7-2022

Enhancing a QUBO solver via data driven multi-start and its application to vehicle routing problem

Whei Yeap SUEN
Singapore Management University, wysuen@smu.edu.sg

Matthieu PARIZY

Hoong Chuin LAU
Singapore Management University, hclau@smu.edu.sg

Citation

SUEN, Whei Yeap; PARIZY, Matthieu; and LAU, Hoong Chuin. Enhancing a QUBO solver via data driven multi-start and its application to vehicle routing problem. (2022). *GECCO '22: Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 2251-2257.
Available at: https://ink.library.smu.edu.sg/sis_research/7624

Enhancing a QUBO solver via Data Driven Multi-start and its Application to Vehicle Routing Problem

Whei Yeap SUEN
wysuen@smu.edu.sg
School of Computing and Information Systems,
Singapore Management University
Singapore

Matthieu PARIZY
parizy.matthieu@fujitsu.com
Fujitsu Ltd.
Japan

Hoong Chuin LAU*
hclau@smu.edu.sg
School of Computing and Information Systems,
Singapore Management University
Singapore

ABSTRACT

Quadratic unconstrained binary optimization (QUBO) models have garnered growing interests as a strong alternative modelling framework for solving combinatorial optimization problems. A wide variety of optimization problems that are usually studied using conventional Operations Research approaches can be formulated as QUBO problems. However, QUBO solvers do not guarantee optimality when solving optimization problems. Instead, obtaining high quality solutions using QUBO solvers entails tuning of multiple parameters. Here in our work, we conjecture that the initial states adjustment method used in QUBO solvers can be improved, where careful tuning will yield overall better results. We propose a data-driven multi-start algorithm that complements the QUBO solver (namely Fujitsu Digital Annealer) in solving constrained optimization problems. We implemented our algorithm on a variety of capacitated vehicle routing problem instances, and empirical results show that different initial state adjustment methods do make a difference in the quality of solutions obtained by the QUBO solver.

KEYWORDS

QUBO, optimization, vehicle routing, quantum-inspired

ACM Reference Format:
Whei Yeap SUEN, Matthieu PARIZY, and Hoong Chuin LAU. 2022. Enhancing a QUBO solver via Data Driven Multi-start and its Application to Vehicle Routing Problem. In *Genetic and Evolutionary Computation Conference Companion (GECCO '22 Companion), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3520304.3533988

1 INTRODUCTION

The development of quantum technologies in recent years has lead scientists and researchers to revisit ubiquitous optimization problems with real world applications through the quantum lens. Currently, quantum related technologies come in many form, ranging from quantum-inspired classical machines [21, 42], to classical-quantum hybrid [11], to quantum hardware [15, 24]. Most of these quantum related hardwares are built to solve optimization problems in the form of quadratic unconstrained binary optimization (QUBO) problems [25, 33, 36]. As a result, a broad spectrum of optimization problems that are usually studied under the traditional operations research community can also be approached as QUBO problems, e.g. finance [10, 26, 35], computational biology [28], logistics optimization [27, 40], to machine learning [13] and many more highlighted in [16].

Given the growing prevalence of QUBO models, there is an increasing interest to study new techniques and algorithms to improve performance of QUBO solvers. In general, QUBO solvers do not guarantee optimality when solving an optimization problem. Obtaining high quality solutions, i.e. feasible solutions that are as close to the provably optimal solution as possible, relies on several factors. In the case of constrained optimization problems, the solution obtained by QUBO solvers also heavily depend on the penalty terms coefficients, as previously mentioned in [16]. In general, large penalty term coefficients used to enforce constraints can overwhelm the solution landscape, causing solvers to be stuck in local minima, while having too small a penalty allows infeasible solutions. Choosing the right penalty coefficients to use is a difficult task, and not easily generalizable. For example, [41] presents a framework to obtain a lower bound of the penalty coefficients in QUBO models of problems having a single equality constraint. In addition, hyper parameter optimization methods such as hyperopt [3] and Optuna [1] can also be used to try to tune the penalty term coefficients of QUBO models.

In this paper, we propose an approach that utilizes a multi-start scheme driven by a prediction model to improve the overall performance of QUBO solvers. Our experiments will be conducted using the third generation quantum-inspired Fujitsu Digital Annealer (DA) as our QUBO solver, and our focus will be on the Capacitated Vehicle Routing Problem (CVRP), even though the same approach may potentially be applied to solve other constrained optimization problems. This paper is divided into the following sections. We first discuss some related work in the area of solving CVRP using quantum related methods in Section 2. Sections 3 and 4 discuss our methodology, which involves the QUBO model formulation for the CVRP, and the data-driven algorithm we considered. We present and discuss our results in Section 5, and in Section 6, we conclude our findings and lay out potential extensions of this work.

*Corresponding author

GECCO '22 Companion, July 9–13, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9268-6/22/07…$15.00
https://doi.org/10.1145/3520304.3533988

2 LITERATURE REVIEW

The CVRP [12] is an NP-hard constrained optimization problem that has many real world applications. Throughout the years, various techniques and algorithms have been developed to solve the CVRP. Some traditional heuristic algorithms to solve the CVRP goes back as far as [4, 17, 39]. Aside from that, [38] provides a substantial list of excellent algorithms to solve the CVRP.

While the CVRP is a well-known and well studied problem, our interest in this paper lies in the quantum approach of solving the CVRP. Recent development in quantum technologies has given rise to new formulations and techniques to solve the CVRP. In 2017, In [37] presented a simplified and systematic approach to tuning a quantum annealing algorithm for large scale vehicle scheduling problems implemented on a classical computer. In [22], the authors proposed a new QUBO formulation of the vehicle routing problem (VRP) with time window and capacity, and subsequently implemented it using the D-Wave 2000Q machine. Work done in[19] presented a quantum annealing approach to solving the dynamic multi-depot CVRP. There, the authors modeled the dynamic multi-depot CVRP as a QUBO problem, and proposed a solution approach to solving it on the quantum annealing hardware from D-Wave. Taking a slightly different approach, [14] introduced a hybrid method for the CVRP using a quantum annealer. In their approach, the authors performed a 2-Phase-Heuristics onto the CVRP problem, where the first phase involves clustering customers based on their locations and demand, and the in the second phase, the algorithm perform routing optimization to determine the shortest path inside each cluster. Similarly, [5] also introduced new hybrid algorithms for solving both VRP and CVRP using D-Wave's Leap framework on well-established benchmark cases. The authors found that their hybrid methods produce results that are competitive with existing classical algorithms.

In contrast, our work in this paper focuses on developing a data driven multi-start algorithm to *enhance* the performance of a QUBO solver when solving the CVRP. Our work will be centered around the third generation quantum-inspired Fujitsu Digital Annealer (DA). In the next section, we present a QUBO formulation for the CVRP that fits the Fujitsu DA, and an algorithm that enhances it performance when applied to the CVRP.

3 METHODOLOGY

3.1 General QUBO Solving Framework

A combinatorial optimization problem can be formulated as a quadratic unconstrained binary optimization problem that takes the following form:

$$E(x_i) = \sum_{i,j} A_{i,j} x_i x_j \tag{1}$$

where $x_i \in \{0, 1\}$ is the binary decision variable of the problem, and $A_{i,j} \in \mathbb{R}$ are coefficients. The QUBO form in Equation (1) contains no constraints other than those requiring the decision variables to be binary. Nonetheless, most real world problems of interest consist of constraints that must be satisfied as the solver searches for optimal solutions. Combinatorial constrained optimization problems can be formulated as QUBO problems by introducing penalty terms into the objective function as a way to explicitly impose constraints. This results in augmented objective functions of the original problems. The penalty terms are formulated such that they equal zero for feasible solutions, but when they are violated, they contribute to a positive increase in the augmented objective function. Therefore, when a solver attempts to minimize the augmented objective function, the penalty terms are naturally driven to zero.

In general, when solving a QUBO problem, the solver's algorithm is repeated many times with varying initial states and penalty coefficients to search for the optimal results. Nonetheless, in this paper, our focus will be on the initial states selection to improve the performance of a QUBO solver. In most QUBO solving algorithms, the search algorithm is repeated with different initial states periodically so that the solution state of the solver does not get stuck in a local minimum. We decided to study the effects of the initial state in a QUBO solver, and explore initial state adjustment methods to obtain better results when solving optimization problems.

In this work, our results are centered around the third generation Fujitsu Digital Annealer (DA) as the main QUBO solver [32]. The Fujitsu DA is a quantum-inspired dedicated hardware architecture designed to solve combinatorial optimization problems by mapping to an Ising model [31]. It has fully coupled bit connectivity and high coupling resolution which allows it to solve various combinatorial optimization problems. Some applications of DA in the field of optimization problems include [2, 8, 9, 20, 23, 30]. In this paper, we use the third generation DA wwhich has a hybrid problem-solving system consisting of a software intervention layer (SIL) and a search core to solve binary quadratic programming (BQP) problem of up to 100,000 bits. In addition, the BQP Interface has built-in functions to handle one-hot constraints and linear inequality constraints without needing to convert them to penalty terms. These major features that are non-existent in previous generation DA's allow us to construct a QUBO formulation for the CVRP without using slack variables. As we will see in our problem formulation, the inequality constraint in our CVRP QUBO formulation will be declared separately from the objective function and penalty terms.

3.2 Problem Formulation

Our work focuses on the capacitated vehicle routing problem [12]. The CVRP is one of the many well-known variants of the vehicle routing problem. In the CVRP, we seek to find the shortest route for a fleet of vehicles with limited carrying capacity to pick up or deliver items at various locations. In addition, all vehicles used must start and end their tours at a designated depot. As we have seen from our literature review, there are many different QUBO formulations that have previously been used to solve the CVRP [19, 22, 37] depending on the platform or hardware used. In this paper, our CVRP QUBO formulation is an extension of a TSP formulation introduced in [29]. Using a binary decision variable that keeps track of the location nodes and their position in a tour, we added an extra index to track different vehicles. As a result, the decision variable of the CVRP in our QUBO formulation is defined as follows:

$$x_{il}^k = \begin{cases} 1 & \text{if city } i \text{ is the } l\text{-th city visited in vehicle } k\text{'s tour.} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The QUBO formulation we used for the CVRP is shown in Equation (3), where d_{ij} denotes the distance between city i and j.

$$\begin{aligned}\min \sum_{\substack{i\in[1,N]\\k\in[0,K-1]}} d_{01}x_{i1}^{k} + \sum_{\substack{i,j\in[1,N]\\l\in[1,L-1]\\k\in[0,K-1]}} d_{ij}x_{il}^{k}x_{j,l+1}^{k}\\ + \sum_{\substack{i\in[1,N]\\k\in[0,K-1]}} d_{10}x_{i,L}^{k} + A\sum_{i\in[1,N]}\left(\sum_{\substack{l\in[1,L]\\k\in[0,K-1]}} x_{il}^{k}-1\right)^2\\ + A\sum_{\substack{k\in[0,K-1]\\l\in[1,L]}}\left(\sum_{i\in[1,N]} x_{il}^{k}-1\right)^2 \end{aligned} \quad (3)$$

where N is the total number of locations including the depot, L is the number of customer locations each vehicle has to travel to, and K is the number of vehicles used. In our formulation, $i = 0$ is always the designated depot. The first three terms in Equation (3) together represent the objective function. As CVRP requires all vehicles to start and end at the depot, the variables $x_{00}^{k} = 1$ and $x_{0,L+1}^{k} = 1$ for all k's, as captured in the first and third term in Equation (3). The fourth term is a penalty term to ensure that each customer location can only be visited once. Similarly, the fifth and last term in Equation (3) ensures that each tour position of each vehicle can only be assigned to one customer location. In general, the two penalties can take different coefficients. However, since the goal of our work is to study an algorithm that focuses on the effects of initial state adjustment in solving QUBO problems, the penalty term coefficients are set to be the same for both penalties. In addition to the objective function and penalty terms, the CVRP also has another inequality constraint that ensures that each vehicle does not exceed their limited capacity when serving multiple customers, where

$$\sum_{\substack{i\in[1,N]\\l\in[1,L]}} q_i x_{il}^{k} \le Q \qquad \forall k \quad (4)$$

where q_i denotes the demand of customer i, and Q is the capacity limit of one vehicle. In this paper, we consider only cases where all vehicles have the same capacity limit. We note that Equation (4) is written separately from the objective function and penalty terms in Equation (3), because as we mentioned, the third generation Fujitsu DA is able to handle linear inequality constraints separately without having to convert them into penalty terms using slack variables.

4 DATA DRIVEN MULTI-START ALGORITHM

A prior work in [6] introduces an approach that learns an evaluation function that predicts the outcome of a local search algorithm, based on features of states visited during search. The approach consists of a primary solver denoted by π, and a surrogate model denoted by $\tilde{V}^{\pi}$. The solutions obtained by the primary solver will be used as new training data for the surrogate model designed to learn how different initial states will lead to better objective values, i.e. closer to the optimal value. Using this general framework, we developed a data-driven multi-start algorithm to improve the search algorithm carried out by the DA. Figure 1 illustrates our algorithm, which henceforth will be referred to as multi-start DA. The algorithm is described in Algorithm 1, with the following notations:

- N_s is the number of repetitions of the algorithm.
- T total time set for DA to solve the QUBO problem.
- Δt is the time interval where intermediate solution states are generated.
- $\vec{x}_n$'s are the intermediate states generated at intervals of Δt.
- $m \in [0, M]$ is the number of hill climbing steps.
- $F(\vec{x}_n)$ denotes the feature encoding function.
- E_f is the objective value obtained by DA after T.

Algorithm 1: Multi-start DA

for $s \leftarrow 0$ **to** 3 **do**
  Start with a random initial state and a fixed set of penalty coefficients.
  for $t \leftarrow 0$ **to** T **do**
    DA is used to solve the QUBO problem.
    At every interval of size Δt, the features of the solution state, $F(\vec{x}_n)$ is computed and stored.
  end
  The corresponding energy of $\vec{x}_f$ is denoted by E_f.
  A surrogate model is built/updated using the set $\{F(\vec{x}_n)\}$ as variables and E_f as the target.
end
for $s \leftarrow 3$ **to** N_s **do**
  for $t \leftarrow 0$ **to** T **do**
    DA is used to solve a QUBO problem.
    At every interval of size Δt, the features of the solution state, $F(\vec{x}_n)$ is computed and stored.
  end
  When DA finishes, the final solution state obtained, denoted as $\vec{x}_f$ is stored.
  The corresponding energy of $\vec{x}_f$ is denoted by E_f.
  The surrogate model is updated using the set $\{F(\vec{x}_n)\}$ as variables and E_f as the target.
  A hill climbing algorithm is performed to find the next initial state to be used by DA:
  The best state obtained after M hilclimbing steps will be fed back to DA as an initial state.
end

The goal of the multi-start DA algorithm is to improve the initial state selection of DA using past data that DA provides. Using features of solution states explored by the DA, the multi-start DA algorithm tries to learn the energy landscape of a corresponding problem through a surrogate model. To this end, there are three components that are important to the performance of the algorithm as a whole, namely the type of feature encoding function, surrogate model, and the local neighbourhood perturbation scheme. Our choices for each of these components heavily hinges on the type or problem we are solving, and their compatibility with the Fujitsu DA.

Firstly, the main challenge with exploring different feature encoding functions to encode solution states is that the solution states of QUBO problems are binary vectors. There are not many general

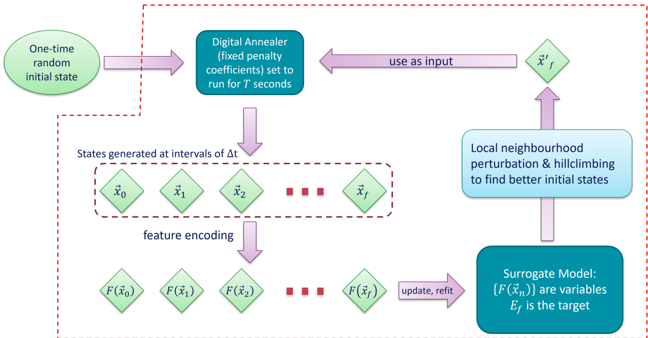

Figure 1: Illustration of our data-driven multi-start algorithm. Processes inside the red dotted box are repeated for N_s iterations.

features that we can extract from a binary solution state that can be applied to various types of problems, but on the other hand, studying features that are overly problem specific can sacrifice the overall robustness of the algorithm when applied on different types of problems. For the case of the CVRP, we define our feature encoding function as:

$$F\left(\vec{x}_f\right) = (\mu_d, \sigma_d, \mu_q, \sigma_q) \tag{5}$$

where μ_d and σ_d denotes the mean and standard deviation of the distance traveled by a vehicle in the solution respectively. Similarly, μ_q and σ_q denotes the mean and standard deviation of the occupied capacity of a vehicle in the solution respectively. Secondly, the surrogate model that we use in this work is a quadratic regression model. The quadratic regression model is chosen because it is simple to implement, quick to execute, and suits the type of data that we have. Last but not least, the local neighbourhood perturbation scheme we used in this work is described in Algorithm 2. The idea behind the local neighbourhood perturbation scheme in the multi-start DA algorithm is to use the current solution state to search for another feasible solution state that can be used as an initial state for the next DA input to obtain a better solution with the help of the surrogate model. This is akin to existing multi-start heuristics algorithm used in solving various other optimization problems [7, 18, 34]. Nonetheless, for our mult-start DA algorithm, we opted for something simpler and much more efficient to implement, so as to not further increase the computational cost of the algorithm. Not only that, the simplicity of the local neighbourhood perturbation algorithm described in Algorithm 2 is also easily customized to apply to other types of constrained optimization problems under our multi-start DA framework.

5 EXPERIMENTAL RESULTS

Our main goal of developing the data driven multi-start DA algorithm is to improve the general performance of the third generation Fujitsu DA. Thus, we design our experiments to compare the performances of these two methods. We briefly summarize our experimental setups for both methods in the following.

Algorithm 2: Local Neighbourhood Perturbation

```
Using the solution state x⃗_f.
for m ← 0 to M do
    while 1 do
        Randomly select 2 vehicles (k_1 and k_2), and denote
          the vehicles' used capacity as q_t^1 and q_t^2
          respectively.
        A random customer from k_1's tour is selected, and
          set their demand as q_i. Let the customer's position
          in vehicle k_1's tour be l_1.
        A random customer from k_2's tour is selected, and
          set their demand as q_j. Let the customer's position
          in vehicle k_2's tour be l_2.
        if q_t^1 − q_i + q_j ≤ Q and q_t^2 − q_j + q_i ≤ Q then
            The customer with demand q_i will be swapped
              to vehicle k_2, and will take over the position in
              tour l_1, and vice versa.
            break
        end
        else
            Re-select vehicles and customers randomly.
        end
    end
end
```

5.1 DA benchmark algorithm setup

The third generation Fujitsu DA is a general QUBO solver with a proprietary architecture and built-in algorithm. In order for us to ascertain that the multi-start DA algorithm is able to offer improvements when solving the same QUBO problem, we try to set up the DA experiments to be as similar as the multi-start DA experiments in terms of computational resources, problem parameters, and hardware settings. Our benchmark DA algorithm has the following settings and parameters:

- Total runtime of $T = 600s$.
- The penalty term coefficients, $A = 10d_{\max}$, where $d_{\max}$ is the maximum distance between any two cities in the problem instance.

5.2 Multi-start DA setup

In order to obtain a meaningful comparison with respect to the benchmark algorithm, our multi-start DA parameters and settings are defined to be as similar to the benchmark algorithm as possible. We set the following settings and parameters for the multi-start DA algorithm

- Total runtime per iteration is $60s$, with $\Delta t = 20$, and total iteration $N_s = 10$. This results in the same computational time limit for both algorithms.
- The penalty term coefficients, $A = 10d_{\max}$.
- Number of hill climbing steps $M = 200$. From our experiments, this part of the algorithm takes less than a second to implement, thus it does not drastically increase our total computational time.

5.3 Results

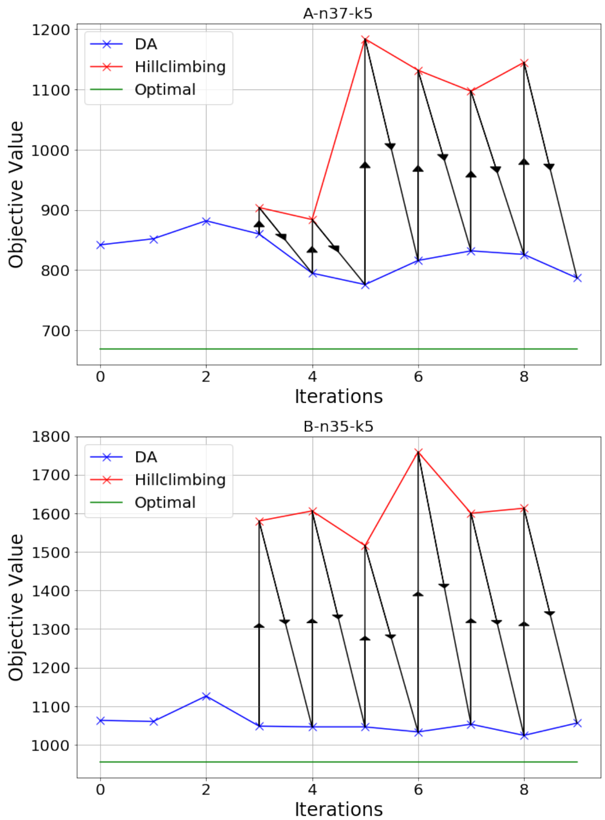

Figure 2: Illustration of single of the multi-start DA executions CVRP instances. The blue data points represent solutions obtained by the DA, while the red data points represent initial states predicted by performing hill climbing algorithm onto the surrogate model. In the first 3 iterations of DA, no hill climbing was performed. From the 4th iteration onwards, the black arrows represent the trajectory of solution states being fed back and forth between the DA and the surrogate model, in a continuous effort to update the surrogate model, then use it to predict a better initial state for the next DA iteration.

Before we discuss the difference in performance between the two different DA methods, we first draw our attention to the behaviour of a single execution of the multi-start DA algorithm. Figure 2 shows the typical evolutions of a single execution of the multi-start DA algorithm when used to solve a CVRP instance. There are a few important observations here. As indicated by the legend in Figure 2, blue data points represent solution states obtained by DA, and red data points represent predicted initial states by performing a hill climbing algorithm onto the surrogate model. Firstly, in the beginning of the algorithm, there is no hill climbing in place. This is because we want to generate a some data points before building the surrogate model, and performing hill climbing based on the resulting surrogate model. Subsequently, we observe the black arrows that goes back and forth between a blue data point and a red data point. This helps us illustrate the exchange of solution states that happen between the DA and the hill climbing algorithm:

- A black arrow that **starts from a blue data point and ends at a red data point** indicates that a solution state obtained by DA is being used by the surrogate and hill climbing algorithm to predict a new initial state that should result in a lower objective value. The solution state from DA (the blue data point) is first used to update the surrogate model. Then, a hill climbing algorithm is performed on the surrogate model to predict an initial state that will result in a lower objective function. This predicted initial state has the objective function represented by the red data point at the end of the arrow.
- Conversely, a black arrow that **starts from a red data point and ends at a blue data point** indicates that the predicted initial state is being relayed back into DA to be used as the initial state of the next iteration of DA's search algorithm. This results in a new solution state obtained by DA, with an objective value represented by the blue data point at the end of the arrow.

The results illustrated in Figure 2 has significant implications. We recall that the goal of our work is to enhance the performance of the third generation quantum-inspired Fujitsu DA using our data driven multi-start approach. Figure 2 shows that our multi-start DA algorithm is able to use solutions generated by DA to construct a surrogate model that can then be used to predict *good* initial states to use, i.e., initial states that will lead to a lower objective value when used in DA. The gap between the red and blue line implies that when solution states obtained by DA are used to update the surrogate model, it is able to suggest initial states that will lead to better solution states when when these suggested initial states are used by DA. Furthermore, our experimental setup indicates that this is achieved without increasing the total computational time significantly.

Next, Table 1 shows the performance difference between the benchmark DA algorithm and our multi-start DA algorithm when solving multiple CVRP instances. As a way to compare the general performance, we conduct 30 experiments on the same instance with the same fixed penalty coefficients and compared the mean optimality gap obtained by both methods. We define the optimality gap as

$$\Delta = 100 \times \frac{\text{obj}_{\text{algo}} - \text{obj}_{\text{opt}}}{\text{obj}_{\text{opt}}} \tag{6}$$

where obj_{algo} is the objective value obtained by either one of the two algorithms studied: the benchmark DA algorithm or our mult-start DA algorithm. obj_{opt} denotes the optimal solution of the problem instance. The optimal solutions of all instances studied here can be found in CVRPLIB [1]. Table 1 shows that our data driven multi-start algorithm using the DA is able to obtain significantly better solutions when solving CVRP instances. There are a few important discussion points we wish to highlight from the results in Table 1. Firstly, initial state adjustment method plays an important role in

[1]http://vrp.atd-lab.inf.puc-rio.br/index.php/en/

obtaining better solutions when solving constrained optimization problems. Our results showed that by developing an algorithm around improving the initial state adjustment between subsequent iterations, we can significantly improve the performance of a QUBO solver such as the Fujitsu DA. Besides that, our experiments are also structured in a way that is agnostic to the inherent algorithm of the DA. In our work, the multi-start DA algorithm uses the intermediary solution state outputs of the DA to build a surrogate model that is able to suggest better initial states to be used to obtain solution states that are closer to optimality. This general framework is not unique to DA and can be applied to other QUBO solvers.

Instances	Optimality Gap (%)	
	DA benchmark	Multi-start DA
A-n32-k5	24.28 ± 3.84	**9.75 ± 3.23**
A-n33-k5	24.44 ± 4.03	**11.57 ± 1.89**
A-n44-k6	46.87 ± 5.02	**27.62 ± 4.90**
A-n48-k7	40.90 ± 5.38	**19.94 ± 3.65**
B-n35-k5	18.97 ± 3.27	**6.13 ± 1.48**
B-n34-k5	13.96 ± 3.33	**3.89 ± 1.32**
B-n45-k6	51.30 ± 4.32	**24.27 ± 6.15**
B-n51-k7	46.67 ± 5.58	**18.65 ± 5.69**
E-n33-k4	20.23 ± 2.83	**9.40 ± 2.16**
E-n30-k3	21.65 ± 3.88	**6.90 ± 2.64**
P-n50-k7	33.59 ± 4.15	**16.81 ± 2.67**
P-n50-k10	28.06 ± 3.31	**12.60 ± 3.18**
P-n40-k5	27.92 ± 4.80	**6.59 ± 4.75**

Table 1: Comparing the performances of the DA benchmark algorithm with multi-start DA algorithm when solving CVRP. Numbers in bold denote lower optimality gaps between the two algorithms.

6 CONCLUSION AND FUTURE WORK

On the whole, our work explored the potential of initial states adjustment algorithms in improving performance of QUBO solvers using the third generation quantum-inspired Fujitsu DA as the core hardware in our study. We conjecture that a data driven multi-start approach that enhances the selection process of initial states in the DA's search algorithm can improve its overall performance. To this end, we implemented a data driven multi-start DA algorithm that uses a surrogate model created by learning solution states obtained by DA to predict initial states that will lead to better solutions when used by DA in subsequent iterations. Using the CVRP as the problem of interest, we compared the solutions obtained by both the multi-start DA and the benchmark DA algorithm.

Our work has shown that improving QUBO solvers' performance can also come in the form of better initial state adjustments in between iterations. Our empirical results with CVRP showed that we can construct effective algorithms to introduce better initial states into QUBO solvers, and there are much more to be explored in this area. A natural extension of our work would be to explore different types of optimization problems with this framework. Throughout our experiments, we have tried our best to keep the algorithm to be as generalizable as possible, such as our choice of surrogate model, feature encoding function, and local neighbourhood perturbation scheme. Furthermore, there are also compelling questions regarding the components in the overall multi-start DA algorithm framework. It will be interesting to study the effects of different surrogate models, feature encoding functions, and local neighbourhood perturbation schemes have in terms of enhancing the performance of QUBO solvers.

In addition, we would also like to investigate a more comprehensive QUBO solver framework that utilizes both penalty term coefficients tuning and initial state adjustment method. Recall that in our current work, the focus is on obtaining better initial states for the QUBO solver's search algorithm to produce better solutions. However, in any general QUBO solver, the optimization task involves penalty term coefficients tuning as well. Being able to systematically tune the penalty term coefficients while improving the initial states adjustment will provide a more complete insight to solving combinatorial optimization problems using QUBO solvers.

REFERENCES

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2623–2631.

[2] Hinako Asaoka and Kazue Kudo. 2020. Image Analysis Based on Nonnegative/Binary Matrix Factorization. *Journal of the Physical Society of Japan* 89, 8 (2020), 085001.

[3] James Bergstra, Dan Yamins, David D Cox, et al. 2013. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in science conference*, Vol. 13. Citeseer, 20.

[4] Lawrence D Bodin and Lon Berman. 1979. Routing and scheduling of school buses by computer. *Transportation Science* 13, 2 (1979), 113–129.

[5] Michał Borowski, Paweł Gora, Katarzyna Karnas, Mateusz Błajda, Krystian Król, Artur Matyjasek, Damian Burczyk, Miron Szewczyk, and Michał Kutwin. 2020. New hybrid quantum annealing algorithms for solving vehicle routing problem. In *International Conference on Computational Science*. Springer, 546–561.

[6] Justin Boyan and Andrew W Moore. 2000. Learning evaluation functions to improve optimization by local search. *Journal of Machine Learning Research* 1, Nov (2000), 77–112.

[7] Olli Bräysy, Geir Hasle, and Wout Dullaert. 2004. A multi-start local search algorithm for the vehicle routing problem with time windows. *European Journal of Operational Research* 159, 3 (2004), 586–605.

[8] Philippe Codognet. 2022. Modeling the Costas Array Problem in QUBO for Quantum Annealing. In *European Conference on Evolutionary Computation in Combinatorial Optimization (Part of EvoStar)*. Springer, 143–158.

[9] Eldan Cohen, Arik Senderovich, and J Christopher Beck. 2020. An Ising framework for constrained clustering on special purpose hardware. In *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Springer, 130–147.

[10] Jeffrey Cohen, Alex Khan, and Clark Alexander. 2020. Portfolio optimization of 60 stocks using classical and quantum algorithms. *arXiv preprint arXiv:2008.08669* (2020).

[11] Andrew Cross. 2018. The IBM Q experience and QISKit open-source quantum computing software. *APS* 2018 (2018), L58–003.

[12] George B Dantzig and John H Ramser. 1959. The truck dispatching problem. *Management science* 6, 1 (1959), 80–91.

[13] Prasanna Date, Davis Arthur, and Lauren Pusey-Nazzaro. 2021. QUBO formulations for training machine learning models. *Scientific Reports* 11, 1 (2021), 1–10.

[14] Sebastian Feld, Christoph Roch, Thomas Gabor, Christian Seidel, Florian Neukart, Isabella Galter, Wolfgang Mauerer, and Claudia Linnhoff-Popien. 2019. A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer. *Frontiers in ICT* 6 (2019), 13.

[15] Elizabeth Gibney. 2017. D-wave upgrade: How scientists are using the world's most controversial quantum computer. *Nature News* 541, 7638 (2017), 447.
[16] Fred Glover, Gary Kochenberger, and Yu Du. 2019. Quantum Bridge Analytics I: a tutorial on formulating and using QUBO models. *4OR* 17, 4 (2019), 335–371.
[17] Bruce L Golden, Thomas L Magnanti, and Hien Q Nguyen. 1977. Implementing vehicle routing algorithms. *Networks* 7, 2 (1977), 113–148.
[18] András György and Levente Kocsis. 2011. Efficient multi-start strategies for local search algorithms. *Journal of Artificial Intelligence Research* 41 (2011), 407–444.
[19] Ramkumar Harikrishnakumar, Saideep Nannapaneni, Nam H Nguyen, James E Steck, and Elizabeth C Behrman. 2020. A quantum annealing approach for dynamic multi-depot capacitated vehicle routing problem. *arXiv preprint arXiv:2005.12478* (2020).
[20] Seo Hong, Pierre Miasnikof, Roy Kwon, and Yuri Lawryshyn. 2020. Market Graph Clustering Via QUBO and Digital Annealing. *arXiv preprint arXiv:2006.10716* (2020).
[21] Takahiro Inagaki, Yoshitaka Haribara, Koji Igarashi, Tomohiro Sonobe, Shuhei Tamate, Toshimori Honjo, Alireza Marandi, Peter L McMahon, Takeshi Umeki, Koji Enbutsu, et al. 2016. A coherent Ising machine for 2000-node optimization problems. *Science* 354, 6312 (2016), 603–606.
[22] Hirotaka Irie, Goragot Wongpaisarnsin, Masayoshi Terabe, Akira Miki, and Shinichirou Taguchi. 2019. Quantum annealing of vehicle routing problem with time, state and capacity. In *International Workshop on Quantum Technology and Optimization Problems*. Springer, 145–156.
[23] Mohammad Javad-Kalbasi, Keivan Dabiri, Shahrokh Valaee, and Ali Sheikholeslami. 2019. Digitally Annealed Solution for the Vertex Cover Problem with Application in Cyber Security. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2642–2646.
[24] Mark W Johnson, Mohammad HS Amin, Suzanne Gildert, Trevor Lanting, Firas Hamze, Neil Dickson, Richard Harris, Andrew J Berkley, Jan Johansson, Paul Bunyk, et al. 2011. Quantum annealing with manufactured spins. *Nature* 473, 7346 (2011), 194–198.
[25] Gary Kochenberger, Jin-Kao Hao, Fred Glover, Mark Lewis, Zhipeng Lü, Haibo Wang, and Yang Wang. 2014. The unconstrained binary quadratic programming problem: a survey. *Journal of combinatorial optimization* 28, 1 (2014), 58–81.
[26] G Kochenberger and M Ma. 2019. Quantum computing applications of QUBO models to portfolio optimization. *White paper. University of Colorado, Denver* (2019).
[27] Mark Lewis, Bahram Alidaee, and Gary Kochenberger. 2005. Using xQx to model and solve the uncapacitated task allocation problem. *Operations research letters* 33, 2 (2005), 176–182.
[28] Mark W Lewis, Amit Verma, and Todd T Eckdahl. 2021. Qfold: a new modeling paradigm for the RNA folding problem. *Journal of Heuristics* (2021), 1–23.
[29] Andrew Lucas. 2014. Ising formulations of many NP problems. *Frontiers in physics* (2014), 5.
[30] Osman Asif Malik, Hayato Ushijima-Mwesigwa, Arnab Roy, Avradip Mandal, and Indradeep Ghosh. 2020. Binary Matrix Factorization on Special Purpose Hardware. *arXiv preprint arXiv:2010.08693* (2020).
[31] Satoshi Matsubara, Motomu Takatsu, Toshiyuki Miyazawa, Takayuki Shibasaki, Yasuhiro Watanabe, Kazuya Takemoto, and Hirotaka Tamura. 2020. Digital annealer for high-speed solving of combinatorial optimization problems and its applications. In *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 667–672.
[32] Hiroshi Nakayama, Junpei Koyama, Noboru Yoneoka, and Toshiyuki Miyazawa. 2021. Description: Third generation Digital Annealer technology.
[33] Robert Patton, Catherine Schuman, Thomas Potok, et al. 2019. Efficiently embedding QUBO problems on adiabatic quantum computers. *Quantum Information Processing* 18, 4 (2019), 1–31.
[34] Alfredo S Ramos, Elias Olivares-Benitez, and Pablo A Miranda-Gonzalez. 2021. Multi-start iterated local search metaheuristic for the multi-mode resource-constrained project scheduling problem. *Expert Systems* (2021), e12830.
[35] G Rosenberg. 2016. Finding optimal arbitrage opportunities using a quantum annealer. *1QB Information Technologies Write Paper* (2016), 1–7.
[36] Yuya Sugie, Yuki Yoshida, Normann Mertig, Takashi Takemoto, Hiroshi Teramoto, Atsuyoshi Nakamura, Ichigaku Takigawa, Shin-ichi Minato, Masanao Yamaoka, and Tamiki Komatsuzaki. 2021. Minor-embedding heuristics for large-scale annealing processors with sparse hardware graphs of up to 102,400 nodes. *Soft Computing* 25, 3 (2021), 1731–1749.
[37] Alex Syrichas and Alan Crispin. 2017. Large-scale vehicle routing problems: quantum annealing, tunings and results. *Computers & Operations Research* 87 (2017), 52–62.
[38] Paolo Toth and Daniele Vigo. 2002. *The vehicle routing problem*. SIAM.
[39] EG Unwin. 1968. Bases for vehicle fleet scheduling. *Journal of the Operational Research Society* 19, 2 (1968), 201–202.
[40] Davide Venturelli, D Marchand, and Galo Rojo. 2016. Job shop scheduling solver based on quantum annealing. In *Proc. of ICAPS-16 Workshop on Constraint Satisfaction Techniques for Planning and Scheduling (COPLAS)*. 25–34.
[41] Amit Verma and Mark Lewis. 2020. Penalty and partitioning techniques to improve performance of QUBO solvers. *Discrete Optimization* (2020), 100594.
[42] Masanao Yamaoka, Chihiro Yoshimura, Masato Hayashi, Takuya Okuyama, Hidetaka Aoki, and Hiroyuki Mizuno. 2015. A 20k-spin Ising chip to solve combinatorial optimization problems with CMOS annealing. *IEEE Journal of Solid-State Circuits* 51, 1 (2015), 303–309.