Quantum and Quantum-Inspired Algorithm For Wireless Communications

Week 10

Oskar Iwaniuk, Brian Chacon, Shehzad Sultan
Mentored by Dr. Minsung Kim & Dr. Byungjun Kim

Brian
Rising Sophomore

Oskar Iwaniuk
Rising Sophomore

Shehzad Sultan
B.S Computer Science

Minsung Kim

Rutgers Assistant Professor, Dept. of Computer Science

Byungjun Kim

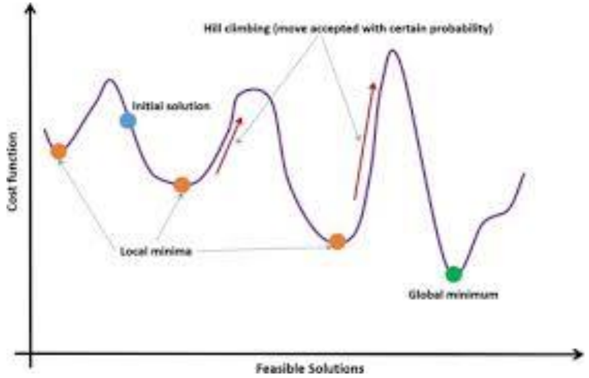
Rutgers Postdoc Researcher

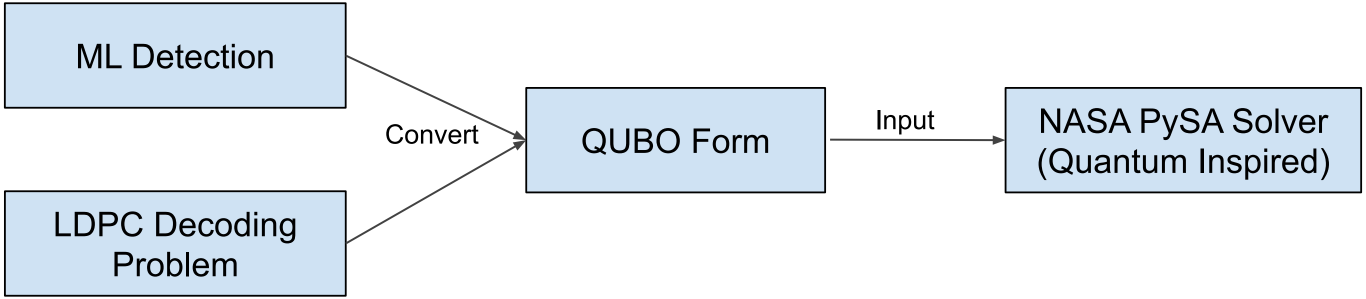
Overview for Simulated Annealing

- Explore techniques to improve PySA for Maximum Likelihood Detection for signals

- What is Simulated Annealing?
    - Imagine a ball on a hill
    - Ball rolls to the lowest point of the landscape

PySA -> simulates the problem in an energy landscape




ML Detection
LDPC Decoding Problem
Convert
QUBO Form
Input
NASA PySA Solver (Quantum Inspired)

PySA: LDPC Decoder

Motivation: As wireless demands for speed and reduced latency increase, classical methods will soon be outpaced

Objective: research and test the use of simulated annealing to solve the decoding section of the problem

- Avoids multiple iterations and takes a physics based approach
- Does not scale as fast as problem size increases, compared to modern methods of decoding

PySA: integer reformulation

Motivation: extend simulated annealing to support high-order QAM modulation (e.g., 64-QAM, 256-QAM)

- by using integers instead of binary variables

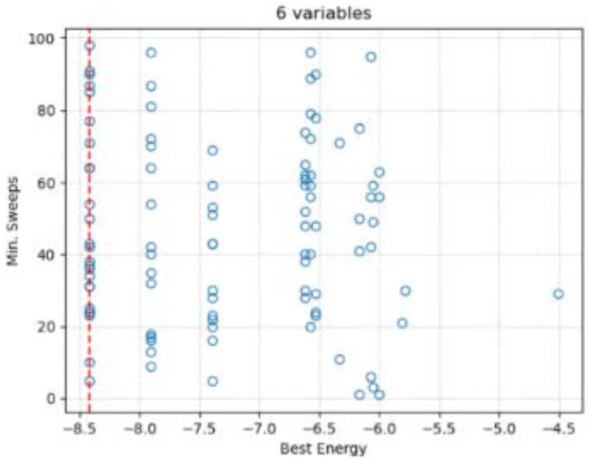
Objective: research and test the use of integer variables with simulated annealing

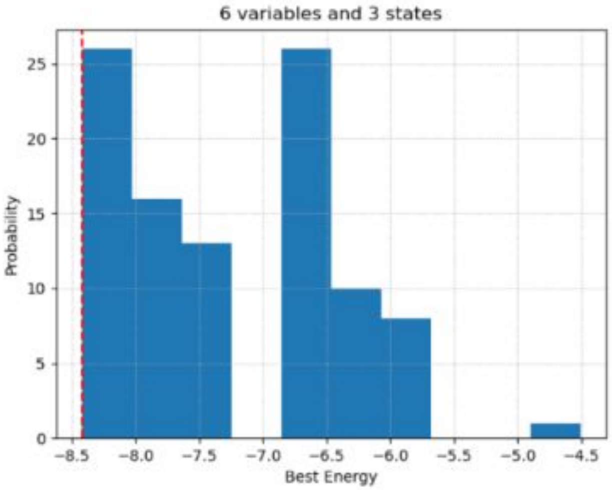
- Avoids binary encoding (multiple binary variables vs 1 integer variable)
- Reduces complexity in a more structured way.

PySA: integer reformulation

Step 1: Test Integers

Tested feasibility of integer-based simulated annealing

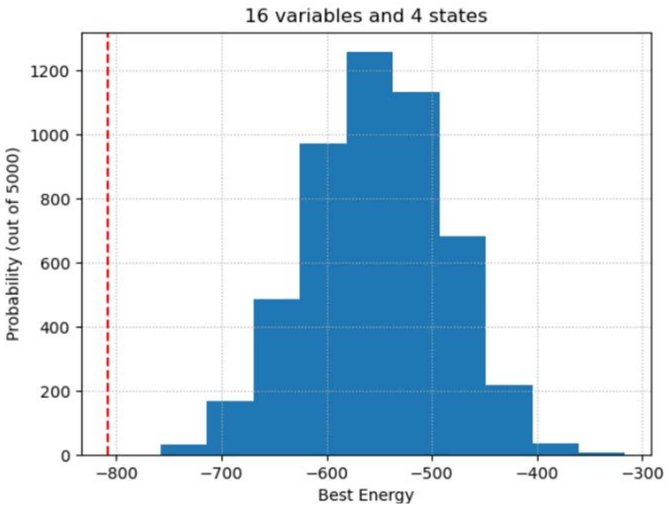


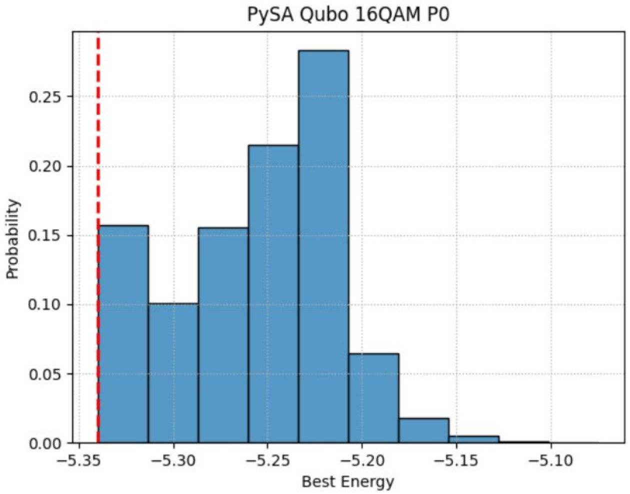


PySA: integer reformulation

Step 2: Test Integers

Tested real problems given by my mentor

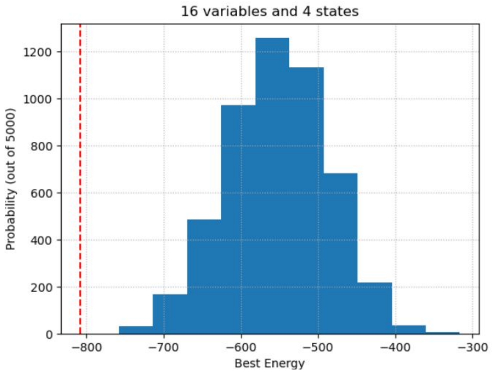




PySA: integer reformulation

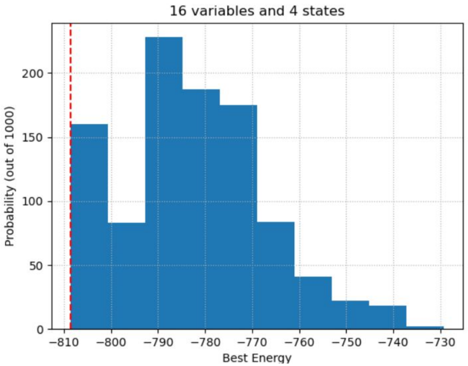
Step 3: Test different helper methods

Tested different methods to improve the probability of convergence



Before

→

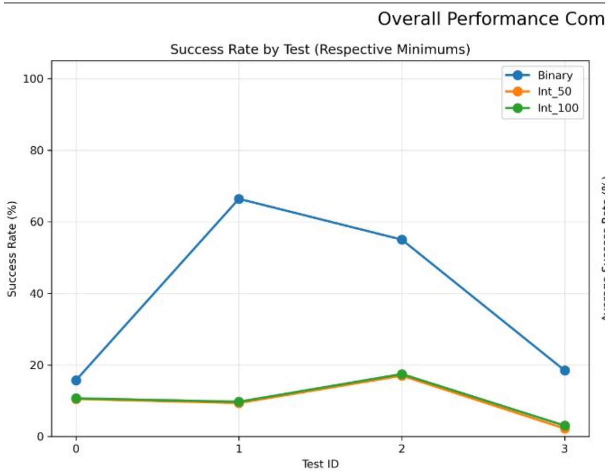


After

PySA: integer reformulation

Step 4: Test current progress

Compared reformulated algorithm with the standard binary version




```
=== Overall Summary ===
Binary: 38.9% average success rate
  Individual test rates: ['15.7%', '66.4%', '55.0%', '18.4%']
Int_50: 9.8% average success rate
  Individual test rates: ['10.5%', '9.4%', '17.0%', '2.2%']
Int_100: 10.2% average success rate
  Individual test rates: ['10.7%', '9.7%', '17.5%', '3.0%']

Average runtime
Binary: 0.002465 us average runtime
Int_50: 4396.985950 us average runtime
Int_100: 7844.585000 us average runtime
PS C:\Users\acer\Desktop\Coding_Projects\WinLab\PySA-1>
```

Background: Isoenergetic Cluster Move Algorithm(ICM)

Enhances Parallel Tempering (PT), which itself builds on Simulated Annealing (SA).

PT runs multiple SA replicas at different temperatures and swaps them to improve search diversity.

ICM improves PT by identifying connected clusters of variables that differ between replicas and swaps them without changing total energy.

These large, neutral swaps help the algorithm jump out of local traps that even PT struggles with.

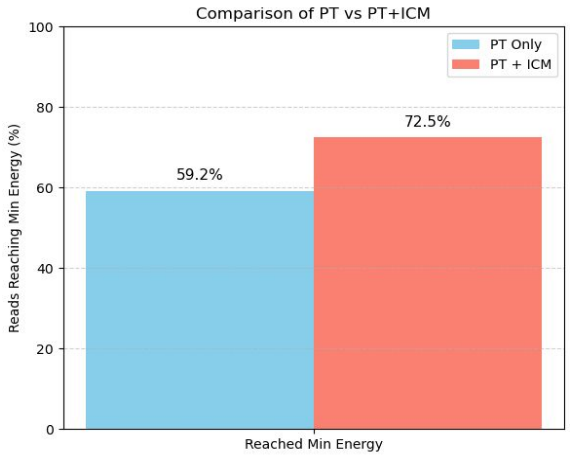
Results: Isoenergetic Cluster Move Algorithm(ICM)

PT+ICM is shown to find the best possible solution in less time than PT alone

PT+ICM also finds the best possible energy at a higher rate than just PT

=== PT vs PT+ICM Summary ===

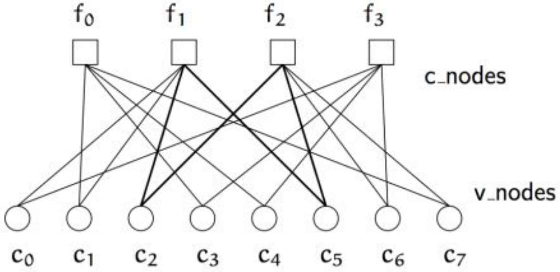
Metric	PT Only	PT+ICM
Best Possible Energy	-101.56826	-101.56826
Average Energy	-94.43826	-98.35826
Samples Reaching Best (%)	59.2	72.5
Avg Sweeps to Best	32.84	22.99



Background: LDPC Codes

Low Density Parity Check (LDPC) Codes:
A form of error correction codes, encode a signal into a low density matrix

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$



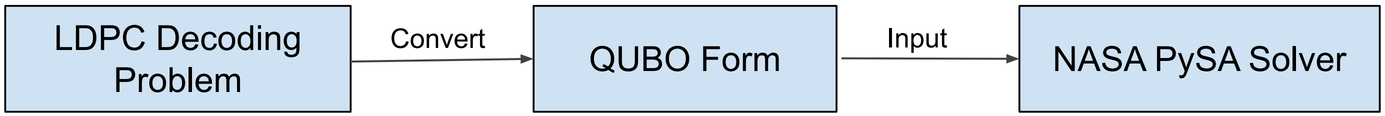
What we want to do

Encoding for LDPC Codes → Simple

Decoding for LDPC Codes → Computationally Difficult

Current Process for Decoding → Belief Propagation (Increases rapidly with code size)

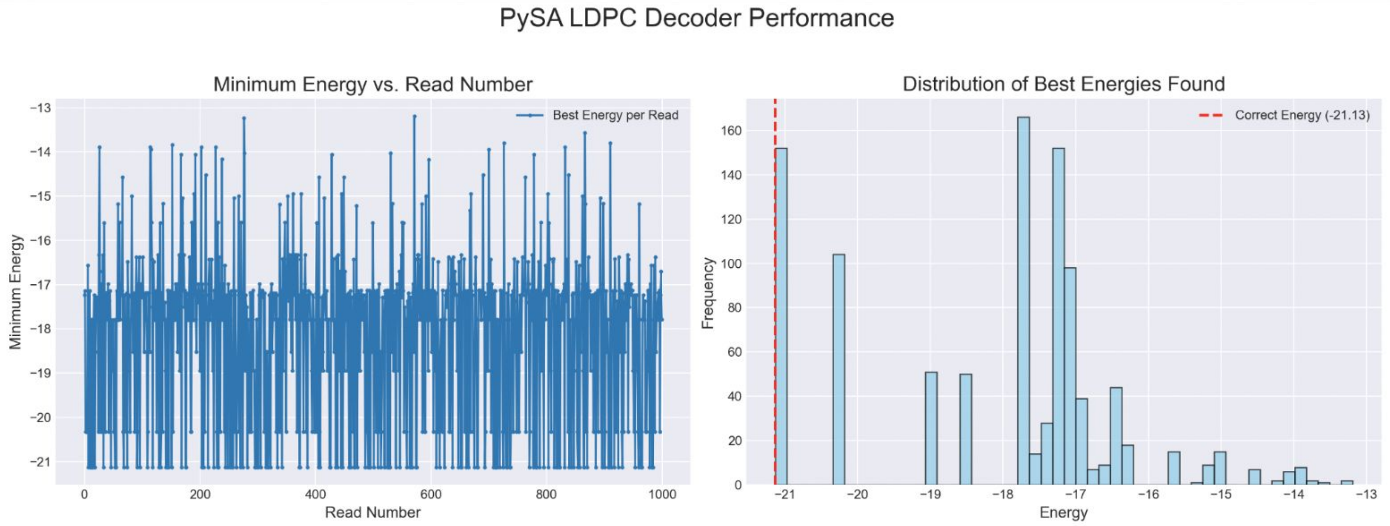
Our Proposed Process → Simulated/Quantum Annealing + Parallel Tempering

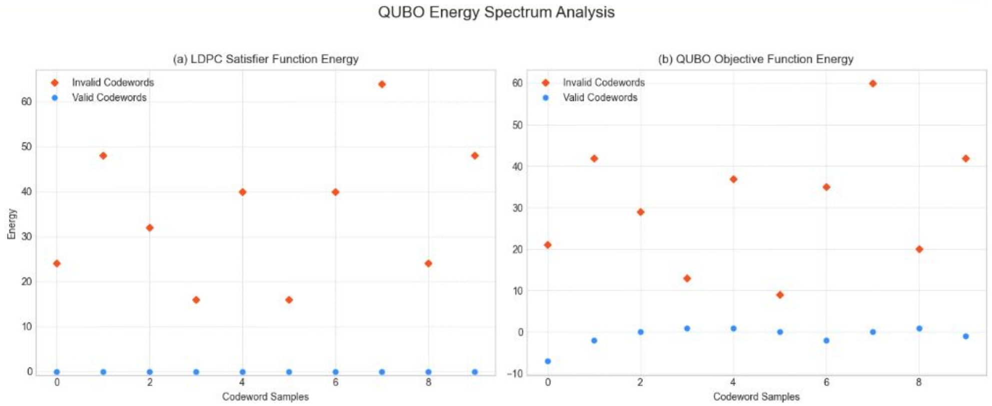


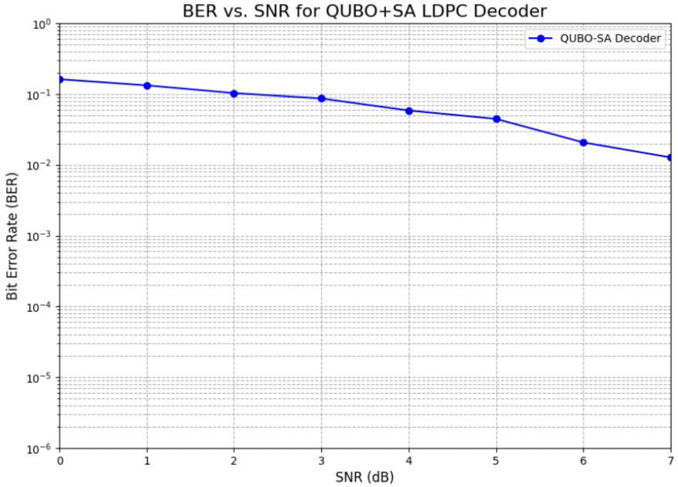
```
--- Running Solver with Final Parameters ---
  num_sweeps: 100
  num_reads: 1000
  min_temp: 0.1
  max_temp: 3
  use_pt: True
```

```
Original message: [1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1]
Decoded message:  [1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1]
Found energy:     -21.1314

Success: The message was correctly decoded.
```




QUBO Energy Spectrum Analysis
(a) LDPC Satisfier Function Energy
Invalid Codewords
Valid Codewords
Energy
Codeword Samples
(b) QUBO Objective Function Energy
Invalid Codewords
Valid Codewords
Codeword Samples


BER vs. SNR for QUBO+SA LDPC Decoder
QUBO-SA Decoder
Bit Error Rate (BER)
SNR (dB)

Any questions?