Markowitz Portfolio Optimization with a Quantum Annealer

Erica Grant, Travis Humble
Oak Ridge National Laboratory
University of Tennessee
Portfolio Selection

Goals:
• Maximize returns
• Minimize risk
• Stay within budget

Inputs:
• Uniform random historical price data
• Budget
• Risk tolerance

Output:
• A portfolio representing a list of investments and the expected return

<table>
<thead>
<tr>
<th>Investment</th>
<th>Price</th>
<th>Expected Return/Day</th>
<th>1= buy 0 = pass</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stock A</td>
<td>$150</td>
<td>+$0.10</td>
<td>0</td>
</tr>
<tr>
<td>Stock B</td>
<td>$200</td>
<td>+$0.25</td>
<td>1</td>
</tr>
<tr>
<td>Stock C</td>
<td>$250</td>
<td>+$0.50</td>
<td>0</td>
</tr>
</tbody>
</table>

Budget: $200
Risk Tolerance: Low
Portfolio Selection as Binary Integer Programming

\[
\begin{align*}
\max_x & \quad \sum_i r_i x_i \\
\text{s.t.} & \quad \sum_i p_i x_i \leq b
\end{align*}
\]

\( x \in \{0, 1\} \)

\( i = \text{asset number} \)

\( p_i = \text{price} \)

\( r_i = \text{expected return} \)

\( b = \text{budget} \)
Unconstrained Markowitz Formulation

\[ \max_x f(x) \]

\[
f(x) = \theta_1 \sum_i x_i r_{i} x_i - \theta_2 \left( \sum_i x_i p_i x_i - b \right)^2 - \theta_3 \sum_{i, j} x_i \text{cov}(h_i, h_j) x_j
\]

\[
x_i = 1 \rightarrow \text{buy} \quad x_i = 0 \rightarrow \text{don't buy}
\]

- \( b = \text{budget}, \ p_i = \text{price}, \ r_i = \text{expected return}, \ x_i \in \{0, 1\} \)
- Weights: \( \theta_1, \theta_2, \theta_3 \geq 0 \) s.t. \( \theta_1 + \theta_2 + \theta_3 = 1 \)
- \( h_i = \text{vector of historical price data} \)
- \( \text{cov}(h_i, h_j) = \frac{\sum_{k=1}^{m}(h_{i,k} - \bar{h}_i)(h_{j,k} - \bar{h}_j)}{m-1} \) where \( m = \text{number of price points} \)
Unconstrained Markowitz Formulation

\[ \max_x f(x) \]

\[ f(x) = \theta_1 \sum_i x_i r_{ii} x_i - \theta_2 \sum_i (x_i p_i x_i - b)^2 - \theta_3 \sum_{i,j} x_i \text{cov}(h_i, h_j) x_j \]

\[ x_i = 1 \rightarrow \text{buy} \quad x_i = 0 \rightarrow \text{don't buy} \]

- \( b = \text{budget}, \ p_i = \text{price}, \ r_i = \text{expected return}, \ x_i \in \{0, 1\} \)
- \( \text{Weights: } \theta_1, \theta_2, \theta_3 \geq 0 \text{ s.t. } \theta_1 + \theta_2 + \theta_3 = 1 \)
- \( h_i = \text{vector of historical price data} \)
- \( \text{cov}(h_i, h_j) = \frac{\sum_{k=1}^n (h_{i,k} - \bar{h}_i)(h_{j,k} - \bar{h}_j)}{n} \) where \( n = \# \text{of assets} \)
Uncorstrained Markowitz Formulation

\[
\begin{align*}
\max_{x} & \quad f(x) \\
\text{Subject to} & \quad x_i = 1 \rightarrow \text{buy} \quad x_i = 0 \rightarrow \text{don't buy}
\end{align*}
\]

\[
f(x) = \theta_1 \sum_i x_i r_{ii} x_i - \theta_2 \sum_i (x_i p_i x_i - b)^2 - \theta_3 \sum_{i,j} x_i \text{cov}(h_i, h_j) x_j
\]

- \( b = \text{budget} \), \( p_i = \text{price} \), \( r_i = \text{expected return} \), \( x_i \in \{0, 1\} \)
- Weights: \( \theta_1, \theta_2, \theta_3 \geq 0 \) s.t. \( \theta_1 + \theta_2 + \theta_3 = 1 \)
- \( h_i = \text{vector of historical price data} \)
- \( \text{cov}(h_i, h_j) = \frac{\sum_{k=1}^{n} (h_{i,k} - \overline{h_i})(h_{j,k} - \overline{h_j})}{n} \) where \( n = \# \text{of assets} \)
Unconstrained Markowitz Formulation

\[ \max_x f(x) \]

\[ f(x) = \theta_1 \sum_i x_i r_{ii} x_i - \theta_2 \sum_i (x_i p_i x_i - b)^2 - \theta_3 \sum_{i,j} x_i \text{cov}(h_i, h_j) x_j \]

\[ x_i = 1 \rightarrow \text{buy} \quad x_i = 0 \rightarrow \text{don't buy} \]

- \( b = \text{budget} \), \( p_i = \text{price} \), \( r_i = \text{expected return} \), \( x_i \in \{0, 1\} \)
- Weights: \( \theta_1, \theta_2, \theta_3 \geq 0 \) s.t. \( \theta_1 + \theta_2 + \theta_3 = 1 \)
- \( h_i = \text{vector of historical price data} \)
- \( \text{cov}(h_i, h_j) = \frac{\sum_{k=1}^n (h_{i,k} - \bar{h}_i)(h_{j,k} - \bar{h}_j)}{n} \) where \( n = \# \text{of assets} \)
Fitting to Unconstrained Problem

- Assets can be divided into any desired fraction based on the budget.
- Normalize purchase price to the budget.
- Use Binary Fractional series:
  \[
  \frac{1}{2^0}, \frac{1}{2^1}, \frac{1}{2^2}, \ldots, \frac{1}{2^n}
  \]

Special thanks to Benjamin Stump for the binary fractional series idea.
Quantum Annealing for Unstructured Search

- Quantum annealing is a model of quantum computing tailored to unconstrained search.
- QA operates by preparing a uniform superposition of all possible binary combinations.
- The initial state is then evolved toward a Hamiltonian that defines the objective function.
- Measurement samples the prepared probability distribution, which should concentrate at the global extrema.

![Diagram of quantum annealing process](image)
Quantum Annealing for Unstructured Search

- D-Wave Systems provides a fourth-generation quantum annealer, 2000Q
  - Programmable superconducting integrated circuit
  - 2048-qubit register in a 2D Chimera layout
  - EM shielding, UHV, cooled to 14mK

- This is a special-purpose optimization solver that finds the energetic minimum of an Ising model
  - Search uses quantum annealing to find minima

- Non-zero temperature, flux noise, and other fluctuations complicate device physics
QUBO to Quantum Ising Hamiltonian

**QUBO:** $x \in \{0, 1\}$

$$f(x) = -\theta_1 \sum_i x_i r_{ii} x_i + \theta_2 \sum_i (x_i p_i x_i - b)^2 + \theta_3 \sum_{i,j} x_i \text{cov}(p_i, p_j) x_j$$

$$f(x) = \tilde{Q}_{i,j} x_i x_j + q_i x_i$$

$q_i = Q_{ii}$ and $\tilde{Q}_{i,j} = Q_{i,j}$ ($i \neq j$)

**Ising:** $y \in \{-1, 1\}$

\[ J_{i,j} = \frac{1}{4} Q_{i,j} \]

\[ h_i = \frac{q_i}{2} + \sum_j J_{i,j} \]

\[ \gamma = \frac{1}{4} \sum_{i,j} Q_{i,j} + \frac{1}{2} \sum_i q_i \]

\[ min_f(x) \]

$$f(x) = \sum_i h_i y_i + \sum_{i,j} J_{i,j} y_i y_j + \gamma$$

$$\hat{H} = \sum_{i,j} J_{i,j} \hat{x}_i \hat{x}_j + \sum_i h_i \hat{x}_i + \gamma$$

**Coupler Strengths**
QUBO to Quantum Ising Hamiltonian

**QUBO**: $x \in \{0, 1\}$

$$f(x) = -\theta_1 \sum_i x_i r_{ii} x_i + \theta_2 \sum_i (x_i p_i x_i - b)^2 + \theta_3 \sum_{i,j} x_i \text{cov}(p_i, p_j) x_j$$

$$f(x) = Q_i, j x_i x_j + q_i x_i$$

$q_i = Q_{ii}$ and $Q_{i,j} = Q_{j,i} (i \neq j)$

**Ising**: $y \in \{-1, 1\}$

$$J_{i,j} = \frac{1}{4} Q_{i,j}$$

$$h_i = \frac{q_i}{2} + \sum_j J_{i,j}$$

$$\gamma = \frac{1}{4} \sum J_{i,j} Q_{i,j} + \frac{1}{2} \sum q_i$$

$$\min_x f(x)$$

$$f(x) = \sum_i h_i y_i + \sum_{i,j} J_{i,j} y_i y_j + \gamma$$

$$\hat{H} = \sum_{i,j} J_{i,j} \hat{z}_i \hat{z}_j + \sum_i h_i \hat{x}_i + \gamma$$

**Qubit Weights**
QUBO to Quantum Ising Hamiltonian

**QUBO**: \( x \in \{0, 1\} \)

\[
f(x) = -\theta_1 \sum_i x_i r_{ii} x_i + \theta_2 \sum_i (x_i p_i x_i - b)^2 + \theta_3 \sum_{i,j} x_i \text{cov}(p_i, p_j) x_j
\]

\[
f(x) = Q_{\sim,i} x_i x_j + q_i x_i
\]

\( q_i = Q_{ii} \) and \( Q_{\sim,i} = Q_{i,j} \) \((i \neq j)\)

**Ising**: \( y \in \{-1, 1\} \)

\[
J_{i,j} = \frac{1}{4} Q_{\sim,i} \\
J_{i,j} = \frac{1}{2} q_i + \sum_j J_{i,j}
\]

\[
\gamma = \frac{1}{4} \sum_{i,j} Q_{i,j} + \frac{1}{2} \sum_i q_i
\]

\[
\min_x f(x) = \sum_i h_i y_i + \sum_{i,j} J_{i,j} y_i y_j + \gamma
\]

\[
\hat{H} = \sum_{i,j} J_{i,j} \hat{z}_i \hat{z}_j + \sum_i h_i \hat{x}_i + \gamma
\]
QUBO to Quantum Ising Hamiltonian

**QUBO**: \( x \in \{0, 1\} \)

\[
f(x) = -\theta_1 \sum_i x_i r_{ii} x_i + \theta_2 \sum_i (x_i p_i x_i - b)^2 + \theta_3 \sum_{i,j} x_i \text{cov}(p_i, p_j) x_j
\]

\[
f(x) = Q_{i,j} x_i x_j + q_i x_i
\]

\( q_i = Q_{ii} \) and \( Q_{i,j} = Q_{i,j} (i \neq j) \)

**Ising**: \( y \in \{-1, 1\} \)

\[
J_{i,j} = \frac{1}{4} Q_{i,j} \quad h_i = \frac{q_i}{2} + \sum_j J_{i,j}
\]

\[
\gamma = \frac{1}{4} \sum_{i,j} Q_{i,j} + \frac{1}{2} \sum_i q_i
\]

**Ising Form**

\[
\min_x f(x)
\]

\[
f(x) = \sum_i h_i y_i + \sum_{i,j} J_{i,j} y_i y_j + \gamma
\]

\[
\hat{H} = \sum_{i,j} J_{i,j} \hat{x}_i \hat{x}_j + \sum_i h_i \hat{x}_i + \gamma
\]
QUBO to Quantum Ising Hamiltonian

**QUBO:** \( x \in \{0, 1\} \)

\[
f(x) = -\theta_1 \sum_i x_i r_{ii} x_i + \theta_2 \sum_i (x_i p_i x_i - b)^2 + \theta_3 \sum_{i,j} x_i \text{cov}(p_i, p_j) x_j
\]

\[
f(x) = \tilde{Q}_{i,j} x_i x_j + q_i x_i
\]

\( q_i = Q_{ii} \) and \( \tilde{Q}_{i,j} = Q_{i,j} (i \neq j) \)

**Ising:** \( y \in \{-1, 1\} \)

\[
J_{i,j} = \frac{1}{4} \tilde{Q}_{i,j} \quad h_i = \frac{q_i}{2} + \sum_j J_{i,j} \]

\[
\gamma = \frac{1}{4} \sum_{i,j} Q_{i,j} + \frac{1}{2} \sum_i q_i
\]

\[
\min_x f(x) \quad f(x) = \sum_i h_i y_i + \sum_{i,j} J_{i,j} y_i y_j + \gamma
\]

**Quantum Ising**

\[
\hat{H} = \sum_{i,j} J_{i,j} \hat{z}_i \hat{z}_j + \sum_i h_i \hat{x}_i + \gamma
\]
Solving on D-Wave 2000Q

**D-Wave 2000Q:**
- 2048 qubits
- Anneal times
- Embeddings
- Reverse Annealing

**Parameters:**
- $\theta_1, \theta_2, \theta_3$
- Number of assets
- Random historical Price Data

**Outputs:**
- Portfolio:
  
  $[-1, 1, 1, 1, -1, -1...]$ →
  
  $[0, 1, 1, 1, 0, 0...]
  
  - Energy of the portfolio
Benchmarking against Brute Force Solver

• Probability of success:

\[ POS = \frac{l}{N_s} \]

- \( l \) = # of ground state energies found by the D-Wave
- \( N_s \) = number of samples/anneals

• Average Probability of success

\[ [POS] = \sum_{i}^{N_p} \frac{POS(i)}{N_p} \]

- \( N_p \) = # problems
- \( i \) indicates problem

• POS ratio:

\[ \frac{[POS]_R}{[POS]_F} \]

- \([POS]_R\) for reverse annealing
- \([POS]_F\) for forward annealing

• Energy Binning

Energy \( \rightarrow \) Energy Level
Best Fit for Probability of Success

Probability of Success on D-Wave 2000Q

\[ y = 1.2245 e^{-0.5714x^{0.9387}} \]

- Embedding = Find_embedding()
- Slices = 0
- Anneal time = 5\(\mu s\)
- Forward annealing
Forward Annealing Controls

Variation With Anneal Time

- Anneal Time: 5 µs Weights: .3, .5, .2
- Anneal Time: 100 µs Weights: .3, .5, .2
- Anneal Time: 250 µs Weights: .3, .5, .2

Probability of Success vs Number of Assets
Spin Reversal Control

Probability of Success on D-Wave 2000Q

Number of Assets

Probability of Success

Spin Reversals: 0
Spin Reversals: 2
Spin Reversals: 5
Spin Reversals: 10
Example Program Embedding in Hardware

- **find_embedding()**
  - Embedding
  - Problem size = 20
  - 23 unit cells

- **Clique**
  - Embedding
  - 15 unit cells
Clique vs Find Embedding

Forward Annealing:

- Embeddings:
  - find_embedding()
  - Clique
- Anneal Time = 15μs
- Spin Reversal = 0
Clique vs Find Embedding

Anneal time = 15 μs
Number of anneals = 1,000
Forward Annealing: Clique vs Find Embedding

Anneal time = 15 µs
Number of anneals = 1,000
Reverse Annealing

Parameters:

- Initial State
- Reinitialize
- S
- Pause time
Reverse Annealing Sweep

Initial State = Ground
Assets = 5, Slices = 4

Average POS: Reverse/Forward

Pause Time (μs)

> 1

POS: Reverse/Forward

< 1
Reverse Annealing Sweep

Initial State = First Assets = 5, Slices = 4

Average POS: Reverse/Forward

Pause Time (μs)

Pause Time (μs)
Reverse Annealing Sweep

Initial State = Highest Assets = 5, Slices = 4

Average POS: Reverse/Forward

POS: Reverse/Forward
Reverse Annealing Sweep

Initial State = Forward
Assets = 5, Slices = 4
Overall Probability of Success

Forward Annealing:
- Embeddings: find_embedding()
- Clique
- Anneal Time = 15μs
- Spin Reversal = 0

Reverse Annealing:
- S = .9
- Pause = 15μs
- I.S. = Forward
- Reinitialize = False
Conclusions

- Quantum annealing is a method for solving QUBO problems and Markowitz Portfolio optimization is an example which demonstrates real-world application.

- Various parameters both in the QUBO and the D-Wave computer can be controlled/fine-tuned to yield better results.

- Forward annealing reveals a sub-exponential decrease in probability of success as problem size increases.

- Spin reversal transform improves variance.

- Clique Embedding improves probability of success over the find_embedding() method.

- Reverse annealing reveals a better probability of success and a better average solution if initial state is close to ground state.
Future Work

- Measure efficiency of using reverse annealing with forward annealing.
- Compare quantum annealing to classical heuristics.
- Investigate using real market data.
Questions

University of Tennessee, Knoxville
Oak Ridge National Laboratory
Quantum Computing Institute

In collaboration with Khalifa University
Nada Elsokkary, Faisal Khan
<table>
<thead>
<tr>
<th>Asset i</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>350</td>
<td></td>
</tr>
<tr>
<td>375</td>
<td></td>
</tr>
<tr>
<td>312</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td></td>
</tr>
<tr>
<td>330</td>
<td></td>
</tr>
</tbody>
</table>

\[
\frac{p_k}{p_m}
\]

\[
a_i \rightarrow a_{i,k}
\]

<table>
<thead>
<tr>
<th>Asset i</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0606</td>
<td>1.1363</td>
</tr>
<tr>
<td>.94545</td>
<td>.90909</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Asset i, 0</th>
<th>Asset i, 1</th>
<th>Asset i, 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0606</td>
<td>.53030</td>
<td>.26515</td>
</tr>
<tr>
<td>1.1363</td>
<td>.56815</td>
<td>.28408</td>
</tr>
<tr>
<td>.94545</td>
<td>.47273</td>
<td>.23636</td>
</tr>
<tr>
<td>.90909</td>
<td>.45450</td>
<td>.22727</td>
</tr>
<tr>
<td>1</td>
<td>.5</td>
<td>.25</td>
</tr>
</tbody>
</table>
Probability of Success

Frequency of Success on D-Wave 2000Q
Finding the Optimal Solution for 1,000 Problems
Anneal Time: 5 microseconds, Weights: .3, .5, .2

Number of Assets

Problems

Fraction of Success
10,000 samples
0 0.2 0.4 0.6 0.8 1
Optimality Gap

Optimality Gap:
\[ \text{abs} \left( \frac{E_{\text{obs}}}{E_{\text{exp}}} - 1 \right) \]