Requirements Engineering

Raufs Dunamalijevs

Contents

List of Acronyms and Abbreviations

CO	Combinatorial Optimisation
DA	Digital Annealer
M	Penalty Coefficient
QA	Quantum Annealing
QUBO	Quadratic Unconstrained Binary Optimisation

To fulfil the aim of the project, a software solution that will estimate the coefficient *M* of QUBO problems using both novel and existing algorithms must be made. This solution should then be used to conduct experiments to find out how the algorithms compare. As the project is split into two parts, software development and experiments, the requirements will consist of two sections. Each of the sections will be further split into functional and non-functional requirements. The requirements are defined using the MoSCoW methodology. This methodology divides the requirements into four groups of different priority (high to low): *Must Have*, *Should Have*, *Could Have* and *Will Not Have*. Moreover, every requirement could have multiple minor requirements with varying priorities that need to be completed to fulfil the significant requirement. While additional subheadings are not used for them, they are also ordered according to MoSCoW methodology.

Software

Functional

Must

- The solution must generate coefficients *M* to already formulated QUBO problems.
    - The inputted QUBO should be split into the original objective function and the constraint function.
    - It could generate the *M* to the equivalent Ising model.
- At least one novel algorithm must be developed and implemented.
    - The algorithm must generate *M* given the original objective function with the constraint function.
    - It must consider the constraint function as this is the gap in the research identified in the literature review.
- At least one existing algorithm must be implemented.
    - This algorithm could be the Numerical 2 approach from the literature review (4.3).

Should

- It should be easy to specify the problem that needs *M* to be estimated.
    - Two dictionaries that map quadratic variables to coefficients could represent the objective and constraint functions of QUBO.
    - Two matrixes could represent coefficients of objective and constraint functions of QUBO.
- The solution should allow the user to choose the *M* estimation algorithm.
- The software should work with a single constraint function.
    - Although the developed algorithm could potentially work with multiple constraint functions, this is beyond this project's scope.
        - The solution developed will only return a single *M* as only one constraint function is expected.
    - Multiple constraint functions could be united into one. It could be inputted into the solution to generate an *M*.
        - Then the generated *M* would apply to all the original constraint functions.

Will not

- The solution will not solve the QUBO.
    - It only estimates the *M*.
    - The *M* generated can then be used to solve the problem with QUBO solvers, including QA, DA, or other classical algorithms.

Non-functional

Must

- The software must be documented.
    - The documentation must explain the purpose and functionality of implemented classes and methods.
    - The documentation should follow NumPy docstring standard.

Should

- The solution should be programmed in Python 3.
    - Because many QUBO-related libraries listed in the literature review are available in Python 3.
- The basic working version of the solution should be completed by Sunday, 16th of January 2022.
    - As the proof-of-concept demonstration will happen in the following week.
    - The basic version of the software must be completed before the experiments can take place. Having the basic version by the set deadline will leave enough time for thorough experimentation and improvements.

Could

- The software to be made could be a Python module.
    - As QUBO solving libraries are available in Python, generating the *M* in the same language is convenient. Then a single loop can be used to define QUBO, generate *M*, redefine QUBO with new coefficients and solve it. This will make experiments automatable, and the *M* estimation algorithms could be tested on entire datasets efficiently.
    - The produced module can easily be imported into new and existing projects using the *import* keyword, given that the module folder is copied into the project.

Experiments

Functional

Must

- All implemented algorithms must be used for experiments.
    - Including the algorithms that were implemented from the literature and self-proposed algorithms.
    - As the aim of the project is to compare the algorithms.

- The coefficients *M* generated by the algorithms must be used to solve QUBO problems.
  - The solutions and the time required to reach them must be recorded for further analysis.
  - For the same reason, the feasibility of the proposed solutions must be determined and recorded.
- The QUBO problems must belong to two different CO problems and come from two distinct datasets.
  - Datasets must be publicly available.
    - To make the findings reproducible.
    - And to avoid ethical problems when running experiments externally.
  - One of the datasets could include travelling salesman problems.
    - Then the formulated QUBOs for problem instances could be provided by industry advisor, Dr Ayodele.
  - Another dataset could include quadratic assignment problems.
    - Then the formulated QUBOs for problem instances could be provided by industry advisor, Dr Ayodele.
  - Another dataset could include multidimensional knapsack problems.
    - Then the formulated QUBOs for problem instances could be provided by industry advisor, Dr Ayodele.
  - Another dataset could include generalised assignment problems.
  - Another dataset could include bin packing problems.

Should

- The QUBOs with generated coefficients *M* should be solved on a classical computer using QUBO solvers defined in the literature review (3.5).

Could

- An HPC provided by National Subsea Centre could be used to run the classical computer experiments faster.
- A Fujitsu Digital Annealer provided by industry advisor, Dr Ayodele, could be used to run the experiments.

Non-functional

Must

- If QUBOs for the chosen datasets are not available, they must be formulated manually.
  - QUBOs produced should have a single constraint function as the implemented software will only generate a single *M*. Multiple constraint functions, however, could be joined into one by summation.
- Rigorous statistical testing and analysis of results must be performed to compare the implemented *M* estimation algorithms.
  - The results should be tested for significance.
- Visualisations demonstrating the results of experiments should be produced.
  - The plots could be made for every combination of datasets, QUBO solvers and *M* algorithms.

  - A series of bar plots could be produced that compare the proportion of feasible and infeasible QUBO solutions obtained using different *M* estimation algorithms.
  - The run time distributions of QUBO solving could be visualised to see how coefficients *M* generated by different algorithms affect the run time.

Should

- All experimental hypotheses should be tested until the mid-week of February.
- The experiment analysis should be completed by March, including publication-quality visualisations.

Could

- The experiments could be performed using Jupyter Notebooks.
  - As Jupyter Notebooks are easy to follow and eloquent.
  - As the experiments could be reproduced by rerunning the notebooks.