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.

Keywords: QUBO · Black-Box · Quantum Annealing · SAT · Ising

1 Introduction

The goal of black-box optimization is to minimize a function $E(x)$, where this function is not known analytically. Like an oracle, this function can only be queried for a given x: $y = E(x)$. A commonly used solution for this problem is to create a surrogate model $E'(x)$. This surrogate model is trained to provide the same outputs as $E(x)$. Since, unlike $E(x)$, $E'(x)$ is a white-box model, it is easier to search for the best solution $x^* = argmin_x\ E'(x)$. Black-box optimization then iterates between searching the best solution x^* for the surrogate model $E'(x)$, asking the oracle for the actual value $y = E(x^*)$ and re-training the surrogate model E' to output y for the input x^* using the mean squared error as the loss function: $loss = (E'(x^*) - y)^2$. We use the terms "oracle" and "black-box function" interchangeably in this paper.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_4

Following the tradition of machine learning, we use the term 'capacity' to refer to the amount of information a model can memorize. In black-box optimization, there is a trade-off between the capacity of the surrogate model (the higher the capacity of surrogate model $E'(x)$, the better it can approximate the actual values y) and the difficulty of the optimization (the higher the capacity of surrogate model $E'(x)$, the more difficult the optimization $x^* = argmin_x\ E'(x)$). A frequent choice for the model $E'(x)$ recently fell [2,13] on the Quadratic Unconstrained Binary Optimization (QUBO) matrix Q [4], which seems like a good trade-off for the capacity.

2 Background

2.1 MAX-SAT

Satisfiability (SAT) is a canonical NP-complete problem [11]. Let $V = \{v_1, ..., v_n\}$ be a set of Boolean variables and let f be a Boolean formula represented in conjunctive normal form: $f = \bigwedge_{i=1}^{|f|} \bigvee_{l \in C_i} l$; with l being a literal and C_i being the i-th clause. The task is to find an assignment $\underline{V}$ for V such that $f(\underline{V}) = 1$. MAX-SAT is a variant of SAT, where not all clauses have to be satisfied, but only as many as possible. In k-SAT, each clause consists of exactly k literals.

2.2 Feedback Vertex Set (FVS)

Feedback Vertex Set is an NP-complete problem [11]. Let $G = (V, E)$ be a directed graph with vertex set V and edge set E which contains cycles. A cycle is a path (following the edges of E) in the graph starting from any vertex v_S such that one ends up back at vertex v_S. The path can contain any number of edges. The task in Feedback Vertex Set is to select the smallest subset $V' \subset V$ such that graph $G = (V \backslash V', E')$ is cycle-free, where $E' = \{(v_i, v_j) \in E : v_i \notin V' \wedge v_j \notin V'\}$.

2.3 MaxClique

Maximum Clique is another NP-complete problem [11]. Let $G = (V, E)$ be an undirected graph with vertex set V and edge set E. A clique in graph G is a subset $V' \subset V$ of the vertices, so all pairs of vertices from V' are connected with an edge. That is, $\forall (v_i, v_j) \in V' \times V' : (v_i, v_j) \in E$. The task in MaxClique is to find the largest clique of the graph. The size of a clique is the cardinality of the set V': $|V'|$.

2.4 Quadratic Unconstrained Binary Optimization (QUBO)

Let Q be an upper-triangular (n×n)-matrix Q and x be a binary vector of length n. The task of Quadratic Unconstrained Binary Optimization [16] is to solve the following optimization problem: $x^* = argmin_x(x^T Q x)$. QUBO is NP-hard [3] and has been of particular interest recently as special machines, such as the

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_4

Quantum Annealer [4, 8] or the Digital Annealer [1], have been developed to solve these problems. Therefore, in order to solve other problems with these special machines, they must be formulated as QUBO. Numerous problems such as MAX-k-SAT [5,6,19,20], MaxClique [17,18], and FVS [18] have already been formulated as QUBO.

2.5 Cross-Entropy Method

The cross-entropy method [7] is a general iterative optimization method to optimize an oracle $g(x)$ using a parameterized probability distribution $f(x;v)$. The method iterates between sampling p solutions: $X \sim f(x;v)$, sorting them by their values $g(X)$ in increasing order and adjusting the parameters v such that better solutions x get a higher probability $f(x;v)$. This makes it more likely to sample better solutions in the next iteration, starting again with sampling from the probability distribution: $X \sim f(x;v)$, with the new values v.

3 Related Work

The main Related Work to our approach is Factorization Machine Quantum Annealing (*FMQA*) [10,13]. *FMQA*, like our approach, is based on an iteration of three phases: 1) sample from the surrogate model (a QUBO matrix); 2) retrieve the value from the oracle (either via experiments or simulation); 3) update the QUBO matrix. This is in fact a cross-entropy approach with f being the surrogate model (the QUBO matrix). The main difference to our approach is that *FMQA* solely uses regression to train the model, while we use simultaneous regression and classification. The authors subsequently use *FMQA* in their paper to design metamaterials with special properties. A similar approach to *FMQA* is Bayesian Optimization of Combinatorial Structures (BOCS) [2] which also uses a quadratic model as a surrogate model. BOCS additionally uses a sparse prior to facilitating optimization. However, BOCS also only uses regression to optimize the model. In the original paper [2], BOCS was solved using only non-quantum methods (including Simulated Annealing [12]), Koshikawa et al. [14], however, also tested BOCS for optimization with a quantum annealer. In [15] Koshikawa et al. used BOCS for a vehicle design problem and found that it performed slightly better than a random search. In our experiments, we use the white box QUBO formulations for MAX-k-SAT [6], FVS [18] and MaxClique [18] as baselines. In [9], surrogate QUBO solvers were trained to simplify the optimization of hyperparameters arising in the relaxation of constrained optimization problems. Roch et al.

4 Black Box Optimization with Cross Entropy and QUBO (BOX-QUBO)

Main idea: The QUBO matrix is a surrogate model with relatively low capacity since it has only $n(n+1)/2$ trainable parameters for a matrix of size $(n \times$

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_4

n). Previous approaches to black-box optimization with QUBO always attempt full regression on all training data. However, the loss becomes larger the more training data is available for the same size of the QUBO matrix due to the limited number of trainable parameters. The main idea behind our approach is to perform a regression only on a small part of the training data and classification on the remaining data since classification requires less capacity of the model than regression. For this purpose, the training vectors x are sorted according to their solution quality (also called energy) $E(x)$ and divided into two sets: the set of vectors with higher energies H and the set of vectors with lower energies L. There is now a threshold τ such that:

$$\forall x \in H : E(x) \geq \tau$$
$$\forall x \in L : E(x) < \tau$$

Note that the goal is to find vector x^* with minimum energy: $\forall x : E(x) \geq E(x^*)$. The goal of the classification is to classify whether a vector x is in set H or L. The regression is performed exclusively on L. The size of L is determined by a hyperparameter k. For example, using $k = 0.03$, L contains 3% of all data: $|L| = 0.03 \cdot |D|$, thus a more precise regression is possible. *BOX-QUBO* requires as input the oracle (black-box function) $E(x)$ and an initial training set D. The output will be the vector x^* with minimum energy $y^* = E(x^*)$. First, the QUBO matrix Q is randomly initialized. After that, the following three phases alternate:

1. Search the p best solution vectors X^* for the current QUBO matrix Q (e.g. using simulated or quantum annealing)
2. Query the oracle for actual values $E(X^*)$
3. Append $(X^*, E(X^*))$ to the training set D and retrain Q on D

The training of Q using the training set D proceeds as follows. First, D is sorted by the energies $E(x)$ and then subdivided into the two non-overlapping sets H and L. After splitting, Q is trained for *nCycles* iterations. In each cycle, the *temporary* training set T is created dynamically: all vectors $x \in L$ are always part of T, vectors $x \in H$, however, are only part of T if their current prediction $y_{predict} = x^T Q x$ is smaller than τ and thus would violate the classification. The loss function on this temporary training set is now the mean squared error between $y_{predict} = x^T Q x$ and y_{target}. y_{target}, for a vector x, is equal to $E(x)$ if $x \in L$ and it is equal to τ if $x \in H$:

$$Q^* = \underset{Q}{\operatorname{argmin}} \left(\sum_{x \in T} (y_{predict} - y_{target}) \right)^2$$

We optimize Q using gradient descent with respect to this loss function. The complete algorithm is listed in Algorithm 1.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_4

Algorithm 1: BOX-QUBO

Data: Set of initial training data $D = \{x, E(x)\}$; Oracle $E(.)$
Result: Best solution (x^*, y^*) of the oracle

$Q \leftarrow$ init QUBO matrix
$Q \leftarrow train(Q, D)$

for *trainingLength* **do**
  $X \leftarrow$ sample best k solutions from Q (e.g. using quantum annealing)
  $Y = E(X)$ `// query oracle` $\forall x \in X$
  $D.add(X, Y)$
  $Q \leftarrow train(Q, D)$
end

return $(x^*, y^*) \in D$ `// Return best` y `and corresponding` x

Function `train(`Q, D`):`
  $H, L, \tau \leftarrow$ sort and split D `// see chapter` ***Splitting*** H ***and*** L
  for *nCycles* **do**
    $T, Y_{target} = L, E(L)$ `// ensure regression` $\forall x \in L$
    for $x \in H$ **do**
      $y_{predict} = x^T Q x$ `// ensure classification` $\forall x \in H$
      if $y_{predict} < \tau$ **then**
        $T \leftarrow x$
        $Y_{target} \leftarrow \tau$
      end
    end
    $L(x, y_{target}) = (x^T Q x - y_{target})^2$ `// the loss function`
    optimize Q via gradient descent w.r.t. $\nabla_Q L(T, Y_{target})$
  end
  return Q

Splitting H and L: We divide the training data D into the 'good' solutions L and the 'bad' solutions H based on their energies $E(x)$. For this, we use a hyperparameter k (a percentage): *BOX-QUBO(k%)*. k% of the training data are inserted into the set L and the remaining data are inserted into H. In addition, we use the notation *BOX-QUBO(k%\invalids)* to state that all invalid solutions are inserted into the set H. To solve a problem (e.g., MaxClique) with QUBO, the problem must be formulated as QUBO and then the QUBO solution x must be translated back into a problem solution (for example, the set of vertices forming the clique). We call a QUBO solution x 'invalid' if it does not correspond to a valid problem solution (for example, if the QUBO solution x corresponds to a set of vertices V' which is not a clique).

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_4

5 Experiments

In the experiments, we tested whether the approaches *BOX-QUBO* and *FMQA* are able to solve the domains MAX-k-SAT, FVS and MaxClique when these domains are solely accessible as black-box functions. That is, the concrete Boolean formula (for MAX-k-SAT) or the concrete graph (for FVS and MaxClique) are unknown. The oracles return an indirectly proportional value to the solution quality and the value 1 if the QUBO solution x does not translate into a valid problem solution. The goal of *BOX-QUBO* and *FMQA* was then to minimize the black-box functions (oracles). We have chosen as oracles:

$$SAT_{oracle}(x) = -(\#\ satisfied\ clauses\ with\ variable\ assignment\ x)$$
$$FVS_{oracle}(x) = if\ G \backslash x\ still\ has\ cycles\ then\ 1\ else\ (\Sigma_i x_i - |V|)$$
$$MaxClique_{oracle}(x) = if\ x\ is\ a\ valid\ clique\ then\ (-\Sigma_i x_i)\ else\ 1$$

where G is a graph. In MAX-k-SAT, V is the number of boolean variables and C is the number of clauses. In FVS and MaxClique, V is the number of vertices of the graph and E is the number of edges. $(\Sigma_i x_i)$ calculates the number of 1s in the vector x, which represents in MaxClique the size of the clique and in FVS the number of deleted vertices. For each of the four domains, 15 random instances (with different seeds) were created. That is, for Max-4-SAT 15 random formulas were created and for FVS and MaxClique 15 different graphs were generated each. Then, the *BOX-QUBO* and *FMQA* approaches were applied to each of the 15 instances (i.e. on the corresponding oracle functions). Both algorithms used the same initial training set. The best solution from this initial set served as the baseline *base*. In addition, specialized white-box methods were used to determine the true optimal solution to each instance, which we refer to as *optimum*. To better compare performance between domains, we normalized the best solution found for each of the *BOX-QUBO* and *FMQA* approaches, using the formula: $f(y^*) = (y^* - base)/(optimum - base)$. This gives $f(y^*) = 1$ if $y^* = optimum$ and $f(y^*) = 0$ if $y^* = base$, where y^* is the value of the best solution (x^*) found so far. The results are shown in **Figure 1**.

The first result is that black-box optimization (both *BOX-QUBO* and *FMQA*) found significantly better solutions than Choi's white-box solution for Max-4-*SAT*. Using the black-box optimization, we have reduced the size of the QUBO matrix to the information-theoretic minimum. For example, in the first experiment, we considered formulas with $V = 30$ variables and $C = 400$ clauses. Here, Choi's QUBO matrix had size $n = 4 \cdot 400 = 1600$, while the QUBO matrices for *BOX-QUBO* and *FMQA* corresponded to the information-theoretic minimum ($n = V = 30$). However, solving with black-box methods is not always better than solving with white-box methods, as can be seen in the results for *FVS*. The second result is that *BOX-QUBO* was always more successful than *FMQA*. In *FVS*, for example, *BOX-QUBO(10%\invalids)* reached a mean performance of roughly 0.87 after 15 iterations while *FMQA* only reached roughly 0.03. A similar picture emerged for MaxClique, with all *BOX-QUBO* variants reaching the

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_4

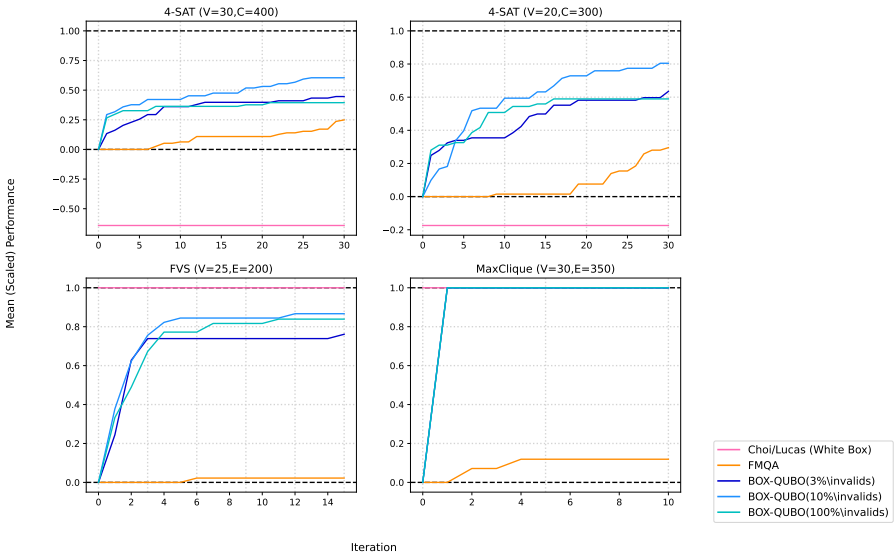

Fig. 1. Normalized performance of *BOX-QUBO*, *FMQA* [13] and the white-box solutions of Choi [6] and Lucas [18]. The x-axes describe the iteration number and the y-axes describe the (normalized) mean performance over 15 instances. In MaxClique, the graphs of *BOX-QUBO(100%\invalids)*, *BOX-QUBO(10%\invalids)* and *BOX-QUBO(3%\invalids)* overlap.

optimum already after the first iteration. The code for *BOX-QUBO* is available on GitHub [`https://github.com/JonasNuesslein/BOX-QUBO`]

6 Conclusion and Future Work

In this paper, we studied black-box optimization using QUBOs as surrogate models. We presented the *BOX-QUBO* approach, which is characterized by the idea to use a simultaneous classification and regression, instead of regression alone. We have tested our approach on the MAX-k-SAT, FVS, and $MaxClique$ domains, and the experiments showed that *BOX-QUBO* consistently outperformed *FMQA*. Besides introducing *BOX-QUBO*, we also presented the idea of solving white-box problems using black-box optimization to reduce the size of the QUBO matrices to the information-theoretic minimum. The experiments on MAX-4-SAT showed that the solutions found using black-box optimization were significantly better than those found using the white-box QUBO formulation.

Acknowledgements: This publication was created as part of the Q-Grid project (13N16179) under the "quantum technologies - from basic research to market" funding program, supported by the German Federal Ministry of Education and Research.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_4

References

1. Aramon, M., Rosenberg, G., Valiante, E., Miyazawa, T., Tamura, H., Katzgraber, H.G.: Physics-inspired optimization for quadratic unconstrained problems using a digital annealer. Frontiers in Physics **7**, 48 (2019)
2. Baptista, R., Poloczek, M.: Bayesian optimization of combinatorial structures. In: International Conference on Machine Learning. pp. 462–471. PMLR (2018)
3. Barahona, F.: On the computational complexity of ising spin glass models. Journal of Physics A: Mathematical and General **15**(10), 3241 (1982)
4. Boothby, K., Bunyk, P., Raymond, J., Roy, A.: Technical report: Next-generation topology of d-wave quantum processors (2019)
5. Chancellor, N., Zohren, S., Warburton, P.A., Benjamin, S.C., Roberts, S.: A direct mapping of max k-SAT and high order parity checks to a chimera graph (2016)
6. Choi, V.: Adiabatic quantum algorithms for the np-complete maximum-weight independent set, exact cover and 3sat problems. arXiv preprint arXiv:1004.2226 (2010)
7. De Boer, P.T., Kroese, D.P., Mannor, S., Rubinstein, R.Y.: A tutorial on the cross-entropy method. Annals of operations research **134**(1), 19–67 (2005)
8. Denchev, V.S., Boixo, S., Isakov, S.V., Ding, N., Babbush, R., Smelyanskiy, V., Martinis, J., Neven, H.: What is the computational value of finite-range tunneling? Physical Review X **6**(3), 031015 (2016)
9. Huang, T., Goh, S.T., Gopalakrishnan, S., Luo, T., Li, Q., Lau, H.C.: Qross: Qubo relaxation parameter optimisation via learning solver surrogates. In: 2021 IEEE 41st International Conference on Distributed Computing Systems Workshops (ICDCSW). pp. 35–40. IEEE (2021)
10. Kadowaki, T., Nishimori, H.: Quantum annealing in the transverse Ising model (1998)
11. Karp, R.M.: Reducibility among combinatorial problems (1972)
12. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing (2000)
13. Kitai, K., Guo, J., Ju, S., Tanaka, S., Tsuda, K., Shiomi, J., Tamura, R.: Designing metamaterials with quantum annealing and factorization machines. Physical Review Research **2**(1), 013319 (2020)
14. Koshikawa, A.S., Ohzeki, M., Kadowaki, T., Tanaka, K.: Benchmark test of black-box optimization using d-wave quantum annealer. Journal of the Physical Society of Japan **90**(6), 064001 (2021)
15. Koshikawa, A.S., Ohzeki, M., Miyama, M.J., Tanaka, K., Yamashita, Y., Stadler, J., Wick, O.: Combinatorial black-box optimization for vehicle design problem. arXiv preprint arXiv:2110.00226 (2021)
16. Lewis, M., Glover, F.: Quadratic unconstrained binary optimization problem preprocessing: Theory and empirical analysis. Networks **70**(2), 79–97 (2017)
17. Lodewijks, B.: Mapping np-hard and np-complete optimisation problems to quadratic unconstrained binary optimisation problems. arXiv preprint arXiv:1911.08043 (2019)
18. Lucas, A.: Ising formulations of many np problems. Frontiers in physics p. 5 (2014)
19. Nüßlein, J., Gabor, T., Linnhoff-Popien, C., Feld, S.: Algorithmic qubo formulations for k-sat and hamiltonian cycles. arXiv preprint arXiv:2204.13539 (2022)
20. Nüßlein, J., Roch, C., Gabor, T., Linnhoff-Popien, C., Feld, S.: Black box optimization using qubo and the cross entropy method. arXiv preprint arXiv:2206.12510 (2022)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-36030-5_4