

QUANTUM ANNEALING OPTIMIZATION OF A HEURISTIC SURROGATE MODEL FOR PWR FUEL LOADING

Andy Whyte¹ and Geoff Parks¹

¹University of Cambridge
Department of Engineering, Trumpington Street, Cambridge CB2 1PZ

ajw287@cam.ac.uk, gtp10@cam.ac.uk

ABSTRACT

An efficient fuel arrangement must be generated by PWR operators every 6–18 months. This complex problem has been extensively researched with two broad approaches, heuristic and stochastic methods, becoming accepted. This initial study qualitatively introduces the concept of encoding full-core PWR fuel loading patterns in a form suitable for quantum annealing. The concepts of adiabatic quantum computers and quantum annealing are introduced, and a surrogate model encoding of a set of heuristics for loading pattern design produced in a form suitable for use in present-day quantum annealers. The simulated results show significant similarity to benchmark loading patterns.

KEYWORDS: quantum computing, optimization, nuclear fuel management, loading pattern, PWR

1. INTRODUCTION

A central task of in-core PWR fuel management is the creation of a loading pattern (LP). It is important to operators and regulators to ensure that the choice of LP meets constraints derived from safety, operational, and other conditions, before carrying out the detailed design studies required by licensing authorities. At the same time, economic factors push operators to find fuel arrangements with reduced power peaking factor (PPF), longer cycle time and lower enrichments. This task has long been recognised as an important part of fuel cycle optimization [1][2]. However, the combined attributes of PWR fuel LPs (high dimensionality, high nonlinearity, lack of direct derivative information, and multiple minima) describe an extremely difficult optimization problem [3]. The high dimensionality of the inputs has been recognised as a particular problem for some time: “The main conclusion of this work is that the basic challenge of the reload configuration design is due to the extremely large search space.” [4]

The combinatorial explosion of LPs means that two broad optimization approaches have been shown to be successful: heuristic search methods and stochastic optimization. Heuristic search aims to reduce the search space to a manageable size by the creation of a set of heuristic, knowledge-based rules, enabling a meaningful search for the global optimum [4]. Stochastic search applies iterative evaluations of potential solutions to balance local improvements in a design with larger changes to escape from basins of attraction and avoid being trapped in local minima [3].

In this paper, heuristic rules, similar to those used in [4], will be encoded into a surrogate model suitable for use by a quantum annealer (QA) [5], a present-day quantum computer, to generate

PWR LPs. QAs use a trick from quantum adiabatic theory, which allows a simple Hamiltonian to be converted into a complex one, while keeping the system in the minimum energy state.

In 1982, Feynman [6] suggested that the inherent complexity of quantum mechanics might be an efficient way to solve complex computational problems. This initiated the field of quantum computing, and, despite significant research effort, it is only in recent years that quantum computers have become available commercially. The following section describes how adiabatic theory permits the discovery of the global optimum of a system. This has been utilised in adiabatic quantum computing and implemented to some extent in QAs.

1.1. Adiabatic Quantum Computers and Quantum Annealing

Adiabatic quantum theory describes a technique for global optimization. The theory applies to a Hamiltonian that evolves in time from a simple system to a complex one, as described by:

$$H(t) = (1 - t/T)H_0 + (t/T)H_1 \quad (1)$$

H_0 describes a simple Hamiltonian shown in Eq. (2), and H_1 is an implementation of the problem that we are looking to solve. In order for our problem to be solvable by an adiabatic system, the problem must be expressed in terms of the classical Ising model, of the form shown in Eq. (3).

$$H_0 = - \sum_{i,j} \sigma_i^X \sigma_j^X \quad (2)$$

$$H_1 = - \sum_{\langle i,j \rangle} \mathcal{J}_{ij} \sigma_i^Z \sigma_j^Z - \mu \sum_i h_i \sigma_i^Z \quad (3)$$

where: H is a Hamiltonian, a function representing the total energy of the system; $\langle i, j \rangle$ signifies connected qubits i and j ; \mathcal{J} is an exchange constant, a matrix of strengths of connections between qubits attracting them to similar or opposite states; σ signifies logical qubits, which can have values -1 or $+1$ upon measurement; and h is a transverse field, an external field strength applied to each qubit that biases its state.

Quantum adiabatic theory states that, if a system like the one above is in its ground state and is evolved over time, then the system remains in its ground state *unless the rate of energy put into the system is enough to push it into a new state*. While this sounds trivial, the implications are important. So long as a ‘speed limit’ for t is not exceeded, then the system will evolve from H_0 to H_1 while remaining in the global minimum energy state. Thus, it is possible to find the global optimum of a problem. However, for an adiabatic quantum computer to optimize an objective function, it must be expressed in terms of the Ising model Hamiltonian, Eq. (3).

Furthermore, existing quantum computers operate at around 10 mK above absolute zero, which means that, although the system is expected to be in its ground state initially, this is not guaranteed. Finally, the ‘speed limit’ is given by Eq. (4), where Δ is the difference of energy between the ground state and other states. Unfortunately, it is more difficult to find the difference between the global minimum and the second best minimum than to find the global minimum in the first place.

$$t_{limit} \approx \frac{1}{(\min(\Delta) \times t)^2} \quad (4)$$

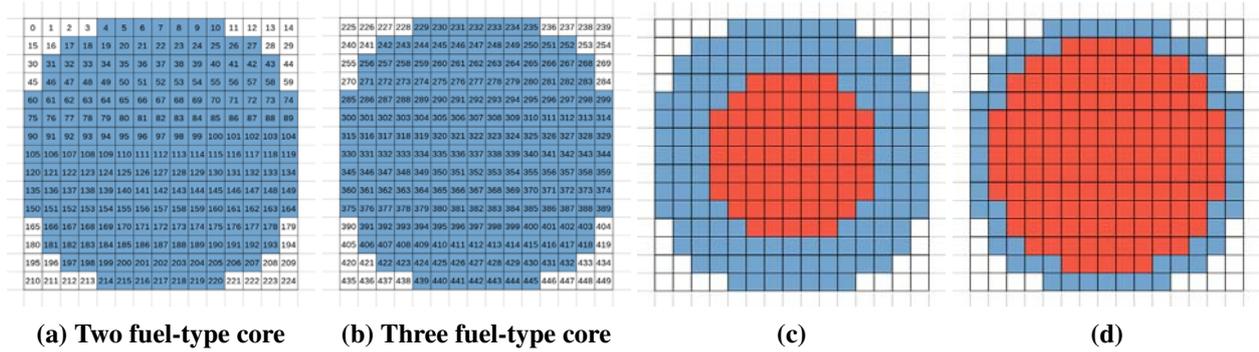


Figure 1: Core representation with qubits on a regular grid (a, b) and examples of transverse fields, h , defining inner and outer regions (c, d).

This leads to the implemented solution: quantum annealing. The system is the same, but the system evolves from H_0 to H_1 at a practical speed, ignoring the speed limit. The process is executed many times and the minimum solution found is taken to be the optimum. This allows the problem to be solved in a realistic amount of time, while still identifying the global optimum, assuming that at least one execution has evolved from the ground state of H_0 to the ground state of H_1 .

2. METHODOLOGY

A PWR reload core with three different fuel enrichments is used as an example problem. The following assumptions have been made to simplify the problem:

1. Burnable poisons (BPs) are neglected.
2. Control rods and instrumentation are considered to be withdrawn.
3. Only the start-up core made up of fresh fuel is considered.

Based on the software stack in Fig. 2, the problem of fuel loading must be encoded into a suitable Ising model. In the first experiment, a two fuel-type reload core is considered. Each assembly in the core is assigned a logical qubit, as shown in Fig. 1a. The qubits are named numerically according to the image for regularity in the code, but only qubits that represent an actual assembly (blue in the figure) are assigned. This means that only 193 logical cubits are required to describe the system. For the three fuel-type experiment, two qubits are assigned for each assembly, as shown in Fig. 1b, which enables the assignment of three meaningful states in an Ising-type model. In order to encode the kind of knowledge-based rules used by Galperin [4], the connectivity of the qubits must be set up and interactions are defined by setting values of the transverse field, h , and the exchange constant, \mathcal{J} , for every connected i and j . Galperin's example heuristic rules are used as the basis for the rule set used in this study. They are as follows:

1. Fresh fuel assemblies should not be loaded into peripheral core positions.
2. Twice-burnt fuel assemblies should not be loaded into innermost core positions.
3. Two fresh fuel assemblies should not be loaded into adjacent positions unless one of them is adjacent to a peripheral one.

- Two twice-burnt assemblies with high accumulated burnup values should not be loaded into adjacent positions.

To apply this rule set to the core design tasks in this paper, the cycle number (burnup state) of the fuel is equated to enrichment, so the highest enrichment fuel is equated to fresh fuel in a batch system, and the lowest enrichment fuel is equated to twice-burnt fuel. Rule 3 is then applied as is, while Rules 1 and 2 are inverted compared to [4], due to a desire to be consistent with the reload core design used in the BEAVRS benchmark [7].

In the Ising model, the transverse field, h , is a blocking force that overrides the connectivity of qubits. In order to equate it to our heuristic rules, it is defined using one of the patterns shown in Figs. 1c and 1d. This allows qubits to have a particular affinity for one type of spin based on position in a particular region (defined radially here). This is used to enforce Rule 1. The exchange constant, \mathcal{J} , is then designed to ensure that the system obeys the Rule 3. This is achieved by creating a connection between each qubit and the adjacent qubits. The extension to 3 batches also connects qubits that are adjacent to the edges of the regions defined in the transverse field.

In the next step, the connections defined in the logical qubits of the Ising model description of the problem are translated to the actual architecture of a target QA. The Ising description can connect any number of qubits together in any arrangement, and actual QAs must emulate this functionality. This process is called the *minor embedding*. A common arrangement of connections for physical qubits is called a *chimera graph*. The box labelled ‘Minor graph embedding’ in Fig. 2 shows a single group of eight qubits from a chimera graph: the groups are locally interconnected and each qubit is connected to two adjacent groups. In order to emulate a logical qubit’s connections, multiple actual qubits are strongly coupled together until the correct connectivity is achieved.

3. RESULTS

3.1. Patterns from Simulated Quantum Annealers

The results in this section were created using simulated quantum annealing (SQA) [8] on conventional hardware. Nevertheless, the algorithm used attempts to model the quantum annealing process accurately, and the software used is easily converted for use on an existing QA. The first

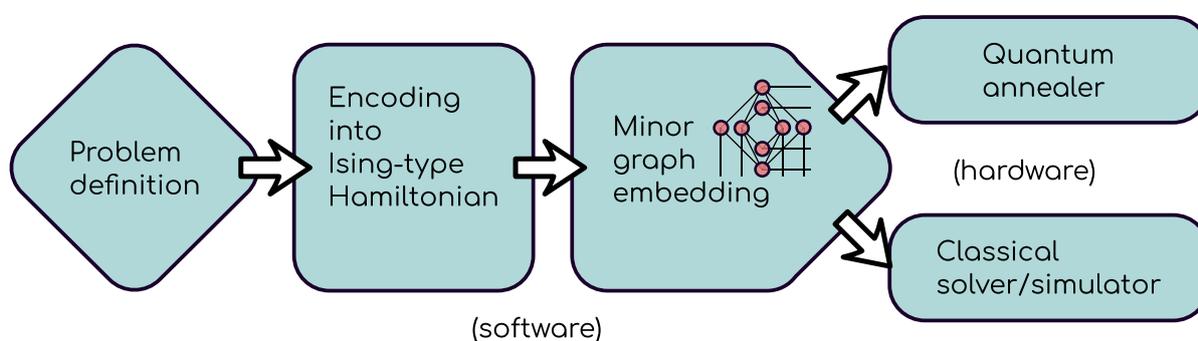


Figure 2: A diagram of showing the operation stack of a quantum annealer (based on [5]).

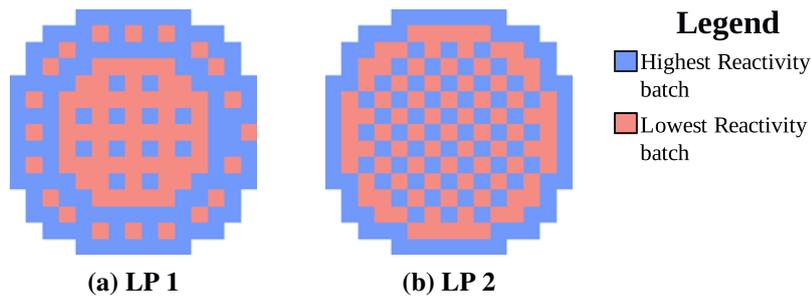


Figure 3: LP designs using SQA for a two fuel-type PWR.

result is that it is possible to transcribe a set of rules for PWR LP optimization into a real-world quantum computer. The two fuel-type problem has a minor embedding that fits within a 16×16 chimera graph, as this is the largest quantum computer at the time of writing. The three fuel-type problem has a minor embedding that fits onto a 25×25 chimera graph, which is expected to be the largest implementation at the time of publication [9].

Two fuel-type LPs optimized using SQA are shown in Fig. 3. The Fig. 3a LP was found when there is a bias for a large radial region with adjacent assemblies experiencing an inverting force. The radial region is smaller for the Fig. 3b LP, and an even split of the fuel types is achieved. Fig. 4 shows LPs found in the three fuel-type case. Rules 1 and 2 create the outer ‘blanket’ region, while Rule 3 generates the ‘checker-boarding’ effect. A start-up core LP from the BEAVRS benchmark [7] is shown in Fig. 4c, for comparison; in this, the highest w/o U235 correlates well with fresh fuel, the next w/o U235 with the once-burnt batch and the lowest w/o U235 with the twice-burnt batch. Note that the BEAVRS start-up core does not exactly conform to Galperin’s rules.

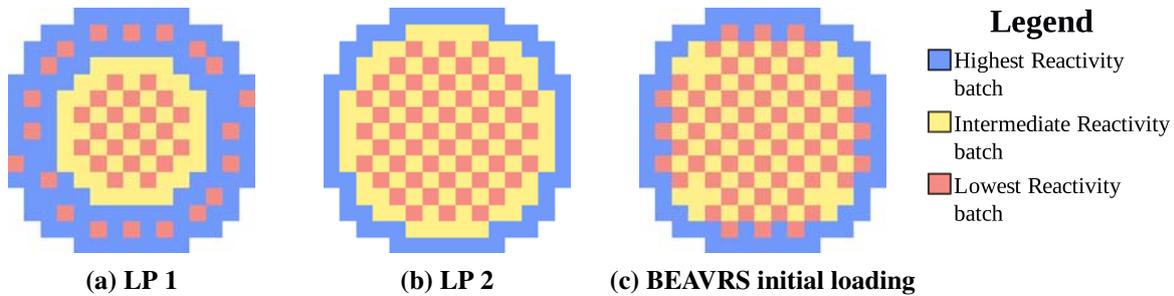


Figure 4: LP designs using SQA for a three fuel-type PWR compared with the initial core LP in the BEAVRS benchmark [7].

Since the problems posed in this paper are all symmetrical, the global optima must also be symmetrical. A simple metric for the effectiveness of the algorithm is needed to establish proximity to the global optimum. This is achieved by calculating the degree of quadrant symmetry, by comparing each quadrant rotated appropriately as follows:

$$R = \frac{\sum_{N=2}^4 \sum_{x=0}^{w/2} \sum_{y=0}^{h/2} Q_{x,y}^1 - Q_{x,y}^N}{w \times h} \quad (5)$$

where: R is a measurement of quadrant rotational similarity; Q^{1-4} are the rotated quadrants of the output ‘core’ LP; x is the horizontal position coordinate; y is the vertical position coordinate; w is the model width; and h is the model height.

This calculation produces a value in the range 0–1, with 0.03125 per non-similar output for 15×15 arrays such as here. Although this does not effectively rank the solutions and has other limitations, it gives an easy-to-calculate measurement of proximity to the global optimum of the heuristic model, based on the assumption that all solutions are within a few pixels of the global optimum. This can be checked by visual inspection of the results (examples shown in Figs. 3 and 4). Table 1 shows the results of this calculation for 30 runs for each of the configurations. Readers should note that this code is only running on a single core and has not been optimized.

Table 1: Simulation quantum annealing data (mean of 30 runs), values to 4 s.f.

Cases	Batches	Quadrant Similarity (R)		Run Time / s	
		mean	σ	mean	σ
Fig. 1c	2	0.1875	0.088 60	547.1	9.510
Fig. 1d	2	0.3521	0.076 03	553.7	22.35
Fig. 1c	3	0.098 61	0.034 14	787.2	20.23
Fig. 1d	3	0.0	0.0	853.36	65.138

3.2. Validating Results in a Nodal Code Simulation

Although visual inspection of the results from the SQA process shows that the designs have a regularity that would be expected of good results, the system is only optimizing heuristic models developed according to a set of rules. To check whether these models have optimal solutions with favourable characteristics in a simulated PWR, they were analysed using a neutronics simulation. LPs from Figs. 3 and 4 were selected and simulated using a WIMS/PANTHER model of the BEAVRS benchmark [7][10]. In order to evaluate the LPs, octant core symmetry is applied, which is considered acceptable in this case since the solutions exactly (or nearly) exhibit this symmetry.

Table 2 shows some indicative statistics for a number of simulated cores. The cores are labelled by the Figs. 3/4 for two and three fuel-types, with Fig. 4c representing a simplified version of the BEAVRS initial LP and the core labelled “2.6 w/o U235” being a uniformly loaded core with an intermediate enrichment. Although the latter does not have favourable statistics, it is included for comparison with the optimized LPs.

The WIMS/PANTHER BEAVRS model used in this study has shown good agreement with the benchmark data [10]. However, since the heuristic surrogate model used here does not attempt to consider BPs, a simplified version of the initial LP was generated (Fig. 4c) for comparison. Although the PANTHER results for Fig. 4c show significant divergence from actual BEAVRS results, the differences are most pronounced at beginning of cycle (BOC). This implies that these discrepancies are due to the removal of BPs, which act to reduce the initial reactivity and the PPF.

The results for three fuel-type LPs are promising. The SQA solution for three fuel-types and a smaller radial region (Fig. 4b) has improved PPF values compared to the Fig. 4c design, albeit at

a shorter cycle duration, as indicated by the end of cycle (EOC) boron concentration. While the two fuel-type LPs are less optimal than the three fuel-type ones, they are superior to the single fuel-type core (2.6 w/o U235) in all aspects except EOC PPF.

Table 2: Results for LPs burned up to 13593 MWd/tonne in PANTHER

Core	BOC PPF	EOC PPF	BOC boron /ppm	EOC boron /ppm
Fig. 3a	1.854	1.299	1816	100
Fig. 3b	2.849	1.247	1682	132
Fig. 4a	1.760	1.127	1345	-41
Fig. 4b	1.760	1.127	1345	-41
Fig. 4c	1.866	1.188	1518	71
BEAVRS (model)	1.462	1.173	961	-32
BEAVRS (actual)	1.438	-	975	0
2.6 w/o U235	2.158	1.163	1888	213

4. DISCUSSION

This paper presents promising results from an initial study of the suitability of present-day and near-present-day quantum computers for optimization of fuel arrangements in nuclear power plants. The exciting conclusion is that, using this technique, a shortcut around the combinatorial explosion might be found.

The two fuel-type implementation gives results that can be seen by inspection to adhere to the simplified rules extracted from [4]. The three fuel-type solutions shown in Fig. 4 compare well with the arrangement of different fuels in a start-up core from the BEAVRS benchmark (Fig. 4c). Some differences can be noted as coming from Rule 3 used to guide the optimization. Note that the annealer is not using symmetry to generate these LPs; symmetry arises from the minimum energy solution to the surrogate model. The generated LPs were then simulated using a nodal reactor physics code to evaluate their actual performance. The results show that the LPs found for three fuel-types are promising when compared with the start-up core designed for a commercial reactor.

This study is by no means a complete investigation of fuel loading optimization with quantum annealing techniques. Future work should look to encode physical aspects of the neutron transport into the problem. Potential exists to express a fission matrix in the Ising model form. The authors feel that this technology shows significant future promise for solving complex nuclear engineering problems. While this initial study gives promising conceptual results, limitations are acknowledged. These include: the list of assumptions on page 2; that the heuristics encoded here, though knowledge-based, are much simpler than the actual interactions of components in a nuclear reactor; the specific histories of each assembly are not considered, as is normal for a multi-fuel-type core. Although the encoding does not enforce that equal numbers of assemblies of each fuel-type (batch) are used, careful design of the \mathcal{J} and h fields can ensure that the minimum energy Hamiltonian is not biased towards a particular fuel category.

5. CONCLUSIONS

This paper has described an example of encoding relatively simple rule sets in order to optimize a full-core LP at the assembly level using a quantum simulated annealer. Although there is a great deal of work still to be done to create a useful tool for PWR fuel management, the basic building blocks of the technique have been presented and the results show promise. Further work should look at ways to transfer real physics connectivities into the Ising simulation, as well as ways to transfer burnup histories and gradients into the fuel.

The technology of adiabatic quantum computing and quantum annealing have been introduced, and an embodied problem solution given. A mapping of heuristic rules into an Ising model is created, and the steps to generate the actual embodied solution are carried out: the model is converted from the Ising model to a minor embedding. This represents a novel approach to LP generation.

Code used in this paper is available for audit, reproducibility and derived works. A copy can be obtained from the repository under the permissive MIT license:

https://ajw287@bitbucket.org/ajw287/quantum_pwr_reload.git

ACKNOWLEDGEMENTS

The authors are grateful to the contributors to software used [11][12] as well as to Chris Mangnall, Daniel Rowlands and Bruno Louriero for discussion and pointing us in the direction of QA.

REFERENCES

- [1] B. Naft and A. Sesonske. “Pressurized Water Reactor Optimal Fuel Management.” *Nuclear Technology*, **volume 14**, pp. 123–132 (1972).
- [2] R. K. Haling. “Operating Strategy for Maintaining an Optimum Power Distribution Throughout Life.” *UNT Libraries Government Documents Department* (1964).
- [3] G. T. Parks. “An Intelligent Stochastic Optimization Routine for Nuclear Fuel Cycle Design.” *Nuclear Technology*, **volume 89**, pp. 233–246 (1990).
- [4] A. Galperin. “Exploration of the Search Space of the In-Core Fuel Management Problem by Knowledge-Based Techniques.” *Nuclear Science & Engineering*, **volume 119**(2), pp. 144–152 (1995).
- [5] M. Fingerhuth, T. Babej, and P. Wittek. “Open source software in quantum computing.” *PLoS ONE*, **volume 13**(12)(e0208561) (2018).
- [6] R. P. Feynman. “Simulating physics with computers.” *International Journal of Theoretical Physics*, **volume 21**(6), pp. 467–488 (1982).
- [7] N. Horelik, B. Herman, B. Forget, and K. Smith. “Benchmark for Evaluation and Validation of Reactor Simulations (BEAVRS), v1.0.1.” In *Proc. Int. Conf. Mathematics and Computational Methods Applied to Nuclear Science & Engineering* (2013).
- [8] A. Condello and contributors. <https://github.com/dwavesystems/dimod> (2018).
- [9] “Press Release: D-Wave Previews Next-Generation Quantum Computing Platform.” <https://www.dwavesys.com/press-releases/d-wave-previews-next-generation-quantum-computing-platform> (2019).
- [10] R. Harrison, G. Startin, B. Lindley, D. Powney, G. Parks, and P. Hutt. “Validation of WIMS/PANTHER PWR fuel reactivity depletion using the BEAVRS benchmark flux map data.” In *Proceedings of PHYSOR 2016*, volume 5, pp. 2833–2846 (2016).
- [11] G. van Rossum. “Python tutorial, Technical Report CS-R9526.” Technical report, Centrum voor Wiskunde en Informatica (CWI), Amsterdam (1995).
- [12] A. Hagberg, P. Swart, and D. S Chult. “Exploring network structure, dynamics, and function using NetworkX.” Technical Report LA-UR-08-05495, Los Alamos National Laboratory, Los Alamos, NM (2008).