Detailed Project Proposal

First name: *Raufs*
Last name: *Dunamalijevs*
Matriculation number: *1803068*
Supervisor: *Professor John McCall*
Industry Advisor: *Dr Mayowa Ayodele*

Table of Contents

Project Title

Estimating Penalty Coefficients for QUBO Problems.

Background

Quadratic Unconstrained Binary Optimization (QUBO) is a model that can be used to represent a lot of combinatorial optimization problems including the Travelling Salesman Problem (Lucas, 2014) and Quadratic Knapsack (Glover, Kochenberger and Du, 2019). It is widely applied in the quantum computing area as such models can be mapped onto a network of qubits and then solved in a process called quantum annealing (Lewis and Glover, 2017).
Quantum annealing is similar to simulated annealing, however, it is quantum fluctuations that cause state transitions instead of thermal fluctuations (Kadowaki and Nishimori, 1998). In an attempt to emulate quantum effects with classical algorithms, Fujitsu Laboratories has developed a Digital Annealer - CMOS hardware designed to solve fully connected QUBO problems (Aramon *et al.*, 2019).

QUBO models are used to represent constrained optimization problems in a single unconstrained function of the following form: $\min x'Qx$, where x is a binary vector and Q is a square matrix of constants (Glover, Kochenberger and Du, 2019). Since the function is unconstrained, the constraints of the original problem are transformed into quadratic penalties, which are added to the objective function. This penalty function will worsen the objective function if the configuration of the binary decision vector is infeasible. The amount by which the objective function will be affected depends on 2 factors:

1. The severity of the broken constraints. Therefore, if no constraints are broken, the penalty function will be equal to 0, leaving the original objective function unimpaired.
2. Penalty coefficient, P, which is multiplied by the penalty function to control the degree by which the penalty function will influence the overall fitness.

If the penalty coefficient is too low, the broken constraints will be undervalued, and the solution produced by the optimizer will be infeasible. On the other hand, if the penalty coefficient is too large, the solution process will be negatively impacted as the penalties will overwhelm the objective function making it harder to differentiate between good and bad solutions (Glover, Kochenberger and Du, 2019).

It is possible to use domain expertise to select an acceptable P in some cases (Glover, Kochenberger and Du, 2019), but it has also been shown that some problems can have solutions of better quality if you choose penalty coefficient carefully (Şeker, Tanoumand and Bodur, 2020).
Finding an optimal penalty value is not trivial and different techniques have been proposed to estimate it (Verma and Lewis, 2020; Huang *et al.*, 2021).

This project will focus on studying QUBO, algorithms used for solving QUBO problems, studying the existing techniques for the penalty coefficient optimization, implementing one of them, making a new method for P optimization and implementing it, and, finally, comparing the quality of the solutions produced using penalties generated by the existing and the new methods with different algorithms.

Motivation

There are many combinatorial optimization problems from industry, government and science that can be reformulated in QUBO (Glover, Kochenberger and Du, 2019). Subsequently, they can be solved on one of the QUBO solvers like quantum annealing devices developed by D-Wave Systems (Johnson *et al.*, 2011) or Digital Annealers developed by Fujitsu Laboratories (Aramon *et al.*, 2019). All the problems that are being solved on D-Wave quantum annealing devices are formulated as an objective function in QUBO format or an equivalent Ising function (Cruz-Santos, Venegas-Andraca and Lanzagorta, 2019), while the problems solved on DA are formulated only in QUBO (Aramon *et al.*, 2019). Both types of devices are commercially available and are actively used to solve real-world problems, some of which include:

1. Finding optimal routes for collecting delivery parts in warehouses (Fujitsu, no date).
2. Searching for molecules with desired properties within a predefined chemical space to design new drugs (Snelling *et al.*, 2020).
3. Optimizing travel routes in real-time in response to traffic conditions (D-Wave Systems, 2021).
4. Optimizing the process of painting car bodies as they travel down the assembly line to minimize the number of times that the workers need to switch between colours (D-Wave Systems, 2021).

Estimating better P coefficients will help to produce solutions of a higher quality and break fewer constraints. The new method, if it proves to be more effective than the existing ones, can be used to find more optimal penalty scalars for problems yet to be solved and for the ones that are already being solved using QUBO formulation. It can be applied to the industry to potentially improve solutions at no significant costs. And since better solutions can lead to greater utilization of resources and save money, it is highly desirable to use a better P estimation algorithm if such is found.

Aims & Objectives

AIM
Create a new method of estimating penalty coefficients for QUBO problems and compare it to an already existing method.

OBJECTIVE 1
Study QUBO and the algorithms used for solving problems in such formulation.

OBJECTIVE 2
Study the existing methods of estimating P and critically evaluate them.

OBJECTIVE 3
Implement one of the studied methods.

OBJECTIVE 4
Propose and implement a new method for estimating P.

OBJECTIVE 5
Formulate QUBOs for instances of a single combinatorial optimization problem and run optimization algorithms with P coefficients generated by an existing method (implemented in Objective 3) and by a new method (implemented in Objective 4).

OBJECTIVE 6
Compare the quality of the produced solutions. Disseminate.

Key Techniques

QUBO formulation

Many combinatorial optimization problems can be modelled as QUBOs. Some of the methods used to reformulate different types of such problems are described by Glover, Kochenberger and Du (2019).

Algorithms for solving QUBOs

There are different algorithms for solving QUBO models. Some of them are tabu search and simulated annealing (Beasley, 1998), quantum annealing (Kadowaki and Nishimori, 1998) and the Digital Annealer's algorithm (Aramon *et al.*, 2019).

PyQUBO

PyQUBO is a Python library that can formulate QUBOs out of the objective functions and the constraints of the combinatorial optimization problems (Zaman, 2021). It allows to solve the defined QUBOs with simulated annealing or using a D-Wave quantum annealing machine if one is available.

qbsolv

Qbsolv is a Python library developed by D-Wave Systems (Booth and Reinhardt, 2017). It can be used to find a solution to an already formulated QUBO problem by splitting it into smaller subQUBOs, solving them separately and combining the results. This is useful when QUBO size is too large, meaning that it cannot be mapped on a D-Wave quantum annealer. The decomposed QUBO can then be solved with a quantum annealer or using tabu search on a classical device.

Analytical approach to estimating P

The analytical approach to estimating P was described by Glover, Kochenberger and Du (2019). It involves using domain knowledge to approximate an average-case value of the original objective function (without taking the penalties or constraints into consideration). A percentage (75% to 150%) of this value is then used to set an initial penalty coefficient, which can be further improved by running the solver and iteratively updating P until a satisfactory solution quality is reached.

Noisy Intermediate-Scale Quantum approach to estimating lower bound of P

Verma and Lewis (2020) proposed a more systematic method of estimating P. Their approach involves calculating the maximum positive change that a single bit flip can do to the original objective function and then setting the penalty scalar to that number.

Machine Learning approach to predicting P

QUBO relaxation parameter optimisation via learning solver surrogates (QROSS) uses machine learning to build surrogate models of QUBO solvers (Huang *et al.*, 2021). This allows to capture the common structure shared by different instances of the same combinatorial optimization problem. The trained model can be used to predict good penalty coefficients for the yet unseen problem instances.

Legal, Social, Ethical, Professional and Security issues

This project involves Fujitsu Laboratories. The penalty coefficients estimated by both the proposed and existing solutions will be tested using multiple optimization algorithms. One of such algorithms will run on a Digital Annealer hardware that belongs to Fujitsu. To minimize legal and security issues, the transformed QUBO problem with proposed penalty values will be sent to the industry advisor, Dr Ayodele, who works at the company. Dr Ayodele will run the optimizations on the Digital Annealer and send back the results.

Project Plan

Objective 1
Study QUBO and the algorithms used for solving problems in such formulation.

- Study QUBO model and its applications.
- Study algorithms used for solving QUBOs.
- Study techniques used for formulating QUBOs.
- Study relevant programming libraries.

Objective 2
Study the existing methods of estimating P and critically evaluate them.

- Study the existing techniques for estimating P.
- Evaluate them.

Objective 3
Implement one of the studied methods.

- Choose one P estimation technique.
- Implement it.

Objective 4
Propose and implement a new method for estimating P.

- Propose a new method for estimating P.
- Explain the advantages of the proposed method over the existing methods.
- Implement it.

Objective 5
Formulate QUBOs for instances of a single combinatorial optimization problem and run optimization algorithms with P coefficients generated by an existing method and by a new method.

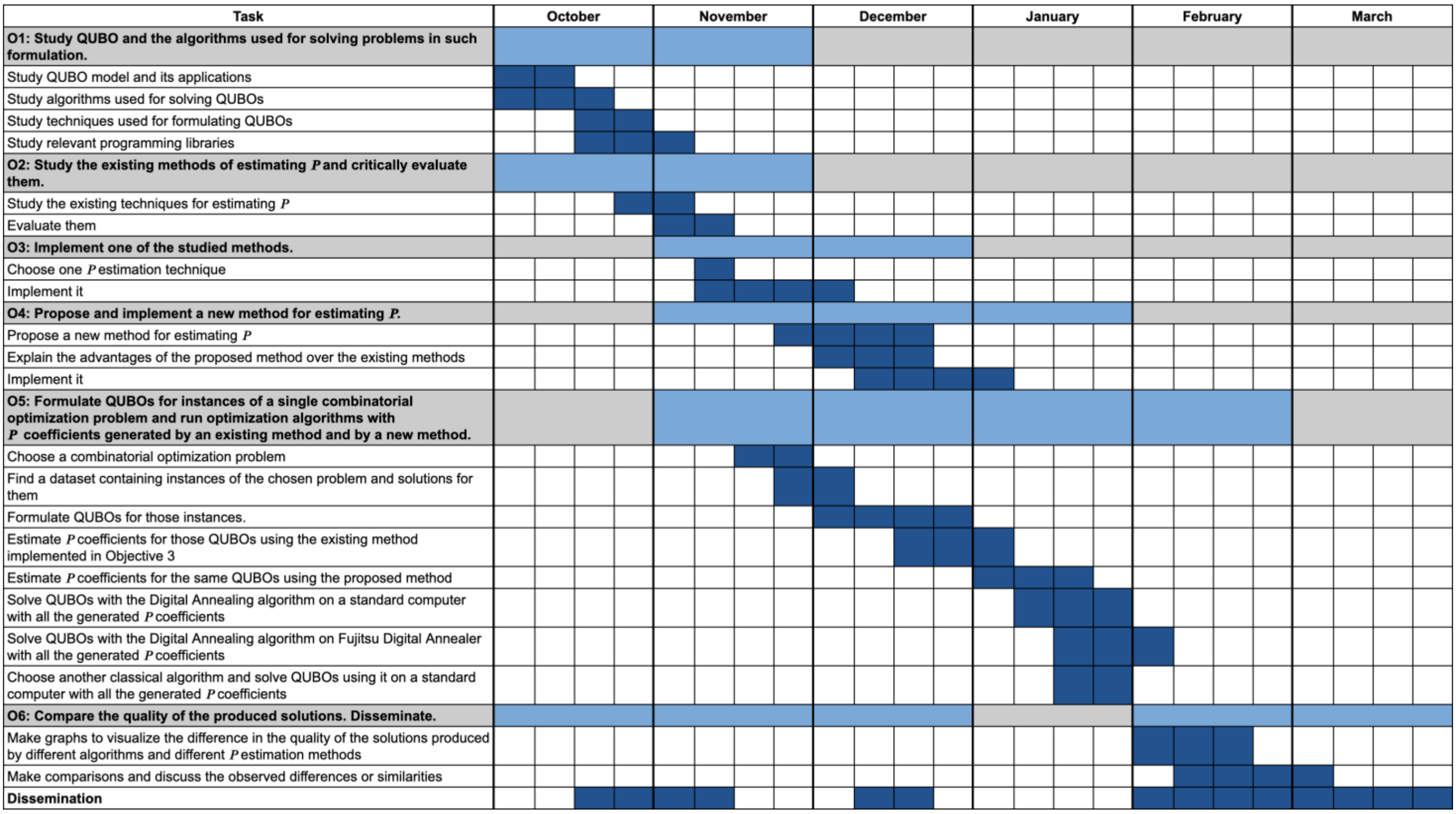
- Choose a combinatorial optimization problem.
- Find a dataset containing instances of the chosen problem and solutions for them.
- Formulate QUBOs for those instances.

- Estimate P coefficients for those QUBOs using the existing method implemented in Objective 3.
- Estimate P coefficients for the same QUBOs using the proposed method.
- Solve QUBOs with the Digital Annealing algorithm on a standard computer with all the generated P coefficients.
- Solve QUBOs with the Digital Annealing algorithm on Fujitsu Digital Annealer with all the generated P coefficients.
- Choose another classical algorithm and solve QUBOs using it on a standard computer with all the generated P coefficients.

Objective 6
Compare the quality of the produced solutions. Disseminate.

- Make graphs to visualize the difference in the quality of the solutions produced by different algorithms and different P estimation methods.
- Make comparisons and discuss the observed differences or similarities.
- Disseminate.

Gantt Chart



References

Aramon, M. *et al.* (2019) ‘Physics-Inspired Optimization for Quadratic Unconstrained Problems Using a Digital Annealer’, *Frontiers in Physics*, 0(APR), p. 48. doi: 10.3389/FPHY.2019.00048.

Beasley, J. (1998) ‘Heuristic algorithms for the unconstrained binary quadratic programming problem’, *undefined*.

Booth, M. and Reinhardt, S. P. (2017) ‘Partitioning Optimization Problems for Hybrid Classical / Quantum Execution TECHNICAL REPORT’.

Cruz-Santos, W., Venegas-Andraca, S. E. and Lanzagorta, M. (2019) ‘A QUBO Formulation of Minimum Multicut Problem Instances in Trees for D-Wave Quantum Annealers’, *Scientific Reports 2019 9:1*, 9(1), pp. 1–12. doi: 10.1038/s41598-019-53585-5.

D-Wave Systems (2021) *Volkswagen: Navigating Tough Automotive Tasks with Quantum Computing*. Available at: www.dwavesys.com/d-wave-launch (Accessed: 5 October 2021).

Fujitsu (no date) *Reducing traveling distance for parts picking operations by up to 45%*. Available at: https://www.fujitsu.com/global/services/business-services/digital-annealer/case-studies/201804-fjit.html (Accessed: 5 October 2021).

Glover, F., Kochenberger, G. and Du, Y. (2019) ‘Quantum Bridge Analytics I: a tutorial on formulating and using QUBO models’, *4OR*, 17(4), pp. 335–371. doi: 10.1007/S10288-019-00424-Y.

Huang, T. *et al.* (2021) ‘QROSS: QUBO Relaxation Parameter optimisation via Learning Solver Surrogates’, *2021 IEEE 41st International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pp. 35–40. doi: 10.1109/ICDCSW53096.2021.00013.

Johnson, M. W. *et al.* (2011) ‘Quantum annealing with manufactured spins’, *Nature 2011 473:7346*, 473(7346), pp. 194–198. doi: 10.1038/nature10012.

Kadowaki, T. and Nishimori, H. (1998) ‘Quantum annealing in the transverse Ising model’, *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 58(5), pp. 5355–5363. doi: 10.1103/PHYSREVE.58.5355.

Lewis, M. and Glover, F. (2017) ‘Quadratic Unconstrained Binary Optimization Problem Preprocessing: Theory and Empirical Analysis’, *Networks*, 70(2), pp. 79–97. Available at: https://arxiv.org/abs/1705.09844v1 (Accessed: 2 October 2021).

Lucas, A. (2014) ‘Ising formulations of many NP problems’, *Frontiers in Physics*, 0, p. 5. doi: 10.3389/FPHY.2014.00005.

Şeker, O., Tanoumand, N. and Bodur, M. (2020) ‘Digital Annealer for quadratic unconstrained binary optimization: a comparative performance analysis’. Available at: https://arxiv.org/abs/2012.12264v1 (Accessed: 3 October 2021).

Snelling, D. *et al.* (2020) ‘A Quantum-Inspired Approach to De-Novo Drug Design’. doi: 10.26434/CHEMRXIV.12229232.V1.

Verma, A. and Lewis, M. (2020) ‘Penalty and partitioning techniques to improve performance of QUBO solvers’, *Discrete Optimization*, p. 100594. doi: 10.1016/J.DISOPT.2020.100594.

Zaman, M. (2021) ‘PyQUBO: Python Library for Mapping Combinatorial Optimization Problems to QUBO Form.’, *CoRR*, abs/2103.01708. Available at: https://arxiv.org/abs/2103.01708 (Accessed: 4 October 2021).