IS 470 Mid Term Report: Quantum Inspired Optimization Methods Research and Benchmarking

Lam Ying Sheng

Advisor: Prof. Lau Hoong Chin

October 2019

1 Introduction

1.1 Problem Statement

There are many intractable optimisation problems in important domains such as vehicle routing, urban logistics, air sea port scheduling, pharmaceutical research, medicine hospitals, and etc which rely on solving hard optimisation problems. Finding global optima for these problems using general purpose solvers is a challenge because an exact solution cannot be proved to be found in less than exponential time.

The project aims to explore the performance of quantum inspired algorithms such as Simulated Quantum Annealing (SQA). By collating the results, a comparison will be done against Fujitsu's Digital Annealer for standard TSPLIB problems.

1.2 Objectives and Goals of Research

The objective of the project is to deliver benchmarked results of standard TSPLIB problems using Simulated Quantum Annealing against Fujitsu's Digital Annealer (DA). Through this research, we hope to better understand if quantum inspired algorithms can provide advantages as a general-purpose solver whether in terms of speed & scale, or in the quality of solution obtained.

The work includes building a simulated quantum annealing package which aims to find the global optima of a given QUBO objective function by using Quantum fluctuations of the energy gap.

The code deliverable of this project include: a package implementing Simulated Quantum Annealing, and a QUBO to Ising problem converting function.

1.3 Research Motivation

To solve optimisation problems, we make use of algorithms that either do hill-climbing, or metropolis algorithms to search a solution space. In the context of a minimisation problem, such algorithms lie in the classical domain which seek the global minimum point by placing a solution at some point in the landscape, and allowing that solution to move based on local variations.

Classical Algorithms will avoid climbing hills that are too high, and are prone to leading the traveler into local minimums which are not the global minimum.

Quantum Annealing (QA) begins with the solution simultaneously occupying many coordinates, making use of the quantum effect of superposition. This idea makes use of the adiabatic theorem of computation and quantum tunneling to gradually increase the probability of finding the solution in states corresponding to deep minima, as annealing progresses.

Quantum tunneling allows the solution to pass through hills—rather by moving to the lower energy state, while the Adiabatic Theorem of Computation allows the system to remain in a state of internal equilibrium.

Phase	Scope of Work	Duration
1	Understand workflow of the Digital Annealer Read through existing literature about formulating QUBO's and simulated quantum annealing / path integral monte carlo Understand about benchmarks to be used	Week 1 - 3 (3 weeks)
2	Start building and doing small tests of packages built. Build simulated quantum annealer package	Week 4 - 10 (6 weeks)
3	Compare and test the different optimizers while performing optimizations of the different classes of solvers	Week 11 - 13 (3 weeks)
4	Consolidation and presentation of results	Week 13 - 14 (2 weeks)

1.4 Timeline of work

2 Introduction to Simulated Quantum Annealing

2.1 Why Simulated Quantum Annealing?

Quantum Annealing associates a cost function $f : \{0,1\}^n \rightarrow R$ with a Hamiltonian H_f such that the ground state of the Hamiltonian H_f corresponds to the tour that minimises f. To achieve H_f, we prepare a superposition of N Qubits in ground state of a universe X transverse field.

$$H_0 = -\sum_{i=1}^{N} \sigma_i^x$$

We make use of the annealing schedule given in Quantum Annealing [2] below, to lower the thermal state of our system to the ground state hamiltonian H_f:

$$H := H(s) = (1-s)H_0 + sH_f$$

As we change s, the equilibrium thermal state of the system will evolve according to the adiabatic theorem of computation [5] with the adiabatic parameter:

$$\sigma(s) := e^{-\beta H(s)},\ Z(s) := tr(e^{-\beta H(s)})$$

The adiabatic theorem guarantees that if we lower s slowly enough ($\beta = \infty$), the system will remain in the ground state.

Quantum mechanically constructed and explored paths can explore regions unavailable to the classical path, making use of tunneling phenomena. This enables highly non-convex systems to be easily explored and solved to an acceptable margin of an optimal solution in reasonable amounts of time.

To make use of Quantum Annealing, we require quantum hardware which performs quantum tunneling by making use of a superposition of Qubit states. However, such hardware does not generalise well and is not available at reasonable cost. Hence, we attempt to simulate it on a classical computer using path integrals.

This is illustrated in Figure 1:

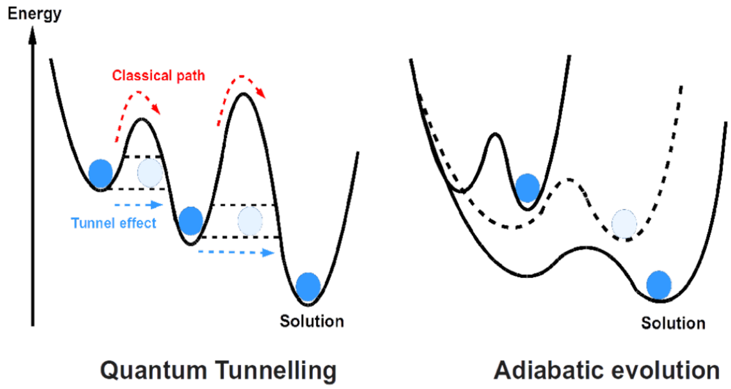


Figure 1: Illustration of the Adiabatic Evolution Process

Path integrals provide an intuitively appealing framework for expressing the thermal state of a system interpreted in quantum mechanics terms, and can be used as an expression for the probability amplitude of a system at the ground state. [6] We use Path Integrals to calculate the total energy of a solution across the solution space, whereby each path represents a route travelled.

The path integral terminology is listed in appendix I.

2.2 Explanation of Simulated Quantum Annealing

Path Integrals can be utilized to solve TSP problems by means of Path Integral Monte Carlo. This is done by expressing a path as a route, and performing monte carlo simulation of the system across a period of time.

Simulated Quantum Annealing makes use of Path Integral Monte Carlo by expressing a low temperature density operator (representing the state of the system,) as a product of high temperature density operators.

The target function we try to approximate with Simulated Quantum Annealing, is the Quantum Partition Function $Z(s)$. It incorporates the quantum effects into the simulation using the concept of path integrals. By mathematically splitting up the non-commuting terms using the Suzuki Trotter Expansion, we can approximate $Z(s)$.

SQA works by approximating the thermal density matrix from one state to another state in a path integral expansion. We map the classical system into an Ising formulation. Since each city is now represented by n bits, each of the paths represent a route configuration for touring a city.

To approximate $Z(s)$, we express $Z(s)$ as a sum over an exponential number of nonnegative terms. SQA then proceeds by discretizing the adiabatic path and using the Markov chain Monte Carlo method to sample from π at various values of the adiabatic parameter $s_1 \approx 0; \ldots\ s_{max} \approx 1$.

For each trial move, a bit is randomly swapped with another bit and we calculate the difference in action resulting from the change in the bit from the previous time step to the current step.

The algorithm is covered in the next section.

2.3 Simulated Quantum Annealing Algorithm

1. Convert Objective Function (min Cost Function for TSP) into a Distance Matrix
2. Set the annealing parameters (generic): $s = 1.0$, $\beta = 10$, Geometric Parameter: 0.999, Trotters (L) = 10, MC Sweeps per Trotter = 30
3. Generate N paths randomly for each of L Trotters and initialize all spins to -1 except for one randomly selected city in each N, for each of L trotters
4. Perform MC moves, while trying to maintain the Boltzmann Distribution
5. Perform Acceptance / Rejection and use for the next update
6. Update s by geometric parameter
7. Go back to (3), until we reach $S \to 0$
8. Repeat Experiment N times

2.4 Quantum-ness of SQA

The quantumness of the algorithm is incorporated in trying to calculate the partition function at each step. From the algorithm, this is reflected in step (4) and step (5). A deeper explanation is given.

*4. Perform MC moves, while trying to maintain the classical Boltzmann Distribution**

For this step, Monte carlo moves will incorporate an extra tanh term which modifies the . The more detailed breakdown is as follows:

1. For each Trotter path, randomly swap the path while maintaining the Hamiltonian cycle. Stitch together the route and calculate the energy of the route.
2. Approximate the Quantum Partition Function $Z(s)$ and use it to normalise each of the paths for that sweep
3. For each Trotter route, calculate the ratio of the route against $Z(s)$.

*5. Perform Acceptance / rejection and use for the next update**

1. Calculate the difference in action from each route. If the energy of the path is lower, accept the path and use it as for the next step. If the energy of the route is higher, accept the route with probability $\frac{1}{2nL} min(1, \frac{\pi(x')}{\pi(x)})$
2. If the route is rejected, resample from the routes and pick the one with the lowest route energy to partition function ratio.

2.5 Trotter Suzuki Decomposition

In a many particle system, the Hamiltonian H can be written as sum of kinetic and potential energy operators:

$$H = P + K$$

When we try to split up the terms of a Hamiltonian, we encounter a problem: Potential and Kinetic enrgy operators do not commute ($i.e. AB \neq BA$). We make use of the Trotter Suzuki Expansion to split up the non commuting terms, and obtain an error bounded by the order of L (the number of paths) we want to decompose the Hamiltonian into. [1]

To bound the error, we have to ensure that $\frac{\beta|H|}{L}$ needs to be sufficiently small or in other words, $L >> \beta|H|$. [2]

The first order of L is chosen for this simulation so that the partition function is $Z(s) = tr\ e^{A+B}$:

This gives us the mathematical expansion where:

$$Z(s) := e^{\frac{A}{L}} * e^{\frac{B}{L}} + \mathcal{O}(t)$$

where $A = \beta_s Hf$ and $B = -\beta(1-s)H_0$.

We can now make use of this result to compute the energy operators during the PIMC simulation.

2.6 Discussion of Results

The simulation was run over 100 iterations for Western Sahara [4] with 29 cities. The result is compared against the performance of PIMC versus a greedy algorithm, and the optimal solution.

From the results, we notice a few things:

1. SQA initially begins at an extremely high cost but very quickly improves and shows decreasing cost
2. At some point 0.8, the algorithm starts to perform better than the greedy algorithm, occasionally spiking above the cost of greedy
3. The algorithm then proceeds toward optimality, and it's best cost achieves a final margin of 5% from the optimal greedy solution

As the algorithm was run for a relatively short amount of time

The results are shown in Appendix II.

2.7 Hyperparameter Tuning

From further experimentation, it was found that:

- The choice of β highly affects the quality of solution. This in turn affects the number of paths we require (L)
- Rate of cooling s affects the quality of solution run-time (currently: 0.999)

Both β and s can be tuned to result in either a faster or more accurate algorithm. They will be further tested using metrics such as time-to-solution [3] in order to understand how they really affect the runtime and solution.

2.8 Future Works

Currently, the SQA algorithm is not easily generalizable to other types of TSP. This is because a challenge is posed in converting any further constraints into a penalty and applied onto the Ising Hamiltonian for any general class of problems under TSP (including Vehicle Routing, Time Window Problems, e.t.c)

There is also a need to find and tune for optimal hyperparameters such that we can trade off time-to-solution versus solution quality. Further measurements on the time-to-solution will be run in further tests on the algorithm.

Currently, the algorithm makes use of local bit flips to modify the route taken by any given path. This may not preserve feasibility in more complicated situations and is not truly quantum. Annealing on Quantum Hardware may be able to overcome this problem, where constraints can be embedded in ancilla bits.

The algorithm makes use of many computations across the nL space of bits and paths to calculate the path integral. This makes the algorithm highly unsuitable for scaling, as it runs in exponential time. An illustration of the complexities are as follows:

- Space Complexity $\mathcal{O}(n2 * L)$
- Time Complexity $\mathcal{O}(n * L * Steps) + O(2 * *N) \to O(2 * *N)$

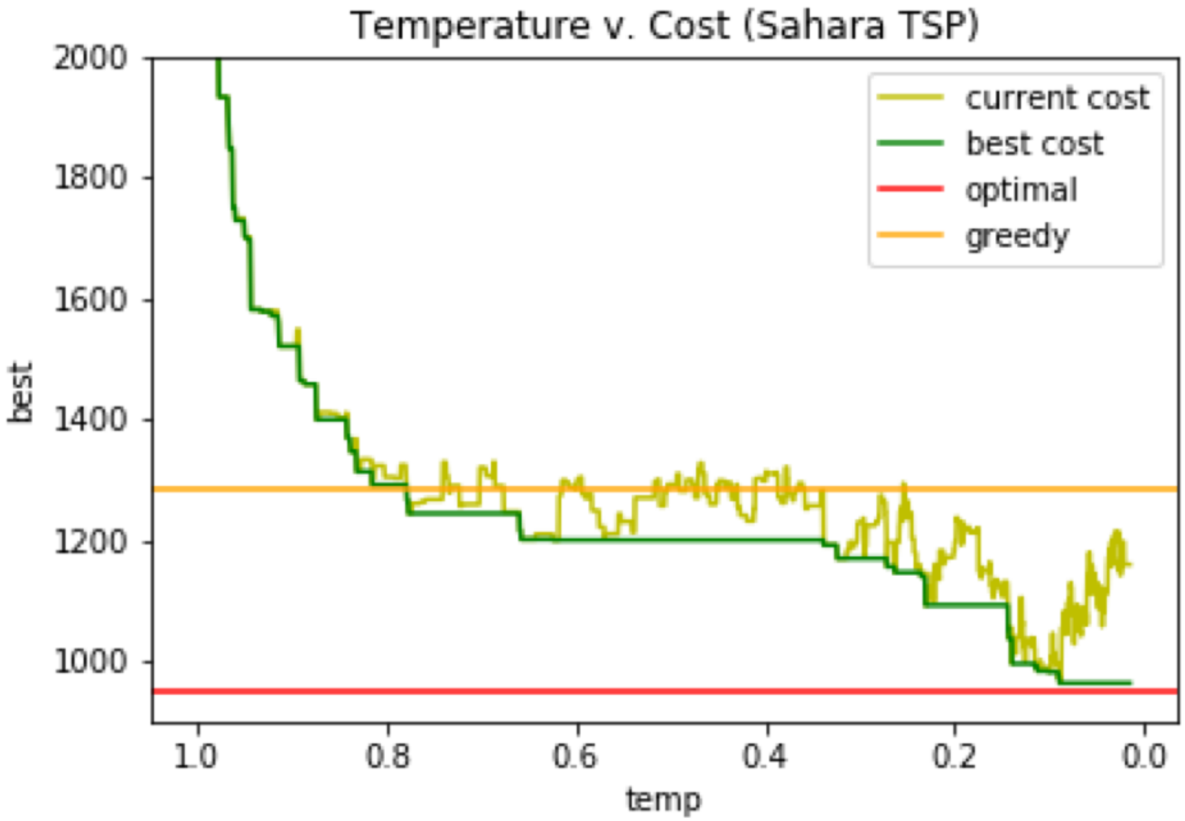
  These problems will potentially hinder the scalability of this algorithm. However, there is potential to incorporate some ideas from SQA such as system state perturbation into other classical algorithms in the future. This is an open research question and can be furthered in the future.

3 Appendix

3.1 Path Integral Terminology

Symbol	Explanation
L	Trotter Number (Number of paths explored)
N	Number of Bits used (# of paths)
Beta	Temperature of the System
S	Annealing Parameter

3.2 Results of PIMC Simulation



References

[1] Jens Bonin Alexey Filinov and Michael Bonitz. Introduction to path integral monte carlo. part i. 2008.

[2] Aram W. Harrow Elizabeth Crosson. Simulated quantum annealing can be exponentially faster than classical simulated annealing. *Proc of FOCS 2016*, 2016.

[3] Maliheh Aramon; Gili Rosenberg; Elisabetta Valiante; Toshiyuki Miyazawa; Hirotaka Tamura; Helmut G. Katzgraber. Physics-inspired optimization for quadratic unconstrained problems using a digital annealer. *Front. Phys. 7*, 48, 2019.

[4] University of Waterloo. National travelling salesman problems.

[5] Martin Fraas Sven Bachmann, Wojciech De Roeck. Adiabatic theorem for quantum spin systems. *Phys. Rev. Lett. 119*, 060201, 2017.

[6] Vvedensky Westbroek, King and Durr. User's guide to monte carlo methods for evaluating path integrals. *American Journal of Physics*, 86, 2018.