Automated QUBO Generation

Automated QUBO Generation: BOX-QUBO

Black Box Optimization Using QUBO and the Cross Entropy Method

Jonas Nüßlein[1] [0000-0001-7129-1237], Christoph Roch[1], Thomas Gabor[1], Jonas Stein[1], Claudia Linnhoff-Popien[1], and Sebastian Feld[2]

[1] Institute of Computer Science, LMU Munich, Germany
jonas.nuesslein@ifi.lmu.de
[2] Faculty of Electrical Engineering, Mathematics and Computer Science, TU Delft, Netherlands

Abstract. Black-box optimization (BBO) can be used to optimize functions whose analytic form is unknown. A common approach to realising BBO is to learn a surrogate model which approximates the target black-box function which can then be solved via white-box optimization methods. In this paper, we present our approach *BOX-QUBO*, where the surrogate model is a QUBO matrix. However, unlike in previous state-of-the-art approaches, this matrix is not trained entirely by regression, but mostly by classification between "good" and "bad" solutions. This better accounts for the low capacity of the QUBO matrix, resulting in significantly better solutions overall. We tested our approach against the state-of-the-art on four domains and in all of them *BOX-QUBO* showed better results. A second contribution of this paper is the idea to also solve white-box problems, i.e. problems which could be directly formulated as QUBO, by means of black-box optimization in order to reduce the size of the QUBOs to the information-theoretic minimum. Experiments show that this significantly improves the results for MAX-k-SAT.

What we have seen so far…

1. Manual, arithmetic QUBO formulations

2. Semi-automated QUBO formulations: finding patterns using brute force

3. Algorithmic QUBO formulations: a more expressive way to manually formulate QUBOs

What we have seen so far…

1. Manual, arithmetic QUBO formulations
2. Semi-automated QUBO formulations: finding patterns using brute force
3. Algorithmic QUBO formulations: a more expressive way to manually formulate QUBOs
4. **Now: fully-automated process of creating QUBOs**

Problem setting

- Let $D = \{(x_i, y_i)\}_i$ be a set of tuples where x is a binary vector and y is a scalar (thus D describes an energy spectrum)
- Goal:

Find a (Q, c_0) such that:

$$y = x^T Q x + c_0 \qquad \forall\, (x, y) \in D$$

Simple solution:

$$Q^* = \underset{Q}{\operatorname{argmin}}\ L(Q, c_0) = \sum_{(x_i, y_i) \in D} ({x_i}^T Q x_i + c_0 - y_i)^2$$

Problem

- QUBO matrices have low capacities (amount of information they can store):

$$\frac{n(n+1)}{2} \text{ trainable parameters for a } (n \times n) \text{ matrix}$$

- The larger D the worse the regression (larger mean squared error)

BOX-QUBO - Idea

- Simultaneous **regression** and **classification**, instead of regression alone

- Sort $D = \{(x_i,\ y_i = f(x_i))\}$ according to y_i

- Divide D into two sets H und L:
  - $\forall x \in L$: $f(x) < \tau$
  - $\forall x \in H$: $f(x) \geq \tau$

- Training for QUBO-matrix Q:
  - Regression on L
  - Classification between L and H

Effect:
- Classification places lower demands on capacity than regression
- This makes the remaining regression on L more accurate

BOX-QUBO - Idea

$$y = x^T Q x + c_0 \qquad \forall\, (x,y) \in D$$

$$\Downarrow$$

$$\left.\begin{array}{ll} y < x^T Q x + c_0 & \forall\, (x,y) \in H \\ y = x^T Q x + c_0 & \forall\, (x,y) \in L \end{array}\right\} \quad \begin{array}{l} H \cup L = D \\ H \cap L = \emptyset \end{array}$$

Example

Let z_1 and z_2 be two numbers represented using 5 qubits each $\rightarrow$ value range $[0; 31]$

$$z_1 = \sum_{i=0}^{4} 2^{4-i} * x_i \qquad z_2 = \sum_{i=5}^{9} 2^{9-i} * x_i$$

0	**0**	**1**	**1**	**1**	**1**	**0**	**1**	**0**	**0**

$$\underbrace{0\ 0\ 1\ 1\ 1}_{z_1 = 7}\ \underbrace{1\ 0\ 1\ 0\ 0}_{z_2 = 20}$$

Let z_1 and z_2 be two numbers represented using 5 qubits each $\rightarrow$ value range $[0; 31]$

We want to encode
this constraint: $z_1 - z_2 = 0$

Solution: $(z_1 - z_2)^2 =$

$$\left(\sum_{i=0}^{4} 2^{4-i} * x_i - \sum_{i=5}^{9} 2^{9-i} * x_i \right)^2$$

256	256	128	64	32	-512	-256	-128	-64	-32
	64	64	32	16	-256	-128	-64	-32	-16
		16	16	8	-128	-64	-32	-16	-8
			4	4	-64	-32	-16	-8	-4
				1	-32	-16	-8	-4	-2
					256	256	128	64	32
						64	64	32	16
							16	16	8
								4	4
									1

		$z_1 = 5$					$z_2 = 5$				
		0	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**1**
	0	256	256	128	64	32	-512	-256	-128	-64	-32
	0		64	64	32	16	-256	-128	-64	-32	-16
$z_1 = 5$	**1**			16	16	8	-128	-64	-32	-16	-8
	0				4	4	-64	-32	-16	-8	-4
	1					1	-32	-16	-8	-4	-2
	0						256	256	128	64	32
	0							64	64	32	16
$z_2 = 5$	**1**								16	16	8
	0									4	4
	1										1

$$
\begin{array}{cc|cccccccccc}
 & & \multicolumn{5}{c}{\overbrace{\hspace{8em}}^{z_1 = 5}} & \multicolumn{5}{c}{\overbrace{\hspace{8em}}^{z_2 = 5}} \\
 & & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} \\
\hline
\multirow{5}{*}{$z_1 = 5$} & \mathbf{0} & 256 & 256 & 128 & 64 & 32 & -512 & -256 & -128 & -64 & -32 \\
 & \mathbf{0} & & 64 & 64 & 32 & 16 & -256 & -128 & -64 & -32 & -16 \\
 & \mathbf{1} & & & 16 & 16 & 8 & -128 & -64 & -32 & -16 & -8 \\
 & \mathbf{0} & & & & 4 & 4 & -64 & -32 & -16 & -8 & -4 \\
 & \mathbf{1} & & & & & 1 & -32 & -16 & -8 & -4 & -2 \\
\multirow{5}{*}{$z_2 = 5$} & \mathbf{0} & & & & & & 256 & 256 & 128 & 64 & 32 \\
 & \mathbf{0} & & & & & & & 64 & 64 & 32 & 16 \\
 & \mathbf{1} & & & & & & & & 16 & 16 & 8 \\
 & \mathbf{0} & & & & & & & & & 4 & 4 \\
 & \mathbf{1} & & & & & & & & & & 1
\end{array}
$$

$$E = 16 + 8 - 32 - 8 + 1 - 8 - 2 + 16 + 8 + 1 = \mathbf{0}$$

$z_1 = 3$ $\qquad$ $z_2 = 5$

Rows 1–5: $z_1 = 3$; rows 6–10: $z_2 = 5$

	0	0	0	1	1	0	0	1	0	1
0	256	256	128	64	32	-512	-256	-128	-64	-32
0		64	64	32	16	-256	-128	-64	-32	-16
0			16	16	8	-128	-64	-32	-16	-8
1				4	4	-64	-32	-16	-8	-4
1					1	-32	-16	-8	-4	-2
0						256	256	128	64	32
0							64	64	32	16
1								16	16	8
0									4	4
1										1

$$E = 4 + 4 - 16 - 4 + 1 - 8 - 2 + 16 + 8 + 1 = \mathbf{4}$$

Let z_1 and z_2 be two numbers represented using 5 qubits each $\rightarrow$ value range $[0; 31]$

We want to encode this constraint: $0 \leq z_1 - z_2 \leq 2$

Solution: $(z_1 - z_2 - a - b)^2 =$

$$\left(\sum_{i=0}^{4} 2^{4-i} * x_i - \sum_{i=5}^{9} 2^{9-i} * x_i - a - b\right)^2$$

256	256	128	64	32	-512	-256	-128	-64	-32	-32	-32
	64	64	32	16	-256	-128	-64	-32	-16	-16	-16
		16	16	8	-128	-64	-32	-16	-8	-8	-8
			4	4	-64	-32	-16	-8	-4	-4	-4
				1	-32	-16	-8	-4	-2	-2	-2
					256	256	128	64	32	32	32
						64	64	32	16	16	16
							16	16	8	8	8
								4	4	4	4
									1	2	2
										1	2
											1

$z_1 = 5$ $z_2 = 5$

	0	0	1	0	1	0	0	1	0	1	0	0
0	256	256	128	64	32	-512	-256	-128	-64	-32	-32	-32
0		64	64	32	16	-256	-128	-64	-32	-16	-16	-16
1			16	16	8	-128	-64	-32	-16	-8	-8	-8
0				4	4	-64	-32	-16	-8	-4	-4	-4
1					1	-32	-16	-8	-4	-2	-2	-2
0						256	256	128	64	32	32	32
0							64	64	32	16	16	16
1								16	16	8	8	8
0									4	4	4	4
1										1	2	2
0											1	2
0												1

$z_1 = 5$

$z_2 = 5$

$E = 0$

$z_1 = 6$ $z_2 = 5$

$z_1 = 6$

$z_2 = 5$

	0	0	1	1	0	0	0	1	0	1	0	1
0	256	256	128	64	32	-512	-256	-128	-64	-32	-32	-32
0		64	64	32	16	-256	-128	-64	-32	-16	-16	-16
1			16	16	8	-128	-64	-32	-16	-8	-8	-8
1				4	4	-64	-32	-16	-8	-4	-4	-4
0					1	-32	-16	-8	-4	-2	-2	-2
0						256	256	128	64	32	32	32
0							64	64	32	16	16	16
1								16	16	8	8	8
0									4	4	4	4
1										1	2	2
0											1	2
1												1

$E = 0$

$$z_1 = 7 \qquad\qquad z_2 = 5$$

	0	0	1	1	1	0	0	1	0	1	1	1
0	256	256	128	64	32	-512	-256	-128	-64	-32	-32	-32
0		64	64	32	16	-256	-128	-64	-32	-16	-16	-16
1			16	16	8	-128	-64	-32	-16	-8	-8	-8
1				4	4	-64	-32	-16	-8	-4	-4	-4
1					1	-32	-16	-8	-4	-2	-2	-2
0						256	256	128	64	32	32	32
0							64	64	32	16	16	16
1								16	16	8	8	8
0									4	4	4	4
1										1	2	2
1											1	2
1												1

$z_1 = 7$

$z_2 = 5$

$E = 0$

$z_1 = 8$ $\qquad$ $z_2 = 5$

$z_1 = 8$ (rows 1–5), $z_2 = 5$ (rows 6–12)

	0	1	0	0	0	0	0	1	0	1	1	1
0	256	256	128	64	32	-512	-256	-128	-64	-32	-32	-32
1		64	64	32	16	-256	-128	-64	-32	-16	-16	-16
0			16	16	8	-128	-64	-32	-16	-8	-8	-8
0				4	4	-64	-32	-16	-8	-4	-4	-4
0					1	-32	-16	-8	-4	-2	-2	-2
0						256	256	128	64	32	32	32
0							64	64	32	16	16	16
1								16	16	8	8	8
0									4	4	4	4
1										1	2	2
1											1	2
1												1

$E = 1 > 0$

$z_1 = 2$ $\qquad$ $z_2 = 29$

$z_1 = 2$ (rows 1–5), $z_2 = 29$ (rows 6–12)

	0	0	0	1	0	1	1	0	1	1	0	0
0	256	256	128	64	32	-512	-256	-128	-64	-32	-32	-32
0		64	64	32	16	-256	-128	-64	-32	-16	-16	-16
0			16	16	8	-128	-64	-32	-16	-8	-8	-8
1				4	4	-64	-32	-16	-8	-4	-4	-4
0					1	-32	-16	-8	-4	-2	-2	-2
1						256	256	128	64	32	32	32
1							64	64	32	16	16	16
0								16	16	8	8	8
1									4	4	4	4
1										1	2	2
0											1	2
0												1

$E = 729 > 0$

Now: using BOX-QUBO we can find a QUBO which encodes the constraint $0 \leq z_1 - z_2 \leq 2$ without the ancilla qubits

Constraint $0 \le z_1 - z_2 \le 2$ as QUBO:

Hand-designed Encoding

256	256	128	64	32	-512	-256	-128	-64	-32	-32	-32
	64	64	32	16	-256	-128	-64	-32	-16	-16	-16
		16	16	8	-128	-64	-32	-16	-8	-8	-8
			4	4	-64	-32	-16	-8	-4	-4	-4
				1	-32	-16	-8	-4	-2	-2	-2
					256	256	128	64	32	32	32
						64	64	32	16	16	16
							16	16	8	8	8
								4	4	4	4
									1	2	2
										1	2
											1

Encoding learned with BOX-QUBO

35	43	26	16	8	-86	-43	-26	-16	-8
	9	13	8	4	-43	-26	-13	-8	-4
		2	4	2	-26	-13	-8	-4	-2
			0	1	-16	-8	-4	-2	-1
				0	-8	-4	-2	-1	-1
					51	43	26	16	8
						17	13	8	4
							6	4	2
								2	1
									1

Constraint $0 \leq z_1 - z_2 \leq 2$ as QUBO:

Hand-designed Encoding

256	256	128	64	32	-512	-256	-128	-64	-32	-32	-32
	64	64	32	16	-256	-128	-64	-32	-16	-16	-16
		16	16	8	-128	-64	-32	-16	-8	-8	-8
			4	4	-64	-32	-16	-8	-4	-4	-4
				1	-32	-16	-8	-4	-2	-2	-2
					256	256	128	64	32	32	32
						64	64	32	16	16	16
							16	16	8	8	8
								4	4	4	4
									1	2	2
										1	2
											1

Encoding learned with BOX-QUBO

35	43	26	16	8	-86	-43	-26	-16	-8
	9	13	8	4	-43	-26	-13	-8	-4
		2	4	2	-26	-13	-8	-4	-2
			0	1	-16	-8	-4	-2	-1
				0	-8	-4	-2	-1	-1
					51	43	26	16	8
						17	13	8	4
							6	4	2
								2	1
									1

Fewer qubits and smaller values

Constraint $0 \leq z_1 - z_2 \leq 2$ as QUBO:

Hand-designed Encoding

256	256	128	64	32	-512	-256	-128	-64	-32	-32	-32
	64	64	32	16	-256	-128	-64	-32	-16	-16	-16
		16	16	8	-128	-64	-32	-16	-8	-8	-8
			4	4	-64	-32	-16	-8	-4	-4	-4
				1	-32	-16	-8	-4	-2	-2	-2
					256	256	128	64	32	32	32
						64	64	32	16	16	16
							16	16	8	8	8
								4	4	4	4
									1	2	2
										1	2
											1

Encoding learned with BOX-QUBO

35	43	26	16	8	-86	-43	-26	-16	-8
	9	13	8	4	-43	-26	-13	-8	-4
		2	4	2	-26	-13	-8	-4	-2
			0	1	-16	-8	-4	-2	-1
				0	-8	-4	-2	-1	-1
					51	43	26	16	8
						17	13	8	4
							6	4	2
								2	1
									1

Fewer qubits and smaller values

$z_1 = 5$ $z_2 = 5$

$z_1 = 5$

$z_2 = 5$

	0	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**1**
0	35	43	26	16	8	-86	-43	-26	-16	-8
0		9	13	8	4	-43	-26	-13	-8	-4
1			2	4	2	-26	-13	-8	-4	-2
0				0	1	-16	-8	-4	-2	-1
1					0	-8	-4	-2	-1	-1
0						51	43	26	16	8
0							17	13	8	4
1								6	4	2
0									2	1
1										1

$E = 0$

		$z_1 = 5$					$z_2 = 5$				
		0	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**1**
$z_1 = 5$	**0**	35	43	26	16	8	-86	-43	-26	-16	-8
	0		9	13	8	4	-43	-26	-13	-8	-4
	1			2	4	2	-26	-13	-8	-4	-2
	0				0	1	-16	-8	-4	-2	-1
	1					0	-8	-4	-2	-1	-1
$z_2 = 5$	**0**						51	43	26	16	8
	0							17	13	8	4
	1								6	4	2
	0									2	1
	1										1

$$E = 2 + 2 - 8 - 2 + 0 - 2 - 1 + 6 + 2 + 1 = \mathbf{0}$$

$z_1 = 6$ $z_2 = 5$

$z_1 = 6$

$z_2 = 5$

	0	**0**	**1**	**1**	**0**	**0**	**0**	**1**	**0**	**1**
0	35	43	26	16	8	-86	-43	-26	-16	-8
0		9	13	8	4	-43	-26	-13	-8	-4
1			2	4	2	-26	-13	-8	-4	-2
1				0	1	-16	-8	-4	-2	-1
0					0	-8	-4	-2	-1	-1
0						51	43	26	16	8
0							17	13	8	4
1								6	4	2
0									2	1
1										1

$$E = 2 + 4 - 8 - 2 - 4 - 1 + 6 + 2 + 1 = \mathbf{0}$$

		$z_1 = 7$					$z_2 = 5$				
		0	**0**	**1**	**1**	**1**	**0**	**0**	**1**	**0**	**1**
$z_1 = 7$	**0**	35	43	26	16	8	-86	-43	-26	-16	-8
	0		9	13	8	4	-43	-26	-13	-8	-4
	1			2	4	2	-26	-13	-8	-4	-2
	1				0	1	-16	-8	-4	-2	-1
	1					0	-8	-4	-2	-1	-1
$z_2 = 5$	**0**						51	43	26	16	8
	0							17	13	8	4
	1								6	4	2
	0									2	1
	1										1

$E = 0$

$z_1 = 8$ $z_2 = 5$

	0	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**
0	35	43	26	16	8	-86	-43	-26	-16	-8
1		9	13	8	4	-43	-26	-13	-8	-4
0			2	4	2	-26	-13	-8	-4	-2
0				0	1	-16	-8	-4	-2	-1
0					0	-8	-4	-2	-1	-1
0						51	43	26	16	8
0							17	13	8	4
1								6	4	2
0									2	1
1										1

$z_1 = 8$

$z_2 = 5$

$E = 1 > 0$

		$z_1 = 3$					$z_2 = 1$				
		0	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**1**
$z_1 = 3$	**0**	35	43	26	16	8	-86	-43	-26	-16	-8
	0		9	13	8	4	-43	-26	-13	-8	-4
	0			2	4	2	-26	-13	-8	-4	-2
	1				0	1	-16	-8	-4	-2	-1
	1					0	-8	-4	-2	-1	-1
$z_2 = 1$	**0**						51	43	26	16	8
	0							17	13	8	4
	0								6	4	2
	0									2	1
	1										1

$E = 0$

$z_1 = 17$ $z_2 = 1$

$z_1 = 17$

$z_2 = 1$

	1	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**1**
1	35	43	26	16	8	-86	-43	-26	-16	-8
0		9	13	8	4	-43	-26	-13	-8	-4
0			2	4	2	-26	-13	-8	-4	-2
0				0	1	-16	-8	-4	-2	-1
1					0	-8	-4	-2	-1	-1
0						51	43	26	16	8
0							17	13	8	4
0								6	4	2
0									2	1
1										1

$E = 35 > 0$

- The QUBO found with BOX-QUBO is correct for all states

Modeling of inequalities possible without ancillae

- It saves 2 qubits here $(12 \rightarrow 10)$ compared to the hand-designed QUBO

- Trained QUBO is hard to interpret for humans
  (analogy to neural networks)

Let's get into the code ...

Black-Box Optimization using QUBO

White Box vs. Black Box Optimization

Optimization: $x^* = \underset{x}{\operatorname{argmin}} f(x)$

White Box Optimization:

Mathematical calculation of $f(x)$ is completely known. Example:

$f(x) = (x-1)^2$

$f(x) = -(\#\text{ satisfied clauses in: } (x_1 \vee x_2) \wedge (-x_1 \vee x_2))$

Black Box Optimization:

- Mathematical calculation of $f(x)$ is not known.
- But we can query $f(x)$ like an oracle

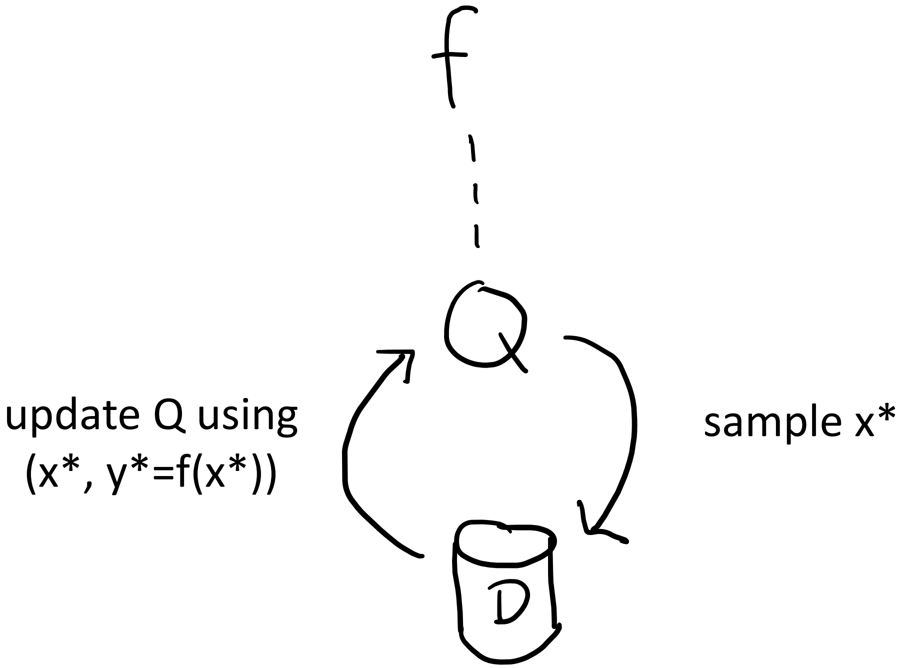
Black-Box Optimization using QUBO

- Let $f(x)$ be an unknown function
- Goal:

$$x^* = \underset{x}{argmin}\, f(x)$$

- Problem: we can‘t formulate f directly as a QUBO (since we don‘t know f)

- Solution:
  - Learn a surrogate model Q (QUBO matrix)
  - Iterate between update and optimization of this surrogate model



Advantages of BBO for QUBO

1. Easy application: just define the oracle (no knowledge needed about how to encode an objective as a QUBO)

2. Smaller QUBO sizes. E.g.: in our 3-SAT example (V=15, C=100)

    - SOTA: 115 logical qubits (for each variable + for each clause)
    - BBO: 15 logical qubits (for each variable)

Let's get into the code ...