Quantum

Optimization for

in-core fuel management

A blue and white machine

Description automatically generated with medium confidence

Fuel loaded into the core of Vogtle Unit 2 in Waynesboro, GA

A screenshot of a grid

Description automatically generated

The reactor core contains 37 cells, each of which must be filled with exactly one fuel assembly—either fresh or once-burned [1]. Same-type fuel assemblies (especially fresh-fresh) must not be placed adjacent to each other, to prevent localized power peaks and ensure neutron flux uniformity [2]. The core configuration must aim to maximize reactor power while satisfying all physical and safety constraints.

The total number of fresh and once-burned assemblies in the core must exactly match the available fuel inventory.

To ensure the physical feasibility and operational safety of the reactor core, the following constraints must be satisfied:

**1-Single Assembly per Cell**  
Each of the 37 core positions (cells) must be occupied by exactly one fuel assembly. No cell may remain empty or contain multiple assemblies

**2-No Same-Type Adjacency**  
Placing same-type fuel assemblies—especially fresh-fresh—adjacent to one another can result in localized power peaks and non-uniform neutron flux

**3-** **Fuel Inventory Consistency**  
The total number of fresh and once-burned fuel assemblies placed in the core must exactly match the available inventory.

**Solution& Pipeline**

We will try to solve this this problem by Quantum annealing we will Follow this

**Pipeline**

**Math Module QUBO Format Algorithm Solution**

**The Evaluation will be between the time of Brute force search and Quantum annealing**

**1st Step of Pipeline Math module**

**Parameters**

**N:** Total number of core cells

**i,J:** Indices that label individual cells. j is used only when talking about a cell that is adjacent to iii.

|  |
| --- |
|  |

**K: 0** burnt

**K: 1** fresh fuel

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1st Constraints (**Single Assembly per Cell**)

A close-up of a number

Description automatically generated

Constrain (**Fuel Inventory Consistency**)

A black and white math equation

Description automatically generated with medium confidence**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Objective Function**

A black symbols on a white background

Description automatically generated



We seek to minimize that bill—ideally driving it to 0 when the lattice and the specified inventories make a perfect alternation possible.

**2nd step of Pipeline Building the QUBO model**

The QUBO module will be built by jij Library [3]

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**3rd Step of Algorithm**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Quantum annealing** is a specialized quantum computing technique focused on finding the optimal solution to optimization problems. It achieves this by leveraging quantum mechanics to explore the solution space more efficiently than classical methods

**4th Step Solution**

We want to test our approach on 2 cases,

**Cases One**

Fresh. = 15, burn =22

**OpenJij best raw energy: 3.00**

A screenshot of a blue screen

Description automatically generated

**Case two**

Fresh. = 16, burn =21

**OpenJij best raw energy: 0.00**

A screenshot of a blue screen

Description automatically generated

**Evaluation**

Compare Between the Classical algorithm and quantum annealing

**Brute Force Search**

A literal brute-force search would have to examine **every possible placement  
of 10 burnt + 27 fresh assemblies in 37 cells**

A black text on a white background

Description automatically generated

**Quantum Annealing**

For a configuration with 16 burnt and 21 fresh assemblies: **Sampling: 10,000 reads**

**Runtime: 15.375 seconds**

Quantum annealing significantly reduces computation time by sampling from a low-energy solution space rather than exhaustively exploring all possibilities.

**Reference & Code**

**[1] Galperin A, 1995. Nuclear Science and Engineering 119, 144–152**.

[2] **Usmanov, S. R., Salakhov, G. V., Bozhedarov, A. A., Kiktenko, E. O., & Fedorov, A. K.** (2023). Quantum and quantum-inspired optimization for an in-core fuel management problem. Russian Quantum Center, Skolkovo, Moscow. arXiv:2308.13348 [quant-ph].

[3]Jij Inc. (2025). OpenJij: A framework for Ising and QUBO optimization via simulated annealing (Version 0.10.7) [Computer software]. <https://www.openjij.org>

Link for the:

[Github Code](https://github.com/MAmr9UH/Nuclear-Reactor-Core-Optimization-Quantum-Computing/tree/main)