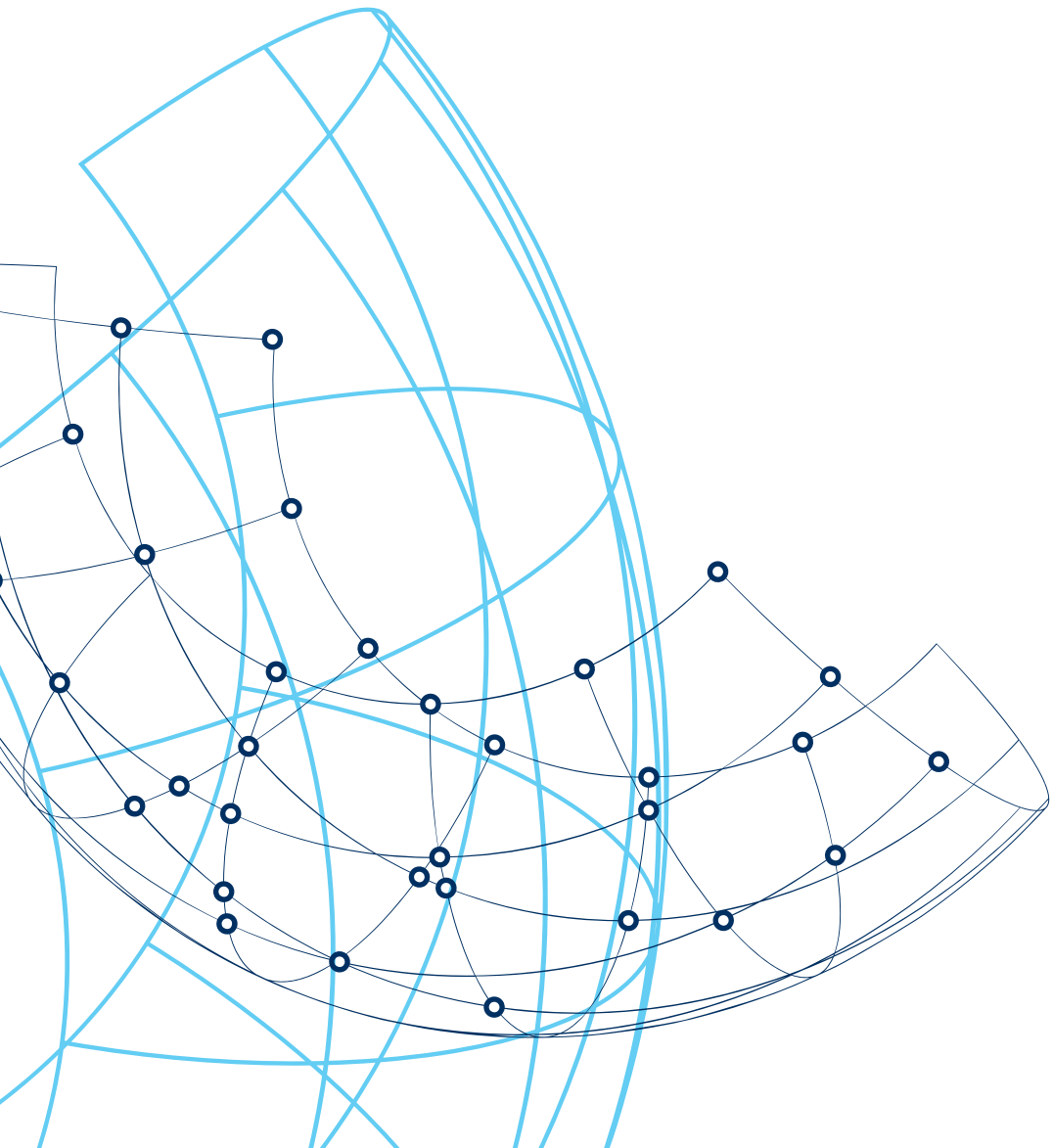
1QBit

Integer Optimization Toolbox

Minimizing Polynomials Over Integer Lattices Using Quantum Annealing



White Paper

Integer Optimization Toolbox

Minimizing Polynomials Over Integer Lattices Using Quantum Annealing

Pooya Ronagh

Abstract

In what follows, we explain how the Integer Optimization Toolbox approaches the problem of minimization of a (low degree) polynomial over an integer lattice. In theory, the method illustrated here can be used for arbitrary large polynomials with a large enough Quantum Annealing oracle.

Keywords: quantum annealing, optimization software

1 Preliminaries

We first go over some preliminaries and fix our notation and terminology.

1.1 Computational model

For us an algorithm is a deterministic Turing Machine in the usual sense. We refer the reader to any famous source on algorithms and computational complexity such as [1] and [3] for theoretical basics. By a Quantum Annealing (QA) oracle we mean an oracle machine that receives a (constructible, rational or float) degree two polynomial

$$f(x) = x^T A x + b^T x + c$$

in finitely many variables $x = \{x_i\}_{i \in I}$ and returns that minimum over all points $\{0,1\}^{\times I}$ in $O(1)$. Our algorithms are written for a Turing Machine that has a QA oracle attached to it. In other words, our goal is Turing reduction of a (non-)decision problem to the procedure of finding the minima of degree two polynomials in binary variables.

1.2 Polytopes

Let $N > 0$ and $n, m \geq 0$ be fixed integers such that $m + n = N$. By an integer lattice $\underline{L}$ on $\mathbb{R}^N$, we mean a monomorphism of abelian groups, $\mathbb{Z}^N \hookrightarrow \mathbb{R}^N$. The canonical subgroup $\mathbb{Z}^N \subseteq \mathbb{R}^N$ provides an action of $\mathbb{Z}^N$ on $\mathbb{R}^N$. As a matter of fact, we may generalize lattices to *pseudo-lattices*, i.e. equivariant mappings $\mathbb{Z}^N \to \mathbb{R}^N$. A polytope $\underline{P} \subseteq \mathbb{R}^N$ is the intersection of m half-planes given by

$$a_i(x_1, \cdots, x_N) \leq b_i \quad (i = 1, \cdots, m)\,,$$

where all a_i are linear combinations of real variables $x_1, \cdots, x_N$. We denote the polytope by the notation

$$A\,\underline{x} \leq b\,,$$

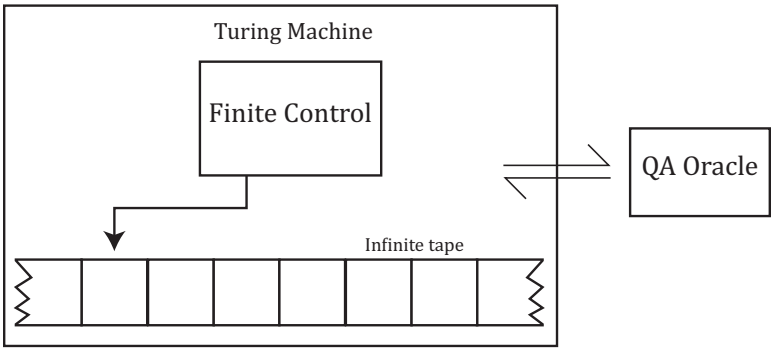


Figure 1: Schematic of a Turing machine equipped with a Quantum Annealing oracle

where A is an $m \times N$ matrix and x and b are column vectors in the obvious way. The polytope is said to be bounded if it is bounded in $\mathbb{R}^N$ as the euclidean metric space. The set $\underline{L}(\underline{P}) = \underline{P} \cap \underline{L}$ will be called the *underlying integer lattice* of $\underline{P}$, or *the set of integer points of the lattice* and the pair $(\underline{P}\,, \underline{L}(\underline{P}))$, will be referred to as a *discrete polytope*.

1.3 Linear equality constraints

Another ingredient of the problem is a set of linear equations

$$c_i(x_1, \cdots, x_N) = d_i \quad (i = 1, \cdots, \ell)\,,$$

where $c_i(x_1, \cdots, x_N) \in \mathbb{Q}[x_1, \cdots, x_N]$ and $d_i \in \mathbb{Q}$[1] and we also shorten this to $C\,\underline{x} = d$. Since C represents linear equality constraints on the space of feasible point, we may assume it is full-rank and therefore (by a permutation of variables) rewrite the linear system as

$$C\,\underline{x} = \begin{pmatrix} C_1 & C_2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = d\,,$$

where C_2 is an invertible square matrix, and by setting $n = N - m$ we have

$$x = (x_1, \cdots, x_n)^t \quad \text{and} \quad y = (x_{n+1}, \cdots, x_N)^t.$$

Consequently,

$$y = y(x) = C_2^{-1}\,(d - C_1\,x)\,. \tag{1}$$

2 Statement of the problem

Let $q\,(x_1, \cdots, x_N)$ be a polynomial in n real variables. Let $\underline{P} \subseteq \mathbb{R}^N$ be a polytope and $C\,\underline{x} = d$ be a system of linear equations as above. Note that by our assumption on rank of C, the set of solutions, S, of the linear system is a codimension-m affine space. We assume that the following axiom is satisfied.

Boundedness Axiom: $P = S \cap \underline{P}$ is a bounded polytope in S when we (non-canonically) identify S with $\mathbb{R}^n$ equipped with the euclidean metric.

Given this setup, the problem is to find an integer point $\underline{x}^* \in \underline{L}(P)$ that minimizes $q = q(\underline{x})$:

$$\underline{x}^* = \text{argmin}_{\underline{x} \in L(P)}\, q(\underline{x}) \quad \text{such that} \quad C\underline{x}^* = d\,.$$

Elimination of linear equality constraints. Equation 1 serves as a parametrization of the last m variables and lets us rewrite q as a polynomial in less number of variables in the sense of Implicit Function Theorem:

$$p(x) = q(x, y(x)).$$

[1] Here we are assuming that L is a subring of the ring of integers of $\mathbb{R}^N$. Otherwise, the rationality assumption on C and d has to change to being elements of matrix algebras over the field of fractions of L.

Note that since y is component-wise a linear combination of $x_1, \cdots, x_n$, this reduction in number of variables does not increase degree of the polynomial; $\deg p = \deg q$.

Induced integer lattice. Since S is an affine space, $L = \underline{L} \cap S$ is only a pseudo-lattice we need to identify. Given the rationality assumption on $C\underline{x} = d$, we know that $\langle 0, C_2^{-1} d \rangle$ is a rational point of S and therefore a multiple of it, ω, sits on L. Using this point in S we can find a fundamental domain for L. This is used to find the induced pseudo-lattice structure L.

Remark. By the above considerations, we can now canonically choose an affine transformation (i.e. a change of basis and a translation by ω) to respectively identify S and L with $\mathbb{R}^n$ and a lattice on it.

Remark. Given a polynomial $q(x_1, \cdots, x_N)$ as above we first reduce it to $p(x_1, \cdots, x_n)$ and afterwards forget about the linear equality constraints $C\underline{x} = d$. The induced polytope $P = \underline{P} \cap S \cong \underline{P} \cap \mathbb{R}^n$ is our new polytope and the induced integer lattice is $P(L) = P \cap L$.

3 Mini-cubes and mini-qubes

We are now reduced to solving the optimization problem of a rational polynomial $p(x) \in \mathbb{Q}[x_1, \cdots, x_n]$ on a discrete polytope $(P, L(P))$ in $\mathbb{R}^n$ without any linear equality constraints:

$$x^* = \text{argmin}_{x \in L(P)}\, p(x).$$

Let $\mathbb{B}^n = [0,1]^n$ be the standard euclidean cube in $\mathbb{R}^n$. By a mini-cube we mean an affine embedding, $\mu : \mathbb{B}^n \to \mathbb{R}^n$ of the euclidean cube to a fundamental domain of L.

Lemma 1. Given a half-plane $c^t\, x \leq d$ in $\mathbb{R}^n$ and a mini-cube $\mu : \mathbb{B}^n \to \mathbb{R}^n$, checking whether all of the image $\mu(\mathbb{B}^n)$ is contained in the half-plane can be done in $O(n^2)$.
Proof. Let μ be given by $x \mapsto Ax + b$. We have to check that $c^t Ax \leq d - c^t b$ only for points x with entries in zeros and ones.

Corollary 1. Let P be a polytope cut out in $\mathbb{R}^n$ by ℓ planes. Then the problem of checking $\mu(\mathbb{B}^n) \subseteq P$ can be solved in $O(\ell.n^2)$.

Corollary 2. Let P be a polytope cut out in $\mathbb{R}^n$ by ℓ planes. Then the problem of checking whether $\mu(\mathbb{B}^n)$ intersects the boundary of P in precisely k faces can be solved in $O(\ell.n^2)$. In particular the problem of checking whether at least one vertex of a mini-cube is inside the polytope is tractable if the mini-cubes intersects at most one face of P.

Lemma 2. The problem of deciding there exists at least one vertex of $\mu(\mathbb{B}^n)$ inside P is NP-complete.
Proof. For NP-hardness, there exists a reduction of the subset-sum problem to this. NP-completeness now follows immediately.

By a change of basis, we may assume that the fundamental domain of L is associated to vectors $v_1, \cdots, v_n$, which are scalar multiples of the standard basis. We also assume that they are respectively of norms $\delta_1, \cdots, \delta_n$. Now, given a bounded polytope $P \subseteq \mathbb{R}^n$, we can find intervals $[a_1, b_1], \cdots, [a_n, b_n]$ such that P is contained in the closed box $\mathbf{B} = \prod_i [a_i, b_i]$. The cardinality of $L(\mathbf{B})$ is bounded by $\prod_i \delta_i^{-1}(b_i - a_i)$. It is easy to see that we can cover this set as vertices of at most $\prod_i (2\delta_i)^{-1}(b_i - a_i)$ mini-cubes. Let's call the family of all these mini-cubes, $\mathfrak{Q}$. Of course we are only interested in the ones that have at least one vertex in the polytope P as illustrated in figure 2.

The upshot of the above results is that we can decide whether a cube is of the green type or the yellow type as in figure 3: However when the mini-cube is intersection P only on its co-dimension 2 skeleton then it is computationally hard to distinguish whether is has any vertex in P or not. The next figure illustrates two such mini-cubes that are hard to distinguish:

Remark. Our software finds a box that contains P first. Then iterates on all mini-cubes μ in $\mathfrak{Q}$ and decides whether $\mu(\mathbb{B}^n)$ is either contained in the polytope (of green type above) or otherwise it is cutting it only on facets (of yellow type

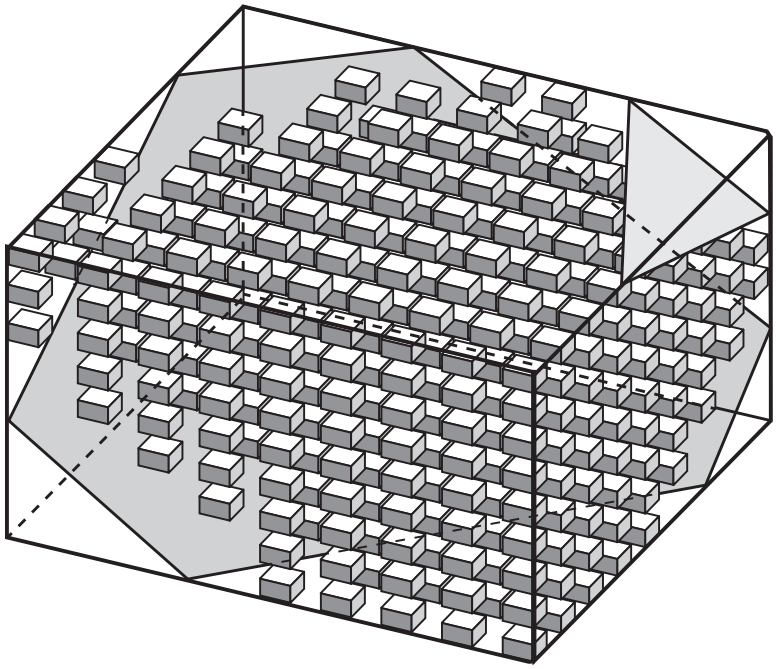


Figure 2: Tiling feasible lattice points with mini-cubes; in this figure, the domain of the minimization problem is the intersection of the circumscribing box with two half-spaces. We consider the collection of all mini-cubes that are contained in the feasibility polytope.

above) and forget about the rest. It makes a table $\mathfrak{Q}_{co-0}$ and a table $\mathfrak{Q}_{co-1}$ for these two different families.

Binary optimization problem on a mini-cube. Let $\mu : \mathbb{B}^n \to \mathbb{R}^n$ be a fixed element in $\mathfrak{Q}$. There exists an affine transformation $T : \mathbb{R}^n \to \mathbb{R}^n$ such that $T\mu$ is the identity on $\mathbb{B}^n \subseteq \mathbb{R}^n$. In particular this means that μ has an extension, $\tilde{\mu}$, to an affine isomorphism of $\mathbb{R}^n$.

Definition 1 (Mini-qube). The pullback of the polynomial $p(x_1, \cdots, x_n)$ along $\tilde{\mu}$, written as $\tilde{\mu}^*(p) = p\tilde{\mu}$ is called a mini-qube.

Remark. The reason for this nomenclature is that the problem of optimization of $p(x_1, \cdots, x_n)$ on vertices of $\mu(\mathbb{B}^n)$ is now the *binary* optimization problem on $\tilde{\mu}^*(p)$.[2]

Remark. The software now finds the associated $\tilde{\mu}^*(p)$ for each of the mini-cubes in the two lists $\mathfrak{Q}_{co-0}$ and $\mathfrak{Q}_{co-1}$.

4 Penalizing $\mathfrak{Q}_{co-1}$ mini-qubes

For each mini-qube $\mu \in \mathfrak{Q}_{co-1}$ the distance of a vertex from the fact μ intersects with, is a linear penalty function, d_μ that (by correct assignment of sign) takes positive values for vertices outside of P and takes negative values for vertices inside. In other words, this assures that the vertices inside P have less energy according to the objective function $\tilde{\mu}^*(p) + Md_\mu$ where $M > 0$ is a large enough constant.

Remark. The software keeps $\tilde{\mu}^*(p)$ as the objective function that is going to be minimized for $\mu \in \mathfrak{Q}_{co-0}$ but modifies the objective function for $\mu \in \mathfrak{Q}_{co-1}$ to $\tilde{\mu}^*(p) + Md_\mu$ where M is a fixed parameter of the program.

[2] In our earlier investigations, we were focused on the case that p was a degree 2 polynomial, hence this affine pull-back simply changed the problem to a QUBO. Something that was a 'QUBO' and 'cube' just begged to be a ...'Qube'!

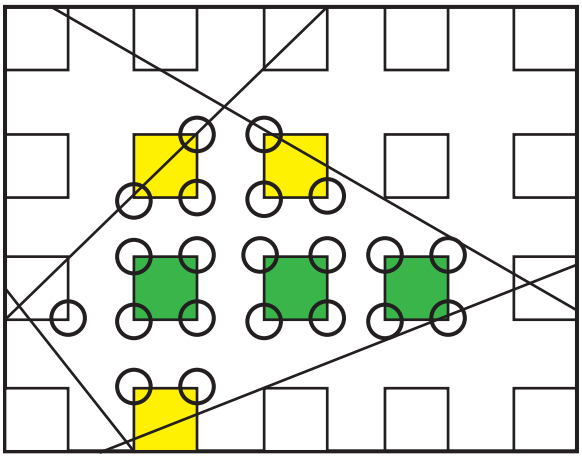


Figure 3: Placement of mini-cubes with respect to the polytope: some of the mini-cubes that contain feasible points are completely contained inside the polytope (coloured green), some others intersect the polytope only on one facet (coloured yellow).

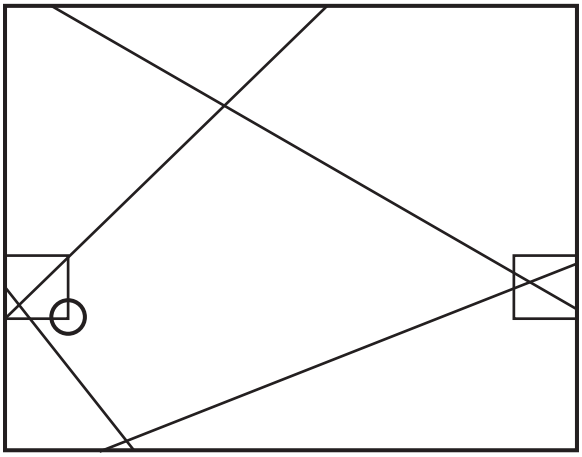


Figure 4: Negligible mini-cubes: Those mini-cubes that intersect lower dimensional faces of the polytope may also contain feasible points as vertices but we avoid taking them into account.

5 HOBO to QUBO

The next step is to use the QA oracle to minimize the associated objective function for all mini-qubes in $\mathfrak{Q}_{co-0} \cup \mathfrak{Q}_{co-1}$. Our algorithm for HOBO-QUBO is explained in [2] and is inspired by smart application of the greedy-algorithm for set-cover problem [1, §35.3].

Remark. The software changes the associated objective function from a higher order polynomial in zeros and ones to a degree two polynomial in zeros and ones with a HOBO-QUBO translation module. The result is then changed to an Ising model and the family of all these Ising models are then fed to QA oracle and the outcome of minimization of each of them is recorded. For elements in $\mathfrak{Q}_{co-1}$ the amount of penalty of the minimum vertex is reduced from it. We now have a set

$$\{(v_\mu, p(v)_\mu)) : v \text{ is the minimal vertex in } \mu \in \mathfrak{Q}_{co-0} \cup \mathfrak{Q}_{co-1}\}.$$

The final step is minimizing the above set. And we are done! The answer is then transferred back to an associated point $\underline{v}^* \in \underline{L}$ of the original lattice and the associated minimal energy $q(\underline{v}^*)$.

6 Example

In this example we show how the software handles optimization of the following degree 3 polynomial in four variables:

$$q(x_1, x_2, x_3, x_4) = 2\,x_1\,x_2\,x_3 - x_2\,x_4 + x_1\,.$$

We start with $\underline{L}$ being the standard integer lattice in $\mathbb{R}^4$ induced by the inclusion of rings $\mathbb{Z}^4 \subseteq \mathbb{R}^4$. We let $\underline{P}$ be the polytope cut out by the simplex

$$x_1 + x_2 + x_3 + x_4 \leq 12, \quad \text{and} \quad x_i \geq 0 \text{ for } i = 1, 2, 3, 4.$$

We also impose the following equality constraints

$$x_1 + x_2 - x_3 = 0, \quad \text{and} \quad x_1 + x_2 - x_4 = 1.$$

By elimination of equality constraints as in section 1.3 we have

$$\begin{pmatrix} x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} x_1 + x_2 \\ x_1 + x_2 - 1 \end{pmatrix}.$$

The induced polytope P in the (x_1, x_2)-plane is therefore given by constraints

$$1 \leq x_1 + x_2 \leq \frac{13}{3}, \quad \text{and} \quad x_i \geq 0 \quad (i = 1, 2).$$

The induced function of two variables x_1 and x_2 is

$$p(x_1, x_2) = 2\,x_1^2\,x_2 + 2\,x_1\,x_2^2 - x_1\,x_2 - x_2^2 + x_1 + x_2\,.$$

The four QUBOs that will be optimized by the QA oracle are

$$\begin{aligned}
p_{(0,0)}(x_1, x_2) &= 3\,x_1\,x_2 + x_1 \ + \text{penalty}, \\
p_{(0,2)}(x_1, x_2) &= 11\,x_1\,x_2 + 9\,x_1 - 4\,x_2 - 2\,, \\
p_{(2,0)}(x_1, x_2) &= 11\,x_1\,x_2 \ + 10\,x_2 + x_1 + 2\,, \\
p_{(2,2)}(x_1, x_2) &= 19\,x_1\,x_2 \ + 31\,x_1 + 22\,x_2 + 28 + \text{penalty}\,.
\end{aligned}$$

For these four inquiries the oracle respectively returns the binary points

$$\begin{aligned}
&(1, 1), \text{ with value } 4 \\
&(0, 1), \text{ with value } -6 \\
&(0, 0), \text{ with value } 2 \\
&(0, 0), \text{ with value } 28
\end{aligned}$$

So the final result of the software for this input instance is the point

$$(x_1, x_2, x_3, x_4) = (0, 3, 3, 2)$$

with minimum value being -6.

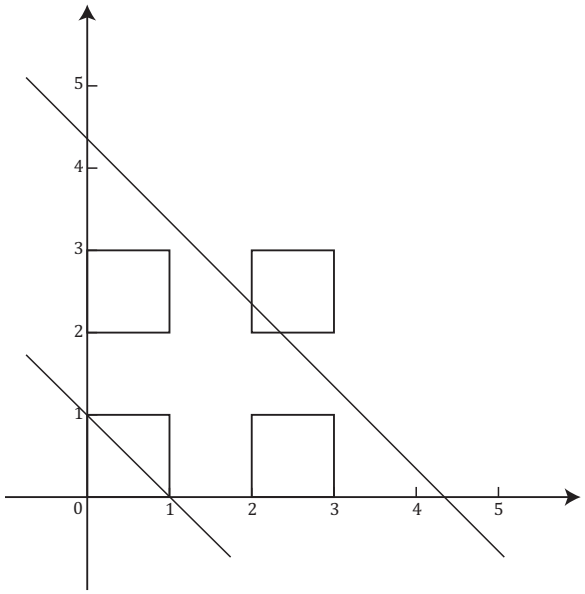


Figure 5: The induced feasible region of the example; depicted four cubes are either completely contained in the polytope or intersect it on co-dimension one faces

7 Pros and cons

Note that for a lattice of size $|L|$ the worst-case time-complexity of finding the minimum using any known algorithm is of $O(|L|)$. Generally speaking, using this software we expect to see an advantage of order of 2^n in worst-case time-complexity above, where n here is the number of variables.

Although the QA oracle works in $O(1)$, it only shows time advantage from conventional computers when the number of variables is more than a certain threshold (e.g. for 128 q-bits system, at least 10 variables).

Another important issue with this method is how it treats lattice points close to the boundaries:

1. It neglects vertices that are contained in mini-cubes that intersect the polytope in co-dimension 2 skeleton.
2. Even for the ones contained in mini-cubes that intersect the polytope strictly on faces, it involves a penalty function that does not vanish on the desired points.

Open problems and possible improvements. One important issue with this algorithm is that it runs in time-complexity of order of the number of mini-cubes tiling the circumscribing box. One desired improvement would be investigating methods of iteration over mini-cubes inside the polytope itself rather than a circumscribing box of it.

References

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 3rd edition, 2009.

[2] 1QBit group. Hobo to qubo. 2013.

[3] Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley Publishing Company, Reading, MA, 1994.

White Paper Summary - <whitepaperurl>
Demonstration Software - http://demo.1qbit.com/

Visit 1QBit.com for more information.

1QBit