Supplementary Material

1 THE RHOMBUS CONVERGENCE

In section 4.2 of the manuscript, we define the rhombus convergence property, Here, we provide the precise statement: Given a rhombus geometry defined by a non-singular matrix $\mathbf{A}$ and a vector $\mathbf{b}$, for a given rhombus with center $\mathbf{x}_0$ and 2^N lattice vectors $\mathbf{y}_k$ (see Figure S1), if the $\mathbf{x}^*$ point with $f(\mathbf{x}^*) = 0$ is inside of the rhombus and between the points $\mathbf{y}_k$, the point $\mathbf{y}_1$ satisfy $f(\mathbf{y}_1) < f(\mathbf{y}_k)$ for all $k \neq 1$, then the point $\mathbf{x}^*$ is inside of the sub-rhombus associated to $\mathbf{y}_1$. Where, $f(\mathbf{x}) = \|\mathbf{A} \cdot \mathbf{x} - \mathbf{b}\|$ and $\mathbf{A} \cdot \mathbf{x}^* = \mathbf{b}$.

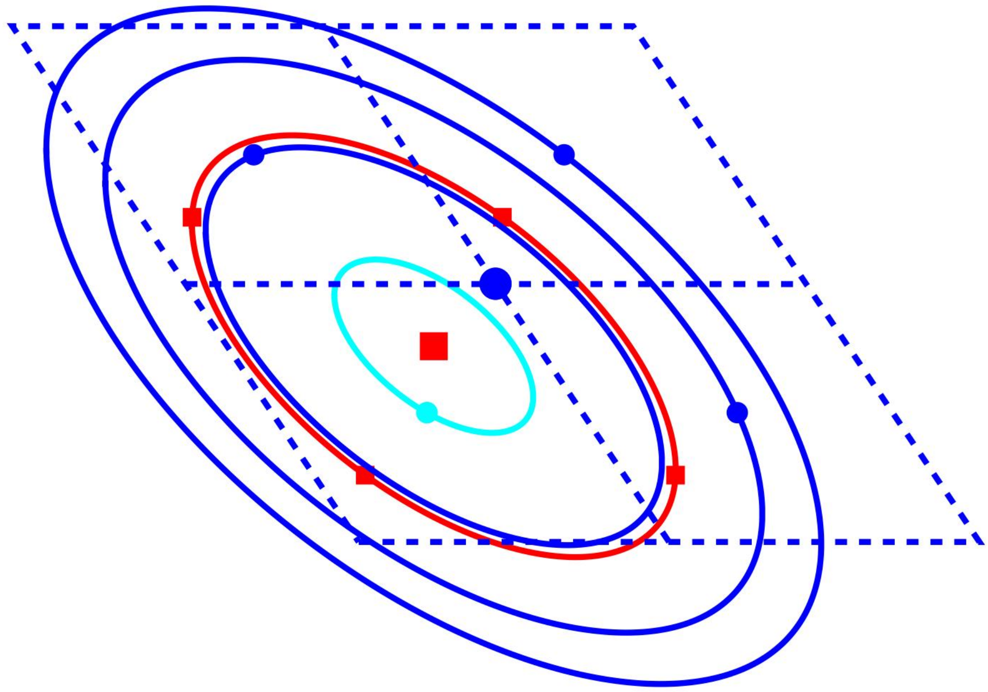

Figure S1. The figure shows the rhombus defined by the geometry of the problem; the cyan and blue points are the QUBO configurations $\mathbf{y}_k$ around the big blue point $\mathbf{x}_0$ in the center of the rhombus. There are 2^N points (in the figure, $N = 2$), and with each point, an associated sub-rhombus. The big red square $\mathbf{x}^*$ satisfy $f(\mathbf{x}^*) = 0$ and $\mathbf{x}^* = \mathbf{x}_0 + \mathbf{t}$, with $\mathbf{t}$ the difference vector. In the figure, the red tiny squares satisfy $\mathbf{x}_k = \mathbf{y}_k + \mathbf{t}$ and $f(\mathbf{x}_k) = C$ for all k. The cyan and blue ellipses express the different values of $f(\mathbf{y}_k)$. The big red square $\mathbf{x}^*$ is contained in the left-inferior sub-rhombus, and the associated cyan point $\mathbf{y}_l$ in their center satisfy $f(\mathbf{y}_l) < f(\mathbf{y}_k)$ for all $k \neq l$.

In order to understand this, we examine Figure S1. Consider the optimal configuration $\mathbf{x}^*$ (big red square in the figure), an arbitrary ellipsoid of Figure 2a centered in $\mathbf{x}^*$ and 2^N points $\mathbf{x}_k$'s defining the rhombus geometry (red ellipse and the four red squares in the figure). Consider another arbitrary point $\mathbf{x}_0$ (big blue point in the figure), which defines a new set of similar rhombus vectors $\mathbf{y}_k$'s (four cyan and blue points in the figure); for $N = 2$, these rhombus vectors define the dashed rhombus, which is divided in 4 sub-rhombus (2^N in the arbitrary case). Suppose that $\mathbf{x}^*$ is inside of this rhombus and, therefore, also is inside of a particular sub-rhombus associated with the point $\mathbf{y}_l$ (cyan point in the left inferior corner of the figure), then, between the four $\mathbf{y}_k$'s the evaluated function $f(\mathbf{y}_l)$ reach the minimal value (in the figure, the cyan point in the left inferior corner is contained in the smaller cyan ellipse). We call this property the rhombus convergence, which is proved below. The property improves the convergence since the point associated with the QUBO solution $\mathbf{x}^*$ in each iteration is also contained in the next constructed rhombus.

Consider that the point $\mathbf{x}^*$ belongs to the sub-rhombus defined by $\mathbf{y}_1$. We prove that the point $\mathbf{y}_m$ that minimize the function $f(\mathbf{y}_k)$ restricted to the QUBO vectors $\mathbf{y}_k$'s satisfy $\mathbf{y}_k = \mathbf{y}_1$. As $\mathbf{x}^*$ belongs to the sub-rhombus defined by $\mathbf{y}_1$ we can write

$$\mathbf{x}^* = \mathbf{y}_1 + \sum_{j=1}^{N} C_j \mathbf{v}_j, \quad \text{with } |C_j| \leq \frac{L}{4} \ \forall \ j, \tag{S1}$$

where L is the side length of the principal rhombus shown in Figure S1 and $\mathbf{v}_j$ is the vector that defines the rhombus geometry. All points inside of the sub-rhombus associated with $\mathbf{y}_1$ satisfied $|C_j| \leq \frac{L}{4} \ \forall \ j$, and any point outside of this sub-rhombus breaks the inequality. The point $\mathbf{x}^*$ also belongs to the principal rhombus, therefore

$$\mathbf{x}^* = \mathbf{x}_0 + \sum_{j=1}^{N} D_j \mathbf{v}_j, \quad \text{with } |D_j| \leq \frac{L}{2} \ \forall \ j. \tag{S2}$$

It is clear that $\mathbf{x}^* - \mathbf{x}_0 = \mathbf{x}_k - \mathbf{y}_k$, or

$$\mathbf{y}_k = \mathbf{x}_k - \sum_{j=1}^{N} D_j \mathbf{v}_j. \tag{S3}$$

The function $f(\mathbf{y}_k)$ restricted to the points $\mathbf{y}_k$'s can be written as

$$f(\mathbf{y}_k) = \|\mathbf{A} \cdot (\mathbf{x}^* - \mathbf{y}_k)\|^2,$$

using eq.(S2) and (S3), we obtain

$$\begin{aligned} f(\mathbf{y}_k) =& \|\mathbf{A} \cdot (\sum_{j=1}^{N} D_j \mathbf{v}_j)\|^2 + \|\mathbf{A} \cdot (\mathbf{x}_k - \mathbf{x}^*)\|^2 \\ & - 2[\mathbf{A} \cdot (\sum_{j=1}^{N} D_j \mathbf{v}_j)] \cdot [\mathbf{A} \cdot (\mathbf{x}_k - \mathbf{x}^*)]. \end{aligned} \tag{S4}$$

The two first terms of (S4) are identical for all the possible $\mathbf{y}_k$ choices. Therefore, the minimal value of $f(\mathbf{y}_m)$ is reached by the $\mathbf{x}_m$ that maximize

$$[\mathbf{A} \cdot (\sum_{j=1}^{N} D_j \mathbf{v}_j)] \cdot [\mathbf{A} \cdot (\mathbf{x}_m - \mathbf{x}^*)].$$

Consider a similar rhombus centered at $\mathbf{x}^*$ with associated rhombus points $\mathbf{x}_k$ (represented by red tiny squares in Figure S1). The similar rhombus is not shown in the figure, but imagine it centered at $\mathbf{x}^*$. $\mathbf{x}_k$

belongs to this similar rhombus centered in $\mathbf{x}^*$ and is written as

$$\mathbf{x}_k = \mathbf{x}^* + \sum_{j=1}^{N} s_j^{(k)} \frac{L}{4} \mathbf{v}_j,$$

with $s_j^{(k)} \in \{-1, 1\}$ for all j. Using the property

$$(\mathbf{A} \cdot \mathbf{v}_i) \cdot (\mathbf{A} \cdot \mathbf{v}_j) = \mathbf{v}_i \cdot \left(\mathbf{A}^T \mathbf{A}\right) \cdot \mathbf{v}_j = h_i \delta_{ij},$$

we have

$$[\mathbf{A} \cdot (\sum_{j=1}^{N} D_j \mathbf{v}_j)] \cdot [\mathbf{A} \cdot (\mathbf{x}_k - \mathbf{x}^*)] = \sum_{j=1}^{N} s_j^{(k)} \frac{L}{4} h_j D_j. \tag{S5}$$

To obtain the configuration that maximize (S5) choose $s_j^{(m)} = \mathrm{Sign}(D_j)$ (the h_i numbers are always positive). Using

$$\|\mathbf{A} \cdot (\sum_{j=1}^{N} D_j \mathbf{v}_j)\|^2 = \sum_{j=1}^{N} h_j D_j^2$$

and

$$\|\mathbf{A} \cdot (\mathbf{x}_m - \mathbf{x}^*)\|^2 = \sum_{j=1}^{N} h_j \frac{L^2}{16}$$

we obtain

$$f(\mathbf{y}_m) = \sum_{j=1}^{N} h_j \left(|D_j| - \frac{L}{4}\right)^2 = \sum_{j=1}^{N} h_j E_j^2, \tag{S6}$$

or

$$\|\mathbf{A} \cdot (\mathbf{x}^* - \mathbf{y}_m)\|^2 = \|\mathbf{A} \cdot (\sum_{j=1}^{N} E_j \mathbf{v}_j)\|^2$$

implying in

$$\mathbf{x}^* = \mathbf{y}_m + \sum_{j=1}^{N} E_j \mathbf{v}_j \tag{S7}$$

If we prove that $|E_j| \leq \frac{L}{4}\ \forall\ j$, then $E_j = C_j$ and $\mathbf{y}_m = \mathbf{y}_1$. From (S6), we have

$$E_j^2 = \left(|D_j| - \frac{L}{4}\right)^2$$

or

$$E_j^2 - \frac{L^2}{16} = |D_j|^2 - \frac{L}{2}|D_j|,$$

but

$$|D_j| \leq \frac{L}{2} \Rightarrow |D_j|^2 - |D_j|\frac{L}{2} \leq 0$$

therefore

$$E_j^2 - \frac{L^2}{16} \leq 0$$

that is equivalent to $|E_j| \leq L/4$, hence $\mathbf{y}_m = \mathbf{y}_1$.

2 ENHACEMENT OF ALGORITHM 2 TO CONSTRUCT THE VECTORS V_K

There is a result that improves the Algorithm 2 for calculating the vectors $\mathbf{v}_k$, with $k \in 1, \cdots, N$. Suppose that we use the generalized Gram-Schmidt procedure $\mathbf{u}_k \to \mathbf{v}_k$, from $k = 1$ until $k = m \leq N$. Therefore, the operator (in bra-ket notation):

$$\mathbf{G}_m \cdot \mathbf{H} = \sum_{i=1}^{m} \left(\frac{1}{\langle \mathbf{v}_i | \mathbf{H} | \mathbf{v}_i \rangle} |\mathbf{v}_i\rangle\langle \mathbf{v}_i| \right) \cdot \mathbf{H},$$

acts as the identity in the subspace generated by $\{\mathbf{u}_1, \cdots, \mathbf{u}_m\}$. In particular, when $m = N$, the operator $\mathbf{G}_N = \sum_{i=1}^{N} \left(\frac{1}{\langle \mathbf{v}_i | \mathbf{H} | \mathbf{v}_i \rangle} |\mathbf{v}_i\rangle\langle \mathbf{v}_i| \right)$ corresponds to the inverse of $\mathbf{H}$.

To prove the last assertion, let the operator

$$\mathbf{C}_m = \mathbf{G}_m \cdot \mathbf{H}$$

be applied to all the vectors $\{\mathbf{u}_1, \cdots, \mathbf{u}_m\}$. Firstly

$$\mathbf{C}_m|\mathbf{u}_1\rangle = \sum_{i=1}^{m} \frac{1}{\langle \mathbf{v}_i | \mathbf{H} | \mathbf{v}_i \rangle} |\mathbf{v}_i\rangle\langle \mathbf{v}_i|\mathbf{H}|\mathbf{u}_1\rangle$$

but $|\mathbf{u}_1\rangle = |\mathbf{v}_1\rangle$, and $\langle \mathbf{v}_i|\mathbf{H}|\mathbf{u}_1\rangle = 0$, if $i > 1$. Therefore

$$\mathbf{C}_m|\mathbf{u}_1\rangle = \frac{1}{\langle \mathbf{v}_1 | \mathbf{H} | \mathbf{v}_1 \rangle} |\mathbf{v}_1\rangle\langle \mathbf{v}_1|\mathbf{H}|\mathbf{u}_1\rangle = |\mathbf{u}_1\rangle.$$

The vectors $|\mathbf{v}_i\rangle$ are normalized in the algorithm, but it is clear that $\mathbf{G}_m$ does not depend on a particular normalization of $|\mathbf{v}_i\rangle$. Consider the unnormalized version of Eq. 8 in the principal manuscript, in bra-ket notation:

$$|\tilde{\mathbf{v}}_i\rangle = |\mathbf{u}_i\rangle + \sum_{k=1}^{i-1} \beta_{ik} |\tilde{\mathbf{v}}_k\rangle \tag{S0}$$

with $\beta_{ik} = -\langle \tilde{\mathbf{v}}_k|\mathbf{H}|\mathbf{u}_i\rangle / \langle \tilde{\mathbf{v}}_k|\mathbf{H}|\tilde{\mathbf{v}}_k\rangle$. Suppose that $j \leq m$, we have

$$\begin{aligned}
\mathbf{C}_m|\mathbf{u}_j\rangle &= \sum_{i=1}^{m} \frac{1}{\langle \tilde{\mathbf{v}}_i|\mathbf{H}|\tilde{\mathbf{v}}_i\rangle}|\tilde{\mathbf{v}}_i\rangle\langle\tilde{\mathbf{v}}_i|\mathbf{H}|\mathbf{u}_j\rangle \\
&= \sum_{i=1}^{j-1} \frac{\langle \tilde{\mathbf{v}}_i|\mathbf{H}|\mathbf{u}_j\rangle}{\langle \tilde{\mathbf{v}}_i|\mathbf{H}|\tilde{\mathbf{v}}_i\rangle}|\tilde{\mathbf{v}}_i\rangle \\
&\quad + \frac{\langle \tilde{\mathbf{v}}_j|\mathbf{H}|\mathbf{u}_j\rangle}{\langle \tilde{\mathbf{v}}_j|\mathbf{H}|\tilde{\mathbf{v}}_j\rangle}|\tilde{\mathbf{v}}_j\rangle \\
&\quad + \sum_{i=j+1}^{m} \frac{\langle \tilde{\mathbf{v}}_i|\mathbf{H}|\mathbf{u}_j\rangle}{\langle \tilde{\mathbf{v}}_i|\mathbf{H}|\tilde{\mathbf{v}}_i\rangle}|\tilde{\mathbf{v}}_i\rangle.
\end{aligned}$$

From $|\tilde{\mathbf{v}}_j\rangle = |\mathbf{u}_j\rangle + \sum_{k=1}^{j-1}\beta_{jk}|\tilde{\mathbf{v}}_k\rangle$, we have

$$\langle \tilde{\mathbf{v}}_i|\mathbf{H}|\tilde{\mathbf{v}}_j\rangle = \langle \tilde{\mathbf{v}}_i|\mathbf{H}|\mathbf{u}_j\rangle + \sum_{k=1}^{j-1}\beta_{jk}\langle \tilde{\mathbf{v}}_i|\mathbf{H}|\tilde{\mathbf{v}}_k\rangle$$

if $i > j$, the previous equation imply in $\langle \tilde{\mathbf{v}}_i|\mathbf{H}|\mathbf{u}_j\rangle = 0$. However, if $i = j$ we have $\langle \tilde{\mathbf{v}}_j|\mathbf{H}|\mathbf{u}_j\rangle = \langle \tilde{\mathbf{v}}_j|\mathbf{H}|\tilde{\mathbf{v}}_j\rangle$. Substituting in eq. (2):

$$\begin{aligned}
\mathbf{C}_m|\mathbf{u}_j\rangle &= \sum_{i=1}^{j-1} \frac{\langle \tilde{\mathbf{v}}_i|\mathbf{H}|\mathbf{u}_j\rangle}{\langle \tilde{\mathbf{v}}_i|\mathbf{H}|\tilde{\mathbf{v}}_i\rangle}|\tilde{\mathbf{v}}_i\rangle \\
&\quad + \frac{\langle \tilde{\mathbf{v}}_j|\mathbf{H}|\tilde{\mathbf{v}}_j\rangle}{\langle \tilde{\mathbf{v}}_j|\mathbf{H}|\tilde{\mathbf{v}}_j\rangle}|\tilde{\mathbf{v}}_j\rangle,
\end{aligned}$$

but $\langle \tilde{\mathbf{v}}_i|\mathbf{H}|\mathbf{u}_j\rangle / \langle \tilde{\mathbf{v}}_i|\mathbf{H}|\tilde{\mathbf{v}}_i\rangle = -\beta_{ji}$ when $i < j$. Therefore

$$\mathbf{C}_m|\mathbf{u}_j\rangle = -\sum_{i=1}^{j-1}\beta_{ji}|\tilde{\mathbf{v}}_i\rangle + |\tilde{\mathbf{v}}_j\rangle = |\mathbf{u}_j\rangle$$

so, $\mathbf{C}_m|\mathbf{u}_j\rangle = |\mathbf{u}_j\rangle$ for all $j \leq m$, but $\{\mathbf{u}_1, \cdots, \mathbf{u}_m\}$ form a basis, therefore $\mathbf{C}_m$ acts as the identity in the subspace generated by the vectors $\{\mathbf{u}_1, \cdots, \mathbf{u}_m\}$, with $m \leq N$.

Equation (2) can be rewritten as:

$$|\tilde{\mathbf{v}}_m\rangle = |\mathbf{u}_m\rangle - \sum_{i=1}^{m-1}\left(\frac{1}{\langle \mathbf{v}_i|\mathbf{H}|\mathbf{v}_i\rangle}|\mathbf{v}_i\rangle\langle\mathbf{v}_i|\right)\cdot\mathbf{H}|\mathbf{u}_m\rangle$$

or

$$|\tilde{\mathbf{v}}_m\rangle = \left(\mathbb{I} - \sum_{i=1}^{m-1}\frac{1}{\langle \mathbf{v}_i|\mathbf{H}|\mathbf{v}_i\rangle}|\mathbf{v}_i\rangle\langle\mathbf{v}_i|\cdot\mathbf{H}\right)|\mathbf{u}_m\rangle,$$

Algorithm 6

Function Ortho($\mathbf{A}$,N):
  $\mathbf{H} \leftarrow \mathbf{A}^T \cdot \mathbf{A}$
  $\mathbf{v}_1 = \mathbf{u}_1$
  $\mathbf{G} \leftarrow (1/H_{11})\, \mathbf{v}_1 \otimes \mathbf{v}_1$
  $\mathbf{w} \leftarrow -\mathbf{G} \cdot \mathbf{h}_2$
  for $k \leftarrow 2$ to N **do**:
    $w_k \leftarrow 1$
    $\mathbf{v}_k = \mathbf{w} / (\mathbf{w} \cdot \mathbf{w})^{1/2}$
    $\mathbf{O} \leftarrow \mathbf{H}.\mathbf{v}_k$
    $C \leftarrow \mathbf{v}_k \cdot \mathbf{O}$
    $\mathbf{G} \leftarrow \mathbf{G} + (1/C)\, \mathbf{v}_k \otimes \mathbf{v}_k$
    If $k < N$ **then**
      $\mathbf{w} = -\mathbf{G} \cdot \mathbf{h}_{k+1}$
    end If
  end for
end Function

Figure S2. Gram-Schmidt procedure for the calculus of the N's $\mathbf{H}$-orthogonal vectors $(\mathbf{v}_1, \cdots, \mathbf{v}_N)$. w_k corresponds to the k-th vector component of $\mathbf{w}$ and $\mathbf{h}_k = \mathbf{H} \cdot \mathbf{u}_k$.

Due to our previous result, the action of the operation within the parentheses is straightforward. We define $\mathbf{h}_m = \mathbf{H} \cdot |\mathbf{u}_m\rangle$, which correspond to the m-th row of $\mathbf{H}$. In vector notation, the vector $\tilde{\mathbf{v}}_m$ is

$$\tilde{\mathbf{v}}_m = \mathbf{u}_m - \mathbf{G}_{m-1} \cdot \mathbf{h}_m$$

with $\mathbf{G}_{m-1} \cdot \mathbf{h}_m$ a vector with dimension equal to $m - 1$.

The modification of algorithm 2 is shown in Figure S2

To conclude, it is interesting to verify that for $m < N$, the operators

$$\mathbf{E}_m = \sum_{i=1}^{m} \frac{1}{\langle \mathbf{v}_i | \mathbf{H} | \mathbf{v}_i \rangle} |\mathbf{v}_i\rangle\langle\mathbf{v}_i| \cdot \mathbf{H}$$

and

$$\mathbf{E}_m^c = \mathbb{I} - \mathbf{E}_m$$

are oblique non-Hermitian projectors.

This is

$$\mathbf{E}_m \cdot \mathbf{E}_m = \mathbf{E}_m,$$

$$\mathbf{E}_m^c \cdot \mathbf{E}_m^c = \mathbf{E}_m^c,$$

$$\mathbb{I} = \mathbf{E}_m + \mathbf{E}_m^c,$$

and

$$\mathbf{E}_m \cdot \mathbf{E}_m^c = \mathbf{E}_m^c \cdot \mathbf{E}_m = \mathbf{0}.$$

with $\mathbf{0}$ the zero matrix.

3 APPLICATION OF SIMPLE EXAMPLES FOR ALGORITHMS 3 AND 5

In Section 2 of the manuscript, we explicitly calculate the first iteration of Algorithm 1 shown in Fig. 2 for a simple linear equation system with $N = 2$. Here, we provide examples for two 4×4 ill-conditioned matrices applied to Algorithms 3 and 5 (see Fig. 5 and Fig. 8 of the manuscript), respectively. We use exact rational numerical expressions to demonstrate that the method works for arbitrary ill-conditioned matrices when the numerical error is controlled.

3.1 Example for algorithm 3

Consider the following ill-conditioned 4×4 matrix studied in Ref. **?**.

$$\mathbf{A} = \begin{pmatrix} -5046135670319638 & -3871391041510136 & -5206336348183639 & -6745986988231149 \\ -640032173419322 & 8694411469684959 & -564323984386760 & -2807912511823001 \\ -16935782447203334 & -18752427538303772 & -8188807358110413 & -14820968618548534 \\ -1069537498856711 & -14079150289610606 & 7074216604373039 & 7257960283978710 \end{pmatrix} \quad \text{(S-12)}$$

Next, we work with the exact rational expressions, but for simplicity and convenience, we express them in raw scientific notation between square brackets. This notation implies that we are working with involved exact rational expressions. For example, in this notation $\mathbf{A}$ is

$$\begin{pmatrix} -[5] \times 10^{15} & -[4] \times 10^{15} & -[5] \times 10^{15} & -[7] \times 10^{15} \\ -[6] \times 10^{14} & [9] \times 10^{15} & -[6] \times 10^{14} & -[3] \times 10^{15} \\ -[2] \times 10^{16} & -[2] \times 10^{16} & -[8] \times 10^{15} & -[1] \times 10^{16} \\ -[1] \times 10^{15} & -[1] \times 10^{16} & [7] \times 10^{15} & [7] \times 10^{15} \end{pmatrix}$$

means that we know the exact expression of $\mathbf{A}$, but we do not show it explicitly. Define $\mathbf{H} = \mathbf{A}^T \cdot \mathbf{A}$, and apply Algorithm 2 or 6 to calculate the geometry vectors. It is convenient to exclude the step $\mathbf{v}_k/(\mathbf{v}_k \cdot \mathbf{v}_k)^{1/2} \to \mathbf{v}_k$ to simplify the calculations. The geometry vectors are now not normalized, but the method still works. Defining $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4)^T$ and $\mathbf{C} = (C_1, C_2, C_3, C_4)$ with $C_k = \mathbf{v}_k \cdot (\mathbf{H} \cdot \mathbf{v}_k)$, we obtain

$$\mathbf{V} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -[1] & 1 & 0 & 0 \\ -[1] & [4] \times 10^{-1} & 1 & 0 \\ -[8] \times 10^{-1} & [2] \times 10^{-1} & -[7] \times 10^{-1} & 1 \end{pmatrix},$$

and $\mathbf{C} = \left([3] \times 10^{32}, [3] \times 10^{32}, [2] \times 10^{31}, [6] \times 10^{-97}\right)$.

Next, we define $\mathbf{b}$ to apply Algorithm 3, choosing

$$\mathbf{b} = \mathbf{A} \cdot (1, 20, 300, 4000)^T.$$

Therefore, at the final step of Algorithm 3, we must have $\mathbf{x}_0 \to (1, 20, 300, 4000)$. We explicitly show the results of the first iteration of the algorithm. Defining $\mathbf{I} = (1, 1, 1, 1)$ and $\mathbf{A_q} = \mathbf{A} \cdot \mathbf{V}^T$ equal to

$$\begin{pmatrix} -[5] \times 10^{15} & [2] \times 10^{15} & -[2] \times 10^{15} & -[6] \times 10^{-49} \\ -[6] \times 10^{14} & [9] \times 10^{15} & [4] \times 10^{15} & -[2] \times 10^{-49} \\ -[2] \times 10^{16} & -[5] \times 10^{13} & [3] \times 10^{14} & [2] \times 10^{-49} \\ -[1] \times 10^{15} & -[1] \times 10^{16} & [2] \times 10^{15} & -[2] \times 10^{-49} \end{pmatrix}$$

Using the initial guess $\mathbf{x}_0 = (0, 0, 0, 0)$ and choosing $L = 10^5$, we calculate

$$\mathbf{b_q} = \frac{1}{L}\left(\mathbf{b} + \frac{L}{2}\mathbf{A_q} \cdot \mathbf{I} - \mathbf{A} \cdot \mathbf{x}_0\right),$$

or $\mathbf{b_q} = \left(-[3] \times 10^{15}, [6] \times 10^{15}, -[9] \times 10^{15}, -[5] \times 10^{15}\right)$.

As we calculate the exact vector geometry of the problem, the QUBO matrix $\mathbf{Q}$ is diagonal. The diagonal is given by

$$\mathbf{Q} = \mathbf{C} - 2\mathbf{A_q}^T \cdot \mathbf{b_q}$$

or, $\mathbf{Q} = \left(-[2] \times 10^{31}, [1] \times 10^{31}, -[1] \times 10^{30}, -[4] \times 10^{-98}\right)$. The configuration that minimizes the previous diagonal QUBO problem is $\mathbf{q} = (1, 0, 1, 1)$. Inserting this into

$$\mathbf{x}_0 + L\mathbf{V}^T \cdot \left(\mathbf{q} - \frac{\mathbf{I}}{2}\right)$$

we obtain the new

$$\mathbf{x}_0 \approx (18712.5, -18623.5, 14709.2, 50000),$$

we redefine $L/2 \to L$, and repeat the procedure as many times as necessary calculating news $\mathbf{b_q}$ and $\mathbf{Q}$. After 50 iterations, the error difference is $|\mathbf{x}^* - \mathbf{x}_0| = 6.9 \times 10^{-11}$, with $\mathbf{x}^* = (1, 20, 300, 4000)$.

3.2 Example for algorithm 5

Now, consider the matrix $\mathbf{A}$

$$\mathbf{A} = \begin{pmatrix} -15000000000001 & 35000000000011 & -14999999999999 & 34999999999989 \\ 35000000000011 & -15000000000001 & 34999999999989 & -14999999999999 \\ -14999999999999 & 34999999999989 & -15000000000001 & 35000000000011 \\ 34999999999989 & -14999999999999 & 35000000000011 & -15000000000001 \end{pmatrix} \quad \text{(S-15)}$$

Calculating $\mathbf{H} = \mathbf{A}^T \cdot \mathbf{A}$, call the first 2×2 block diagonal block of $\mathbf{H}$ as $\mathbf{H}_1$.

$$\mathbf{H}_1 = \begin{pmatrix} [3] \times 10^{27} & -[2] \times 10^{27} \\ -[2] \times 10^{27} & [3] \times 10^{27} \end{pmatrix}$$

We can verify that the eigenvalues of $\mathbf{H}_1$ are

$$E_{\mathbf{H}_1} = ([5] \times 10^{27}, [8] \times 10^{26}).$$

The square root of the quotient of these eigenvalues is 2.5, which is an excellent low condition number for a subproblem. Choose $(a_1, a_2) = (2, 2)$ and use Algorithm 4 to determine the vectors that decompose the original 4×4 problem into two 2×2 subproblems, in the notation of algorithm 4, $\mathbf{H}_I = \mathbf{H}_1^{-1}$. Also here, it is convenient to exclude the step $\mathbf{v}_k/(\mathbf{v}_k \cdot \mathbf{v}_k)^{1/2} \to \mathbf{v}_k$ to simplify the calculations. In the end we obtain

$$\mathbf{V} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -[1] & [2] \times 10^{-25} & 1 & 0 \\ [2] \times 10^{-25} & -[1] & 0 & 1 \end{pmatrix},$$

and

$$\mathbf{H_V} = \mathbf{V} \cdot \left(\mathbf{H} \cdot \mathbf{V}^T\right) = \begin{pmatrix} \mathbf{H}_1 & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} & \mathbf{H}_2 \end{pmatrix}$$

with

$$\mathbf{H}_2 = \begin{pmatrix} [1] \times 10^3 & -[2] \times 10^2 \\ -[2] \times 10^2 & [1] \times 10^3 \end{pmatrix}.$$

As in the previous problem, define

$$\mathbf{b} = \mathbf{A} \cdot (1, 20, 300, 4000)^T,$$

$L = 10^5$, and $\mathbf{x}_0 = (0, 0, 0, 0)$. Put $R = 3$ and calculate $\mathbf{A_V} = \mathbf{A} \cdot \mathbf{V}^T$, $\mathbf{A_q} = \mathbf{A_V} \otimes \left(2^0, 2^{-1}, 2^{-2}\right)$ and

$$\mathbf{I_q}^{(1)} = \mathbf{I_q}^{(2)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \left(2^0, 2^{-1}, 2^{-2}\right),$$

$$\mathbf{H_q}^{(j)} = \mathbf{H}_j \otimes \left(2^0, 2^{-1}, 2^{-2}\right),$$

$$\mathbf{Q}_0^{(j)} = \left(\mathbf{I_q}^{(j)}\right)^T \cdot \mathbf{H_q}^{(j)},$$

with $j \in \{1, 2\}$.

With all these quantities, we can explicitly show the first iteration of Algorithm 5. First, we calculate

$$\mathbf{b_q} = \frac{1}{L}\left(\mathbf{b} + L\frac{2^3 - 1}{2^3}\mathbf{A_V} \cdot \mathbf{I} - \mathbf{A} \cdot \mathbf{x}_0\right),$$

or $\mathbf{b_q} = ([2] \times 10^{13}, [2] \times 10^{13}, [2] \times 10^{13}, [2] \times 10^{13})$. The dimension of $\mathbf{A_q}$ is 4×12, therefore $\mathbf{B} = \mathbf{A_q}^T \cdot \mathbf{b_q}$ has dimension 12. The first $R * a_1 = 6$ components of $\mathbf{B}$ are associated with subproblem 1, and the remaining $6 = R * a_2$ components of $\mathbf{B}$ are associated with subproblem 2. Call such subvectors $\mathbf{B}_1$ and $\mathbf{B}_2$. The two QUBO subproblems to solve in the first iteration are

$$\mathbf{Q}_j = \mathbf{Q}_0^{(j)} - 2 * \mathrm{Diag}^{(a_j)}\left(\mathbf{B}_j\right),$$

where $\mathrm{Diag}^{(a_j)}\left(\mathbf{B}_j\right)$ is a $6 \times 6 = Ra_j \times Ra_j$ diagonal matrix constructed with the vector $\mathbf{B}_j$ for $j \in 1, 2$. Explicitly

$$\mathbf{Q}_1 = 10^{26}\begin{pmatrix} [17] & [15] & [7.3] & -[21] & -[11] & -[5.3] \\ [15] & [1] & [3.6] & -[11] & -[5.3] & -[2.6] \\ [7.3] & [3.6] & -[1.3] & -[5.3] & -[2.6] & -[1.3] \\ -[21] & -[11] & -[5.3] & [13] & [15] & [7.3] \\ -[11] & -[5.3] & -[2.6] & [15] & -[0.85] & [3.6] \\ -[5.3] & -[2.6] & -[1.3] & [7.3] & [3.6] & -[2.2] \end{pmatrix},$$

$$\mathbf{Q}_2 = 10\begin{pmatrix} -[42] & [49] & [24] & -[18] & -[8.8] & -[4.4] \\ [49] & -[45] & [12] & -[8.8] & -[4.4] & -[2.2] \\ [24] & [12] & -[29] & -[4.4] & -[2.2] & -[1.1] \\ -[18] & -[8.8] & -[4.4] & -[50] & [49] & [24] \\ -[8.8] & -[4.4] & -[2.2] & [49] & -[49] & [12] \\ -[4.4] & -[2.2] & -[1.1] & [24] & [12] & -[31] \end{pmatrix},$$

the best QUBO solutions are respectively $\mathbf{q}_1 = \mathbf{q}_2 = (1, 0, 0, 1, 0, 0)$. We multiply each solution by the adequate factor as expressed in eq. 2 and concatenate the two solutions in one. Explicitly

$$\begin{aligned} \mathbf{q}_1 \to& (1 \times 2^0, 0 \times 2^{-1}, 0 \times 2^{-2}, 1 \times 2^0, 0 \times 2^{-1}, 0 \times 2^{-2}) \\ \mathbf{q}_1^* =& (1, 0, 0, 1, 0, 0) \\ \mathbf{q}_2 \to& (1 \times 2^0, 0 \times 2^{-1}, 0 \times 2^{-2}, 1 \times 2^0, 0 \times 2^{-1}, 0 \times 2^{-2}) \\ \mathbf{q}_2^* =& (1, 0, 0, 1, 0, 0) \\ \mathbf{q} =& (\mathbf{q}_1^*, \mathbf{q}_2^*) = (1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0) \end{aligned}$$

to obtain $\hat{\mathbf{x}}$, we add the four adjacent triplets ($R = 3$ and $N = 4$) of $\mathbf{q}$, explicitly

$$\hat{\mathbf{x}} = (\underbrace{1, 0, 0}_{\text{add}}, \underbrace{1, 0, 0}_{\text{add}}, \underbrace{1, 0, 0}_{\text{add}}, \underbrace{1, 0, 0}_{\text{add}}) = (1, 1, 1, 1).$$

Inserting this into

$$\mathbf{x}_0 + L\mathbf{V}^T \cdot \left(\hat{\mathbf{x}} - \frac{2^3 - 1}{2^3}\mathbf{I}\right)$$

we obtain the new

$$\mathbf{x}_0 \approx (6.25 \times 10^{-21}, 6.25 \times 10^{-21}, 1250, 12500),$$

we redefine $L/2 \to L$, and repeat the procedure as many times as necessary calculating news $\mathbf{b}_\mathbf{q}$, $\mathbf{Q}_1$ and $\mathbf{Q}_2$. After 120 iterations, the error difference is $|\mathbf{x}^* - \mathbf{x}_0| = 2.0 \times 10^{-32}$, with $\mathbf{x}^* = (1, 20, 300, 4000)$.