# Literature Review

Describe the chapter and the sections it has.

## 1 Annealing in Combinatorial Optimization

In this section we will discuss a class of metaheuristics that are inspired by the physical process of *annealing*. At the beginning, we will explain what it is. Then we will talk about multiple annealing metaheuristics that are relevant to this work. Finally, we will look at hardware named Digital Annealer.

### Annealing

Annealing is a heat treatment process of matter like glass or metal (Collins, 1921; Wu and Fan, 2020). It involves heating up the material and then carefully cooling it down at a controlled rate. By following that process, it is possible to achieve a stable crystal structure.

If, however, this rate is exceeded, the resulting crystal will have defects as only locally optimal structures were allowed to form (Kirkpatrick, Gelatt and Vecchi, 1983). The lack of equilibrium will make such crystal metastable.

### Simulated Annealing

Simulated annealing (SA) is a metaheuristic that uses the concept of temperature to approximate global optimum of a combinatorial optimization problem (Kirkpatrick, Gelatt and Vecchi, 1983; Bertsimas and Tsitsiklis, 1993) The temperature (*T*) is a variable that controls the likelihood of making locally non-optimal uphill moves. Just as in physical annealing, we start the optimization with a high temperature and decrease it at a controlled rate. If that rate is low enough, we are guaranteed to find a global optimum (Liang, Cheng and Lin, 2014). However, it is possible to find good approximations of the optima significantly faster at higher cooling rates.

The temperature concept allows the algorithm to explore the search space and find the optimal solution (Poole and Mackworth, 2017). The SA starts with a high temperature. At that point the algorithm acts like random walk, choosing random moves with high probability, which prevents us from being stuck at local optima – this is when exploration happens. At the end of the SA, when the temperature is low, the algorithm prioritizes advantageous moves. It acts like greedy algorithm and tries to find the best solution. As we shift from higher to lower temperatures, we gradually shift from exploring the search space to find the optimum.

Let *c* be current node, *n* be next node (neighbour of node *c*) and *evaluation(x)* be the function that evaluates the energy of the node *x*. Then the energy transition (*ΔE*) can be defined by the following formula:

*ΔE = evaluation(n) - evaluation(c)*

*ΔE* and *T* are the variables that will affect the probability of choosing node *n*, where *ΔE* is an energy transition (negative when minimising) that the proposed node will introduce, and *T* is controlling how *ΔE* influences the probability of choosing this node (Johnson *et al.*, 1989).

The *acceptance probability function* is the function that maps current node, proposed node, and Temperature to a probability. There are many functions that suit this purpose, the original being an exponential function (Kirkpatrick, Gelatt and Vecchi, 1983).

When maximizing, the negative sign should be removed from the formula.

If that acceptance probability is higher than a uniformly generated random number, then node *c* is rejected in favour of node *n*. However, when (downhill movement), we will always choose the proposed node (Kirkpatrick, Gelatt and Vecchi, 1983).

Pseudocode of SA (Johnson *et al.*, 1989) is given by:

c ← initial node

**while** T > threshold

n ← generate next node(c)

ΔE ← evaluate(n) – evaluate(c)

**if** ΔE ≤ 0 t

c ← n

**else if** P(c, n, T) ≥ random(0, 1)

c ← n

**end**

T ← cooling(T)

**end**

### Quantum Annealing

Quantum annealing (QA) is a metaheuristic that uses quantum-mechanical fluctuations to find a global minimum of a function (Ohzeki and Nishimori, 2010; Tanaka and Tamura, 2012). In QA we start with a strong quantum field and gradually decrease its strength (quantum fluctuation) to find a ground state that is optimal. This is analogous to SA, where we decreased the temperature (thermal fluctuation) to achieve the same result.

QA begins in a uniform superposition of all possible states, but as we decrease the strength of quantum field, we approach a single solution state (Farhi *et al.*, 2001).

In SA neighbouring nodes are located 1 move away for the current node. In QA the neighbouring nodes are in a certain range. This range is defined by the strength of the quantum field, slowly ceasing in the process of annealing. Transition from one state to another can happen within that range. If there is a short wall between current state, which happens to be a local minimum, and the global minimum, SA algorithm will need to make a series of disadvantageous moves to reach a better solution or in other words *“climb”* over the wall. This is avoided in QA as there is a capacity to tunnel through short walls if the quantum field is strong enough, making QA faster compared to SA when dealing with search spaces with many thin barriers (Crosson and Harrow, 2016). This effect is called *quantum tunnelling*. Comparison of thermal hopping and quantum tunnelling can be seen on Figure 1.

Diagram

Description automatically generated

Figure 1. Insert text

The fundamental feature and the main benefit of QA is that it rapidly samples a wide range of configurations when exploring the energy landscape (Higham and Bedford, 2021). Quantum Annealer has been developed and commercialised by D-wave (Johnson *et al.*, 2011) and is currently used to solve real-world problems (D-Wave Systems, 2021).

### Digital Annealer

Digital Annealer (DA) is quantum-inspired custom CMOS hardware developed by Fujitsu Laboratories (Aramon *et al.*, 2019). Its algorithm is based on SA, but has 3 major differences:

1. Instead of considering a single neighbouring node and choosing it if the acceptance probability is higher than a uniform random number, DA considers all neighbouring nodes in parallel. If more than a single node was accepted, 1 of them will be chosen uniformly at random to move into. This is advantageous as in SA if the node was rejected, we need to generate and then consider a new one from the same position, which takes time. In DA, however, all the nodes are considered in parallel and the acceptance probability per “turn” is therefore higher.
2. DA uses *dynamic offset*, which means that if no nodes were accepted, the next acceptance probability will be artificially increased to help the algorithm overcome narrow barriers.
3. All runs begin with the same random state to save time (in SA that runs in parallel the initial states are generated individually for each of the runs). Otherwise, it would need to evaluate the new initial node and all the new neighbouring nodes for each run.

These differences can be observed in DA pseudocode (Aramon *et al.*, 2019):

initial state ← random state

**for** each run

current state ← initial state

Eoffset ← 0

**for** each step

update temperature if needed

**for** each neighbour *(in parallel)*

calculate ∆Eneighbour

consider neighbour using ∆Eneighbour - ∆Eoffset

if neighbour accepted, record

**end**

**if** accepted neighbours ≥ 1

choose one accepted neighbour uniformly at random

update current state and the energy landscape *(in parallel)*

Eoffset ← 0

**else**

increase Eoffset

**end**

**end**

**end**

Although it is hard to make comparisons as the technology is quickly developing, it has been shown that DA can be faster and produce more precise solutions compared to D-Wave 2000Q (penultimate generation of D-Wave’s QA devices) in certain scenarios (Ohzeki *et al.*, 2019). DA’s are used in industry and science (Fujitsu, no date; Snelling *et al.*, 2020)

Currently QA devices are expensive and difficult to run as the technology is still under development (Şeker *et al.*, 2020). D-Wave 2000Q also suffers from qubit noise that results in lower precision (Katzgraber and Novotny, 2018), and sparse connectivity between spins that makes it less effective (Hamerly *et al.*, 2018). These problems are actively worked on and some of them are addressed in more recent QA devices (Dattani, Szalay and Chancellor, 2019). As QA technology develops, it will likely outperform DA largely due to its quantum parallelism (Boyd, 2018).

## 2 Quadratic Binary Unconstrained Optimization

Both DA and QA accept QUBO form

# 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.

Bertsimas, D. and Tsitsiklis, J. (1993) ‘Simulated Annealing’, *Statistical Science*, 8(1), pp. 10–15.

Boyd, J. (2018) ‘Fujitsu’s CMOS Digital Annealer Produces Quantum Computer Speeds’, *IEEE Spectrum*, 28 May. Available at: https://spectrum.ieee.org/ieee-courses-on-digital-transformation (Accessed: 12 November 2021).

Collins, E. F. (1921) ‘ELECTRICALLY HEATED GLASS ANNEALING LEHR1’, *Journal of the American Ceramic Society*, 4(5), pp. 335–349. doi: 10.1111/J.1151-2916.1921.TB18664.X.

Crosson, E. and Harrow, A. W. (2016) ‘Simulated Quantum Annealing Can Be Exponentially Faster than Classical Simulated Annealing’.

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

Dattani, N., Szalay, S. and Chancellor, N. (2019) ‘Pegasus: The second connectivity graph for large-scale quantum annealing hardware’.

Farhi, E. *et al.* (2001) ‘A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem’.

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).

Hamerly, R. *et al.* (2018) ‘Scaling advantages of all-to-all connectivity in physical annealers: the Coherent Ising Machine vs. D-Wave 2000Q’.

Higham, C. F. and Bedford, A. (2021) ‘Quantum Deep Learning: Sampling Neural Nets with a Quantum Annealer’. Available at: https://arxiv.org/abs/2107.08710v1 (Accessed: 11 November 2021).

Johnson, D. S. *et al.* (1989) ‘Operations Research Society of America This content downloaded from 128’, 37(6).

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

Katzgraber, H. G. and Novotny, M. A. (2018) ‘How Small-World Interactions Can Lead to Improved Quantum Annealer Designs’, *Physical Review Applied*, 10(5), p. 054004. doi: 10.1103/PHYSREVAPPLIED.10.054004/FIGURES/13/MEDIUM.

Kirkpatrick, S., Gelatt, ; C D and Vecchi, ; M P (1983) ‘Optimization by Simulated Annealing’, *New Series*, 220(4598), pp. 671–680.

Liang, F., Cheng, Y. and Lin, G. (2014) ‘Simulated Stochastic Approximation Annealing for Global Optimization With a Square-Root Cooling Schedule’, *https://doi.org/10.1080/01621459.2013.872993*, 109(506), pp. 847–863. doi: 10.1080/01621459.2013.872993.

Ohzeki, M. *et al.* (2019) ‘Control of Automated Guided Vehicles Without Collision by Quantum Annealer and Digital Devices’, *Frontiers in Computer Science*, 1, p. 9. doi: 10.3389/FCOMP.2019.00009/BIBTEX.

Ohzeki, M. and Nishimori, H. (2010) ‘Quantum annealing: An introduction and new developments’.

Poole, D. L. and Mackworth, A. K. (2017) ‘Artificial Intelligence: Foundations of Computational Agents’, pp. 136–137. doi: 10.1017/9781108164085.

Şeker, O. *et al.* (2020) ‘Digital Annealer for quadratic unconstrained binary optimization: a comparative performance analysis’. Available at: http://arxiv.org/abs/2012.12264 (Accessed: 3 October 2021).

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

Tanaka, S. and Tamura, R. (2012) ‘QUANTUM ANNEALING AND QUANTUM FLUCTUATION EFFECT IN FRUSTRATED ISING SYSTEMS’.

Wu, H. and Fan, G. (2020) ‘An overview of tailoring strain delocalization for strength-ductility synergy’, *Progress in Materials Science*, 113, p. 100675. doi: 10.1016/J.PMATSCI.2020.100675.