2648      | 14   | 2790      | 8   2830 |
| 1100 | 2762 | 21 2690   | 1-21 2744 | 160.3     | 2532      | 9.1  | 2752      | 7   2758 |
| 1100 | 2760 | 24 2670   | 1-24 2752 | 104.3     | 2586      | 9    | 2758      | 9   2760*|
| 1100 | 2762 | 21 2664   | 1-21 2760 | 74.9      | 2636      | 10   | 2742      | 10  2736 |
| 1100 | 2702 | 20 2630   | 1-20 2692 | 73.6      | 2532      | 9.6  | 2700      | 11  2702*|
| 1100 | 2652 | 21 2546   | 1-21 2642 | 88.5      | 2446      | 9    | 2636      | 9   2644 |
| 1100 | 2902 | 20 2826   | 1-20 2902*| 86.6      | 2746      | 9    | 2894      | 9   2890 |

Both the method in this paper and that of [14] can be used in a branch-and-bound framework for solving CBPP problems where the bounding strategy is that of solving the Lagrangian dual of the CBPP problem at each node of the branch-and-bound tree. Such a branch-and-bound framework and details of the branching and bounding strategies is studied in [14].

As a final remark on the scope of application of this method, we wish to say that although the proof-of-concept experiment of this paper is only constrained by one quadratic constraint, the power of subgradient direction of descent becomes more prominent as the CBPP problem has multiple constraints.

7. Conclusion

Motivated by the work of [14], we proposed an alternative algorithm for solving the Lagrangian dual of a constrained binary programming problem using a quantum annealer. The method proposed in [14] does not suit current implementations of the quantum annealer because the values of the Lagrangian multipliers can get arbitrarily large. The method proposed in this work increases the penalty values in a more controlled fashion, demonstrating a better success rate with noisy quantum annealing devices.
The generalized quadratic stable set problem was considered and penalty methods for solving this problem were investigated. Theoretical bounds for the penalty coefficients needed for solving this problem were computed and shown to be tight. In our experiments using the D-Wave 2X, we observed that the theoretical bounds could be excessively large and affect the success rate of the method.

We proposed the subgradient descent approach for solving the Lagrangian dual of the constrained problems. In the particular case of the generalized quadratic stable set problem, the subgradient method may be viewed as a technique for finding smaller penalty coefficients that are instance dependent.

The success of subgradient descent depends on the step-size schedule for updating the Lagrange multipliers. A number of step-size schedules for the generalized stable set problem were studied. We showed that the subgradient descent method outperforms the penalty coefficient method for solving the generalized stable set problem.
References

[1] Tameem Albash, Walter Vinci, Anurag Mishra, Paul A. Warburton, and Daniel A. Lidar. Consistency tests of classical and quantum models for a quantum annealer. *Phys. Rev. A*, 91:042314, Apr 2015. URL: [http://link.aps.org/doi/10.1103/PhysRevA.91.042314](http://link.aps.org/doi/10.1103/PhysRevA.91.042314)

[2] P. I. Bunyk, E. M. Hoskinson, M. W. Johnson, E. Tolkacheva, F. Altomare, A. J. Berkley, R. Harris, J. P. Hilton, T. Lanting, A. J. Przybysz, and J. Whittaker. Architectural considerations in the design of a superconducting quantum annealing processor. *IEEE Transactions on Applied Superconductivity*, 24(4):1–10, Aug 2014. doi: [10.1109/TASC.2014.2318294](http://dx.doi.org/10.1109/TASC.2014.2318294)

[3] J. Cai, W. G. Macready, and A. Roy. A practical heuristic for finding graph minors. *arXiv preprint arXiv:1406.2741*, June 2014. [arXiv:1406.2741](http://arxiv.org/abs/1406.2741)

[4] E. Farhi, J. Goldstone, and S. Gutmann. A Numerical Study of the Performance of a Quantum Adiabatic Evolution Algorithm for Satisfiability. *arXiv preprint quant-ph/0007071*, July 2000. [arXiv:quant-ph/0007071](http://arxiv.org/abs/quant-ph/0007071)

[5] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda. A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science*, 292(5516):472–476, 2001. URL: [http://dx.doi.org/10.1126/science.1057726](http://dx.doi.org/10.1126/science.1057726)

[6] F. Furini and E. Traversi. Hybrid sdp bounding procedure. *Experimental Algorithms*, pages 248–259, 2013.

[7] Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2015. URL: [http://www.gurobi.com](http://www.gurobi.com)

[8] H. Ishikawa. Transformation of general binary mrf minimization to the first-order case. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(6):1234–1249, June 2011. doi: [10.1109/TPAMI.2010.91](http://dx.doi.org/10.1109/TPAMI.2010.91)

[9] M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, E. M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, C. J. S. Truncik, S. Uchaikin, J. Wang, B. Wilson, and G. Rose. Quantum annealing with manufactured spins. *Nature*, 473(7346):194–198, 05 2011. URL: [http://dx.doi.org/10.1038/nature10012](http://dx.doi.org/10.1038/nature10012)

[10] Helmut G. Katzgraber, Firas Hamze, Zheng Zhu, Andrew J. Ochoa, and H. Munoz-Bauza. Seeking quantum speedup through spin glasses: The good, the bad, and the ugly. *Phys. Rev. X*, 5:031026, Sep 2015. URL: [http://link.aps.org/doi/10.1103/PhysRevX.5.031026](http://link.aps.org/doi/10.1103/PhysRevX.5.031026)

[11] A. D. King and C. C. McGeoch. Algorithm engineering for a quantum annealing platform. *ArXiv e-prints*, October 2014. [arXiv:1410.2629](http://arxiv.org/abs/1410.2629)

[12] N. Krislock, J. Malick, and F. Roupin. BiqCrunch: A semidefinite branch-and-bound method for solving binary quadratic problems. [http://lipn.univ-paris13.fr/BiqCrunch/](http://lipn.univ-paris13.fr/BiqCrunch/).

[13] Duan Li and Xiaoling Sun. *Nonlinear Integer Programming*. Springer, 2006.

[14] Pooya Ronagh, Brad Woods, and Ehsan Iranmanesh. Solving constrained quadratic binary problems via quantum adiabatic evolution. [http://arxiv.org/abs/1509.05001](http://arxiv.org/abs/1509.05001), 2015.

[15] Wim van Dam, Michele Mosca, and Umesh Vazirani. How powerful is adiabatic quantum computation? In 42nd *IEEE Symposium on Foundations of Computer Science* (Las Vegas, NV, 2001), pages 279–287. IEEE Computer Soc., Los Alamitos, CA, 2001.

[16] Zheng Zhu, Andrew J. Ochoa, Stefan Schnabel, Firas Hamze, and Helmut G. Katzgraber. Best-case performance of quantum annealers on native spin-glass benchmarks: How chaos can affect success probabilities. *Phys. Rev. A*, 93:012317, Jan 2016. URL: [http://link.aps.org/doi/10.1103/PhysRevA.93.012317](http://link.aps.org/doi/10.1103/PhysRevA.93.012317)