WISER QUANTUM PROJECT 2: QUANTUM FOR PORTFOLIO OPTIMIZATION

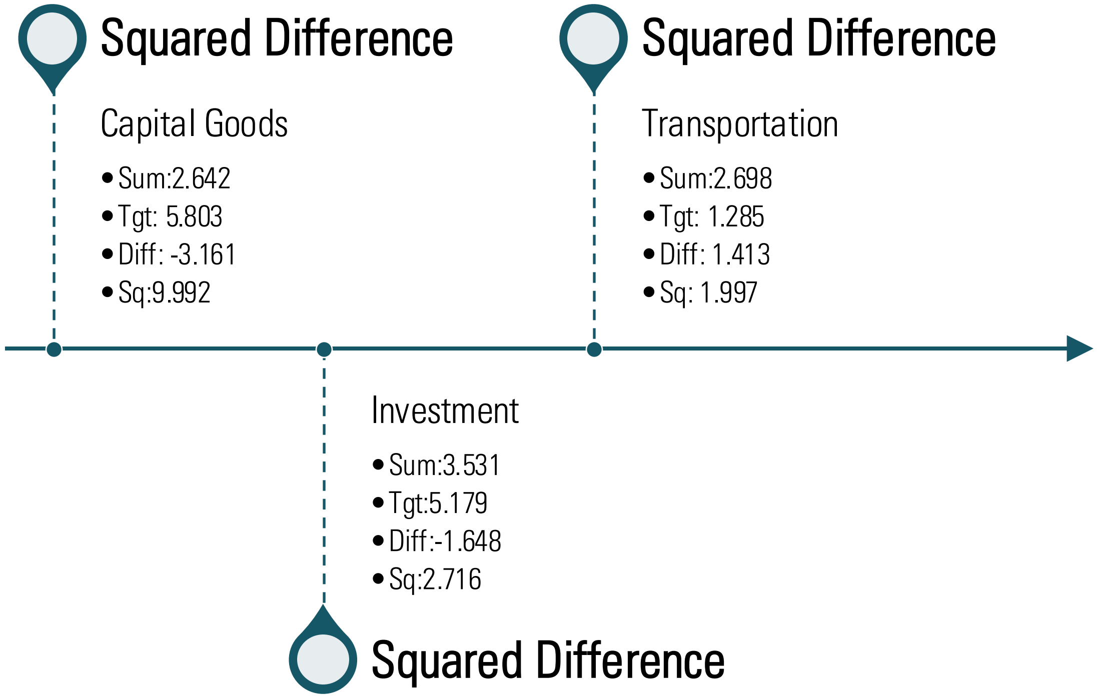
Prepared by: Hao Mack Yang Li

AGENDA

1. PROBLEM STATEMENT (0:10)
2. OUR SOLUTION (0:45)
3. RESULTS & IMPACT (1:40)
4. FUTURE SCOPE (2:40)

PROBLEM STATEMENT

- We are implementing the *simplified OneOpto* optimization model.
- The objective function is a binary quadratic model.
    - Entities include securities, set of risk buckets, set of characteristics, and a set of guardrails.
    - You are to minimize the sum of the squared difference between the target and the current portfolio contribution across all baskets and characteristics.



isin	ccy	assetId	krd1y	oa
1055BJ00	USD	001055BJ0	0.033056371152557	4.80459
1084AS13	USD	001084AS1	0.048588823367606	7.147346
108WAM29	USD	00108WAM2	0.020727166709757	5.197819
108WAP59	USD	00108WAP5	0.040987951949345	6.110968
108WAR16	USD	00108WAR1	0.045538368463281	6.64802
108WAT71	USD	00108WAT7	0.047172649143919	7.136129
115AAR05	USD	00115AAR0	0.044188432993148	7.357668
130HCG83	USD	00130HCG8	0.024547043236489	5.594281
206RGQ92	USD	00206RGQ9	0.037274651529775	4.595146
206RJY99	USD	00206RJY9	0.026847895926904	5.82149
206RKH48	USD	00206RKH4	0.022725597954055	6.497050
206RMM15	USD	00206RMM1	0.026742189942927	7.773578
206RMT67	USD	00206RMT6	0.044475996888843	7.170033
217GAB95	USD	00217GAB9	0.031740965502626	6.338410
2824BQ25	USD	002824BQ2	0.014141440456291	5.312023
287YBX67	USD	00287YBX6	0.029437115786959	4.506160
287YDT38	USD	00287YDT3	0.042026161568062	5.369544
287YDU01	USD	00287YDU0	0.042827438212052	7.340687
440FAA21	USD	00440FAA2	0.069281666971303	4.358510
440KAC71	USD	00440KAC7	0.038005490017179	5.896827
440KAD54	USD	00440KAD5	0.040186212153237	7.849111
510RAD52	USD	00510RAD5	0.021561454838955	5.570760
724PAD15	USD	00724PAD1	0.021774314167194	4.823741
724PAG46	USD	00724PAG4	0.042603342186904	7.421964
7589AD66	USD	007589AD6	0.021723106291976	5.13961
774MAX39	USD	00774MAX3	0.031546980528688	6.245018
774MAY12	USD	00774MAY1	0.033705815187094	7.397469
774MBE49	USD	00774MBE4	0.050479148292221	4.89841
774MBH79	USD	00774MBH7	0.044679594035597	7.105866
774MBK09	USD	00774MBK0	0.057178420415543	4.249826
774MBM64	USD	00774MBM6	0.043469427149448	7.628294
7903BF39	USD	007903BF3	0.035519519576115	6.323033
7944AH47	USD	007944AH4	0.046454659294363	5.916797
7944AK75	USD	007944AK7	0.048165952205359	7.352743
8252AP33	USD	008252AP3	0.031082400946197	4.971313
8252AR98	USD	008252AR9	0.046842924387641	7.418205
846UAM36	USD	00846UAM3	0.020873647356666	5.125274
846UAN19	USD	00846UAN1	0.022869837084699	5.772886
846UAR23	USD	00846UAR2	0.041781376205977	7.699477
8513AA19	USD	008513AA1	0.028263543016454	5.316822
8513AC74	USD	008513AC7	0.027409420598708	7.414848
8513AD57	USD	008513AD5	0.042727195598018	6.452727

WHY DOES IT MATTER

- Numerous trades occur during the day at very high rates, think about high frequency trading.
- The optimization problem becomes **computationally infeasible** classically.
    - The brute force approach is exponential to the number of securities.
    - Local optima and multiple optima can cause optimizers to fail.
    - Even GUROBI may fail to find a global optima within ten minutes, especially with hundreds of risk buckets and tens of characteristics.
- The optimization problem is of a **matrix** form, and is well suited for quantum algorithms, such as QAOA.

A QUANTUM ANNEALING APPROACH

THE VARIABLES

INPUT VARIABLES

- Let C denote the set of securities.
  - Each element c is internally indexed.
  - Each element is expressed as a quintuple:
    - Market price p, Trade range (min, max, inventory), Minimum increment δ
- Let L denote the set of risk buckets
  - Can be a rating or an industry.
- Let J denote the set of characteristics
  - Can be any numeric field in the dataframe.

CONSTRAINT VARIABLES

- There is a guardrail and target K^{target}, K^{low}, K^{up} for each l in L and j in J.
  - There are thus $|L||J|$ such targets, guardrails, and quadratic terms in the objective.
- Let $\mathbf{K}[l]$ denote the subset of bonds belonging to risk bucket l.
- Total bonds N, residual cash guardrail R.
- Output is the $\mathbf{y}$, the included bonds.

THE EQUATION

REDUCED DATASET

- $|C| = 31$
- L = {'Capital Goods', 'Investment', 'Transportation'}
- J = {'PMV'}
- $|K| = 3$
  - $K_{\text{Capital Goods, PMV}} = [4.758, 5.803, 6.847]$
  - $K_{\text{Investment, PMV}} = [4.134, 5.178, 6.223]$
  - $K_{\text{Transportation, PMV}} = [0.240, 1.284, 2.329]$

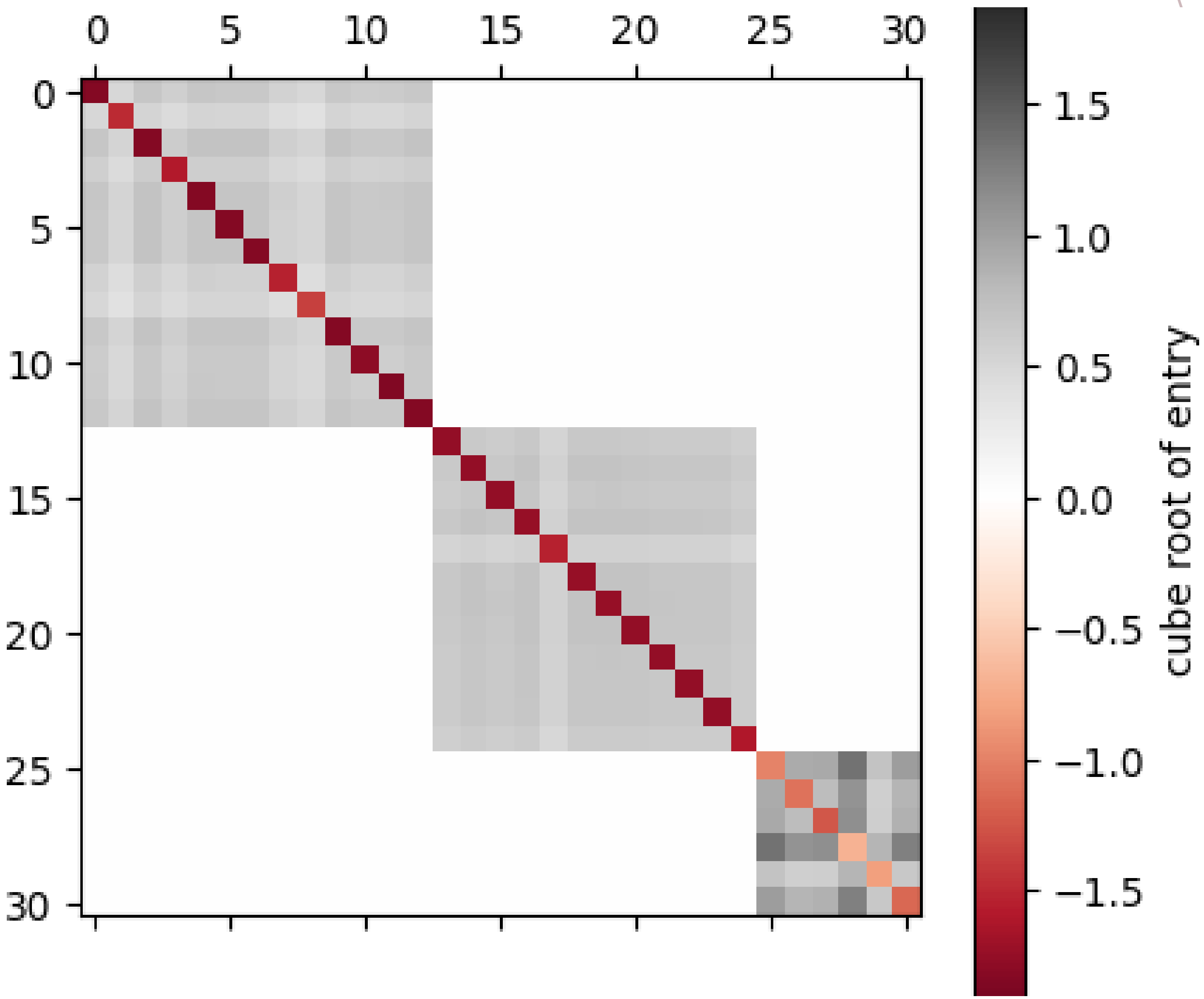
OPTIMIZATION AND CONSTRAINTS

- x_c is a fixed multiple of y_c depending on the minimum and maximum trade amounts.
- ϱ_j and $\beta_{j,c}$ are contribution weights.
- $\min_{\mathbf{y}} \Sigma_l \Sigma_j \varrho_j \left(\Sigma_{c \in \mathbf{K}[l]} \beta_{j,c} x_c - K^{\text{target}}\right)^2$
- $\Sigma_c y_c \leq N$, $\Sigma_c p_c \delta_c \beta_{j,c} x_c \in R$
- $\forall j \forall l \; \Sigma_{c \in \mathbf{K}[l]} p_c \delta_c \beta_{j,c} x_c \in K_{l,j}$

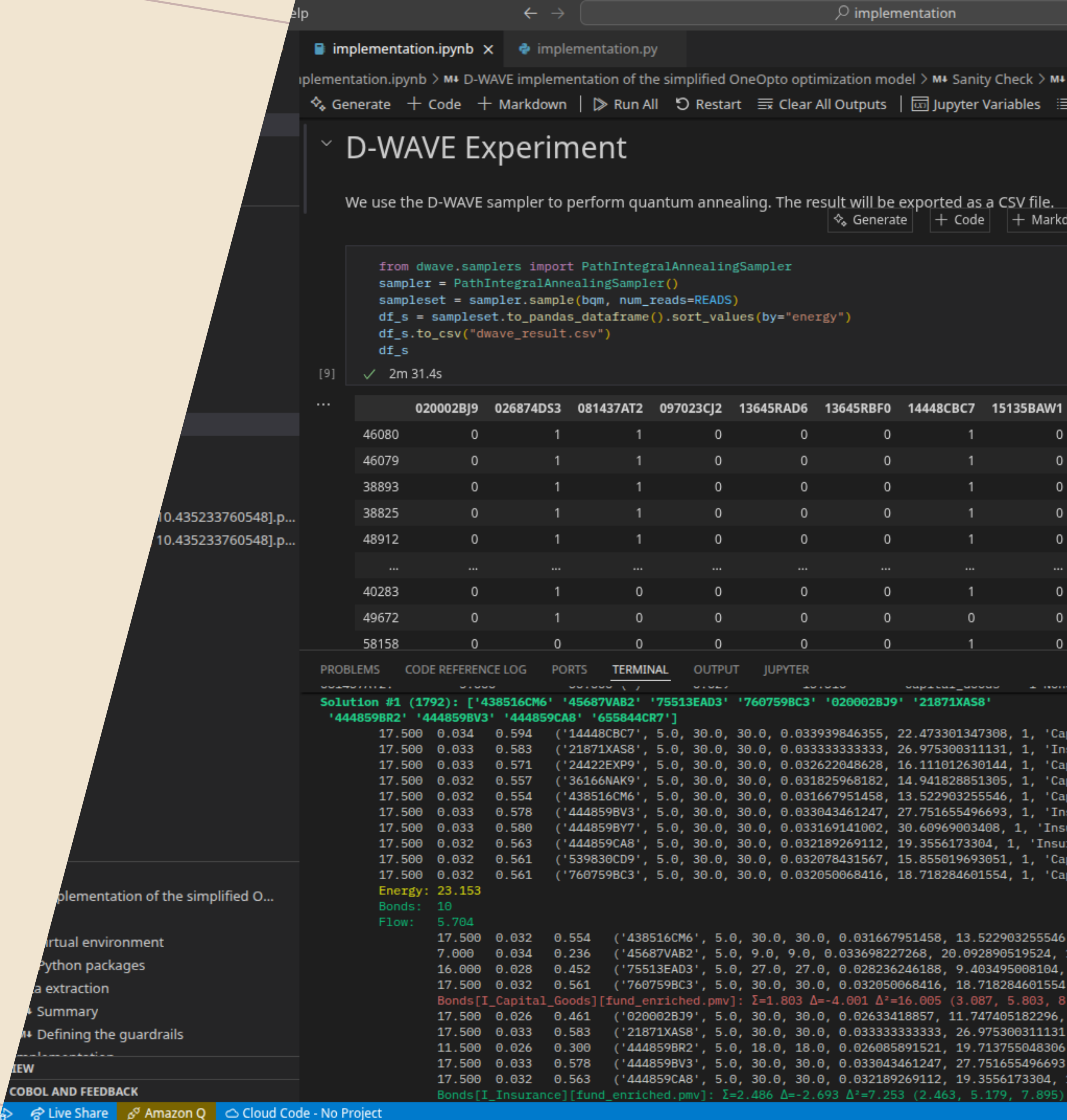
OUR D-WAVE SOLUTION

We implemented our solution in Python, using the *dwave-system* package.

- The optimization equation is converted to expanded form, and then to the format compatible with D-WAVE.
- Each element of $\mathbf{y}$ is named after the bond.
  - Sorted by the risk bucket, so block-diagonal if the buckets do not overlap.
- To the right is the QAOA matrix.
  - Red entries negative, gray entries positive.
  - The off-diagonal entries have much lower magnitudes (order of 0.2 to 1.5) against the diagonal entries (order of –5).



OUR RESULTS



implementation.ipynb
implementation.py
D-WAVE implementation of the simplified OneOpto optimization model > Sanity Check
Generate + Code + Markdown Run All Restart Clear All Outputs Jupyter Variables
D-WAVE Experiment
We use the D-WAVE sampler to perform quantum annealing. The result will be exported as a CSV file.
from dwave.samplers import PathIntegralAnnealingSampler
sampler = PathIntegralAnnealingSampler()
sampleset = sampler.sample(bqm, num_reads=READS)
df_s = sampleset.to_pandas_dataframe().sort_values(by="energy")
df_s.to_csv("dwave_result.csv")
df_s
[9] 2m 31.4s
020002BJ9 026874DS3 081437AT2 097023CJ2 13645RAD6 13645RBF0 14448CBC7 15135BAW1
46080 0 1 1 0 0 0 1 0
46079 0 1 1 0 0 0 1 0
38893 0 1 1 0 0 0 1 0
38825 0 1 1 0 0 0 1 0
48912 0 1 1 0 0 0 1 0
40283 0 1 0 0 0 0 1 0
49672 0 1 0 0 0 0 0 0
58158 0 0 0 0 0 0 1 0
PROBLEMS CODE REFERENCE LOG PORTS TERMINAL OUTPUT JUPYTER
Solution #1 (1792): ['438516CM6' '45687VAB2' '75513EAD3' '760759BC3' '020002BJ9' '21871XAS8' '444859BR2' '444859BV3' '444859CA8' '655844CR7']
Energy: 23.153
Bonds: 10
Flow: 5.704
Live Share
Amazon Q
Cloud Code - No Project

SUMMARY

We are able to successfully implement the minimization of the objective function while enforcing the guardrail conditions.

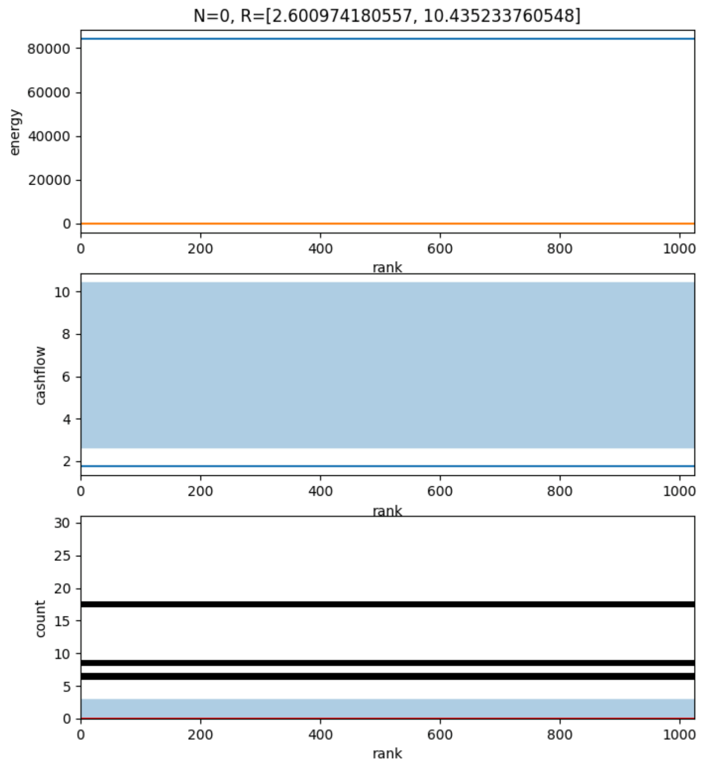
- Unfortunately, we are not able to gain access to the actual quantum resource via LEAP.
- The simulated annealer can provide a rapidly convergent solution. Even with 65536 runs, it only takes three minutes on the 31-bond dataset and three guardrails.

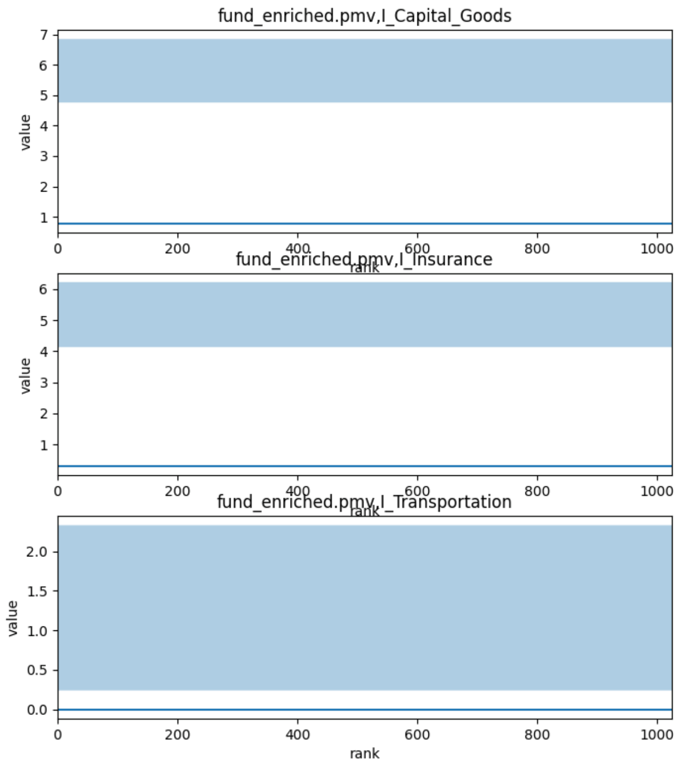
The next slide will show the animation of the samples, with the best one by energy listed *first*.

- D-WAVE energy, QAOA energy
- Cashflow
- Bond selection (by internal index) overlaid with total bonds
- Values and guardrail (assume target is midpoint)

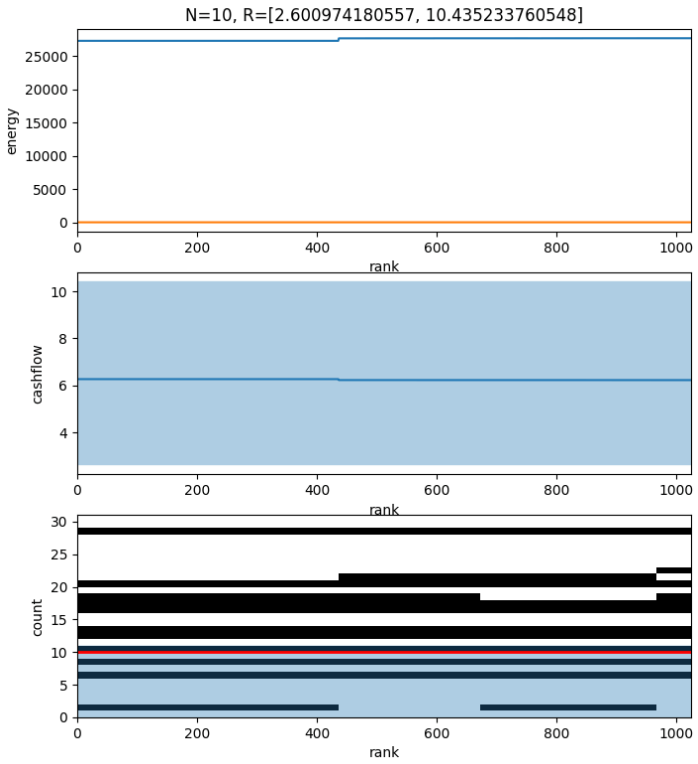
```
Solution #1 (-48): ['081437AT2' '097023CJ2' '14448CBC7' '24422EWZ8' '438516CM6' '443201AC2' '45687VAB2'
'539830CD9' '75513EAD3' '760759BC3' '020002BJ9' '026874DS3' '15135BAW1' '21871XAS8' '444859BR2' '444859BV3'
'444859BY7' '444859CA8' '56501RAN6' '91324PEJ7' '13645RBF0' '314353AA1'] 17.500 0.026 0.461 ('020002BJ9',
5.0, 30.0, 30.0, 0.02633418857, 11.747405182296, 1, 'Insurance', [1]) 17.500 0.032 0.563 ('026874DS3', 5.0,
30.0, 30.0, 0.032168634542, 19.306390960602, 1, 'Insurance', [1]) 17.500 0.029 0.499 ('081437AT2', 5.0,
30.0, 30.0, 0.028510851146, 13.316192340035, 1, 'Capital_Goods', [1]) 9.000 0.027 0.246 ('097023CJ2', 5.0,
13.0, 13.0, 0.027298851045, 34.461503917596, 1, 'Capital_Goods', [1]) 17.500 0.034 0.594 ('14448CBC7', 5.0,
30.0, 30.0, 0.033939846355, 22.473301347308, 1, 'Capital_Goods', [1]) 17.500 0.028 0.487 ('15135BAW1', 5.0,
30.0, 30.0, 0.027845453268, 21.626486622322, 1, 'Insurance', [1]) 17.500 0.033 0.583 ('21871XAS8', 5.0,
30.0, 30.0, 0.033333333333, 26.975300311131, 1, 'Insurance', [1]) 11.000 0.033 0.360 ('24422EWZ8', 5.0,
17.0, 17.0, 0.032714779285, 7.4190547577, 1, 'Capital_Goods', [1]) 17.500 0.033 0.571 ('24422EXP9', 5.0,
30.0, 30.0, 0.032622048628, 16.111012630144, 1, 'Capital_Goods', [1]) 17.500 0.032 0.557 ('36166NAK9', 5.0,
30.0, 30.0, 0.031825968182, 14.941828851305, 1, 'Capital_Goods', [1]) 17.500 0.032 0.554 ('438516CM6', 5.0,
30.0, 30.0, 0.031667951458, 13.522903255546, 1, 'Capital_Goods', [1]) 10.000 0.032 0.319 ('443201AC2', 5.0,
15.0, 15.0, 0.031871754859, 14.154534858858, 1, 'Capital_Goods', [1]) 11.500 0.026 0.300 ('444859BR2', 5.0,
18.0, 18.0, 0.026085891521, 19.713755048306, 1, 'Insurance', [1]) 17.500 0.033 0.578 ('444859BV3', 5.0,
30.0, 30.0, 0.033043461247, 27.751655496693, 1, 'Insurance', [1]) 17.500 0.033 0.580 ('444859BY7', 5.0,
30.0, 30.0, 0.033169141002, 30.60969003408, 1, 'Insurance', [1]) 17.500 0.032 0.563 ('444859CA8', 5.0, 30.0,
30.0, 0.032189269112, 19.3556173304, 1, 'Insurance', [1]) 7.000 0.034 0.236 ('45687VAB2', 5.0, 9.0, 9.0,
0.033698227268, 20.092890519524, 1, 'Capital_Goods', [1]) 17.500 0.032 0.561 ('539830CD9', 5.0, 30.0, 30.0,
0.032078431567, 15.855019693051, 1, 'Capital_Goods', [1]) 17.500 0.030 0.522 ('56501RAN6', 5.0, 30.0, 30.0,
0.029814829851, 13.086051978345, 1, 'Insurance', [1]) 8.500 0.034 0.285 ('655844CT3', 5.0, 12.0, 12.0,
0.033512025872, 0.033512025872, 1, 'Transportation', [1]) 17.500 0.030 0.517 ('759351AP4', 5.0, 30.0, 30.0,
0.029540913811, 11.974870385817, 1, 'Insurance', [1]) 17.500 0.026 0.462 ('760759BA7', 5.0, 30.0, 30.0,
0.026425255721, 15.388018208694, 1, 'Capital_Goods', [1]) Energy: -1.228 Bonds: 22 Flow: 10.399 17.500 0.029
0.499 ('081437AT2', 5.0, 30.0, 30.0, 0.028510851146, 13.316192340035, 1, 'Capital_Goods', [1]) 9.000 0.027
0.246 ('097023CJ2', 5.0, 13.0, 13.0, 0.027298851045, 34.461503917596, 1, 'Capital_Goods', [1]) 17.500 0.034
0.594 ('14448CBC7', 5.0, 30.0, 30.0, 0.033939846355, 22.473301347308, 1, 'Capital_Goods', [1]) 11.000 0.033
0.360 ('24422EWZ8', 5.0, 17.0, 17.0, 0.032714779285, 7.4190547577, 1, 'Capital_Goods', [1]) 17.500 0.032
0.554 ('438516CM6', 5.0, 30.0, 30.0, 0.031667951458, 13.522903255546, 1, 'Capital_Goods', [1]) 10.000 0.032
0.319 ('443201AC2', 5.0, 15.0, 15.0, 0.031871754859, 14.154534858858, 1, 'Capital_Goods', [1]) 7.000 0.034
0.236 ('45687VAB2', 5.0, 9.0, 9.0, 0.033698227268, 20.092890519524, 1, 'Capital_Goods', [1]) 17.500 0.032
0.561 ('539830CD9', 5.0, 30.0, 30.0, 0.032078431567, 15.855019693051, 1, 'Capital_Goods', [1]) 16.000 0.028
0.452 ('75513EAD3', 5.0, 27.0, 27.0, 0.028236246188, 9.403495008104, 1, 'Capital_Goods', [1]) 17.500 0.032
0.561 ('760759BC3', 5.0, 30.0, 30.0, 0.032050068416, 18.718284601554, 1, 'Capital_Goods', [1])
Bonds[I_Capital_Goods][fund_enriched.pmv]: Σ=4.381 Δ=-1.422 Δ²=2.022 (4.759, 5.803, 6.848) 17.500 0.026
0.461 ('020002BJ9', 5.0, 30.0, 30.0, 0.02633418857, 11.747405182296, 1, 'Insurance', [1]) 17.500 0.032 0.563
('026874DS3', 5.0, 30.0, 30.0, 0.032168634542, 19.306390960602, 1, 'Insurance', [1]) 17.500 0.028 0.487
('15135BAW1', 5.0, 30.0, 30.0, 0.027845453268, 21.626486622322, 1, 'Insurance', [1]) 17.500 0.033 0.583
('21871XAS8', 5.0, 30.0, 30.0, 0.033333333333, 26.975300311131, 1, 'Insurance', [1]) 11.500 0.026 0.300
('444859BR2', 5.0, 18.0, 18.0, 0.026085891521, 19.713755048306, 1, 'Insurance', [1]) 17.500 0.033 0.578
('444859BV3', 5.0, 30.0, 30.0, 0.033043461247, 27.751655496693, 1, 'Insurance', [1]) 17.500 0.033 0.580
('444859BY7', 5.0, 30.0, 30.0, 0.033169141002, 30.60969003408, 1, 'Insurance', [1]) 17.500 0.032 0.563
('444859CA8', 5.0, 30.0, 30.0, 0.032189269112, 19.3556173304, 1, 'Insurance', [1]) 17.500 0.030 0.522
('56501RAN6', 5.0, 30.0, 30.0, 0.029814829851, 13.086051978345, 1, 'Insurance', [1]) 13.000 0.031 0.408
('91324PEJ7', 5.0, 21.0, 21.0, 0.031401384436, 11.756014097259, 1, 'Insurance', [1])
Bonds[I_Insurance][fund_enriched.pmv]: Σ=5.046 Δ=-0.133 Δ²=0.018 (4.134, 5.179, 6.224) 23.000 0.028 0.641
('13645RBF0', 5.0, 41.0, 41.0, 0.027884438059, 0.027884438059, 1, 'Transportation', [1]) 32.500 0.021 0.695
('314353AA1', 5.0, 60.0, 60.0, 0.021396294681, 0.021396294681, 1, 'Transportation', [1])
Bonds[I_Transportation][fund_enriched.pmv]: Σ=1.337 Δ=0.052 Δ²=0.003 (0.240, 1.285, 2.329)
```

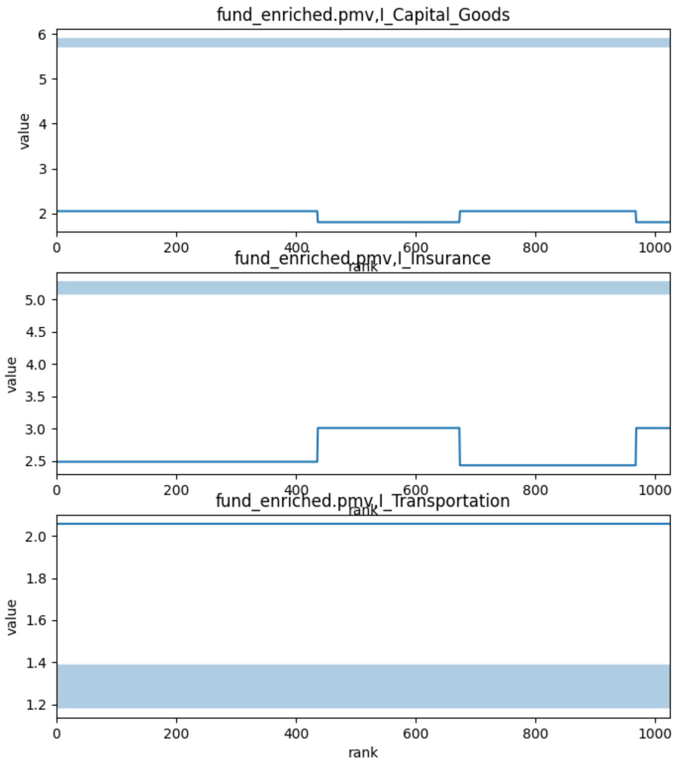
CHANGING BOND LIMIT (30M)





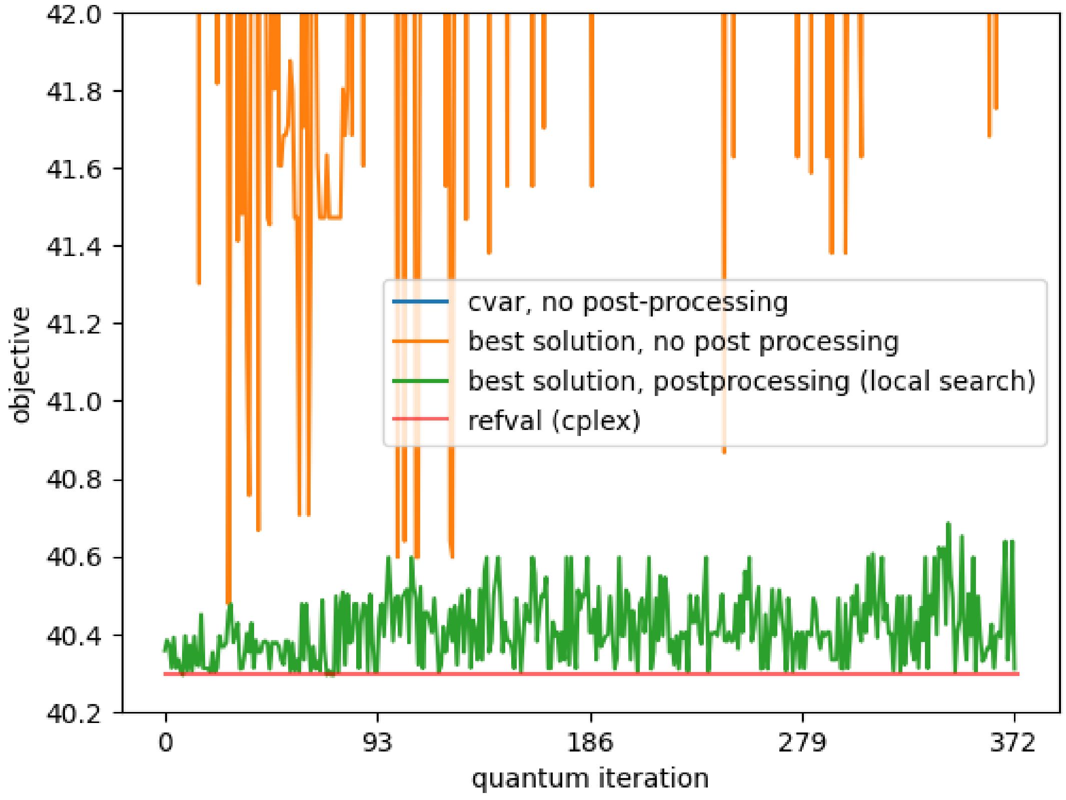
CHANGING GUARDRAILS (2H)

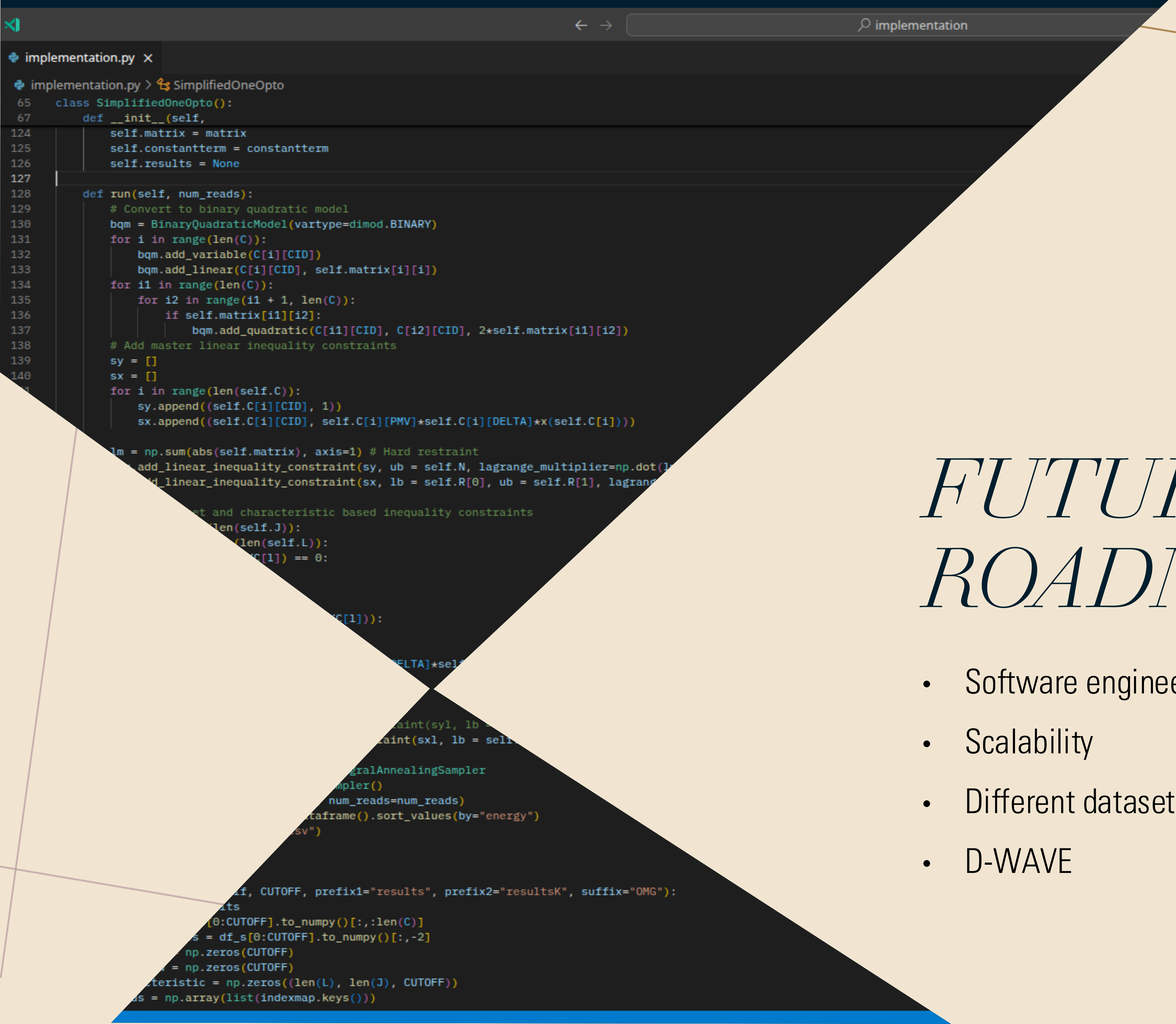




SCALABILITY AND CLASSICAL

- The reference run uses a real Qiskit run with or without postprocessing.
- In the small dataset, the reference optimal value has an energy of 40.3.
  - The actual Qiskit quantum run without postprocessing considerably misses the goal.
  - With postprocessing, it misses the goal by 0.15 units on average.
- D-WAVE energy is separate, but correlated with QAOA energy; D-WAVE energy factors in guardrails as well.





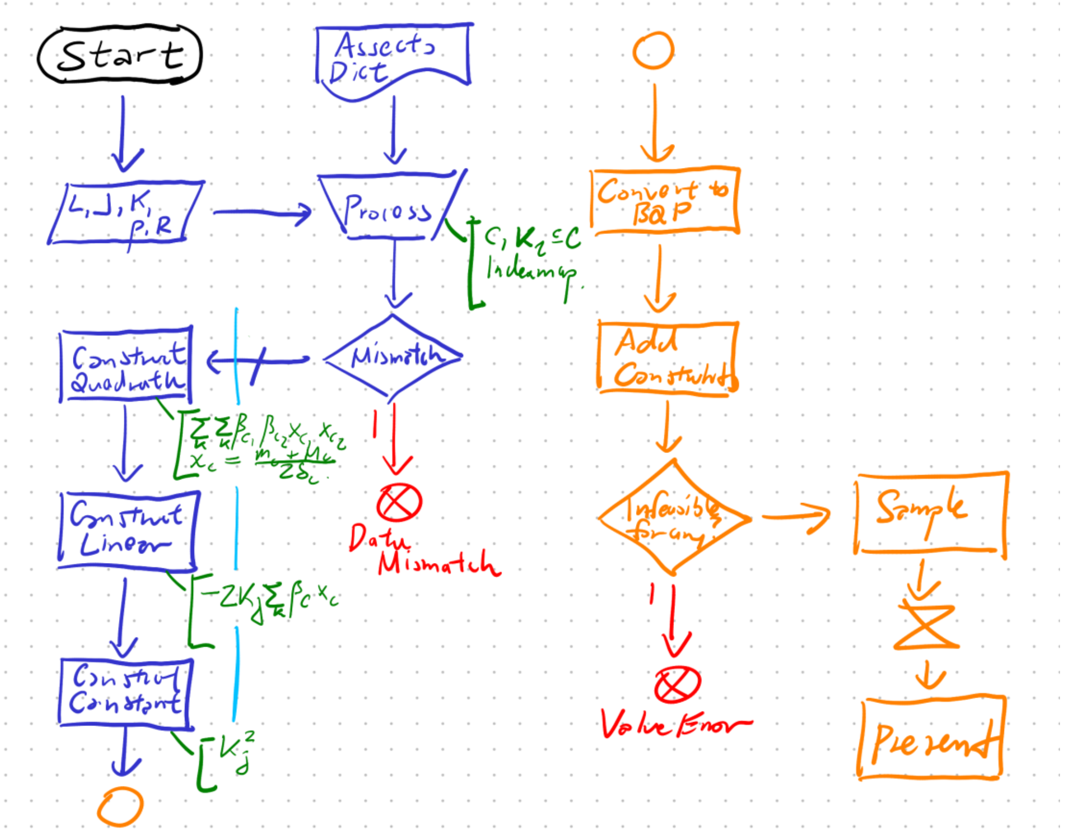
FUTURE ROADMAP

- Software engineering
- Scalability
- Different datasets
- D-WAVE

CODE MAINTENANCE

Much time spent on coding instead visits the theory by providing test examples to affirm or refute the correctness of the program, and to refactor code to adhere to software engineering principles.

- Object-oriented programming
  - A dedicated C and K object.
  - Architecture to repeatedly run the procedure while varying some of the values.
- Exception handling
  - Prevents harder to trace exceptions.
- Code reusability
  - Allows easy inspection, avoids code smells.



LIMITATIONS

- To conclude this presentation, we will address the limitations encountered during the submission period.
  - We did not get our LEAP application approved within the timeframe, so we cannot measure our progress using a real quantum annealer, which can support up to thousands of variables.
  - We mostly discussed the smallest dataset.
  - Though software engineering best practices have been follows and we are able to see results, we need to scrutinize our code again to resolve any anomalies, such as
    - Negative values of the optimization function.
    - The constant bias between different quantum and classical optimization process.
    - Potentially confusing the variables and guardrails they officially provided versus my implementation.
  - It is important to time manage, and not crunch on time.

THANK YOU

COMMENT BELOW FOR ANY QUESTIONS!