BTC_Mining_Simulation

May 23, 2025

[55]:
```python
import pandas as pd
import sim
import importlib
importlib.reload(sim)
```

[55]:
```
<module 'sim' from '/Users/kunjshah/Downloads/BTC_Mining_Project/sim.py'>
```

Assumptions:

- Risk-free rate: $r_f = 0.04$
- Bitcoin mining rate per rig: Mine rate = 0.00008 BTC/day
- Power consumption per rig: 3,250 W/day
- Number of Antminers: 1,000
- Market utilized: Day Ahead
- M_Paths = 500

Node: NSPNEWLOAD

[56]:
```python
nsp= sim.run_sim(seed=None, intraday=False, analysis = False,␣
↪plot_vol_surface=False, Node='NSP')
```

```
LSMC optimal NPV : $967,443
```

[57]:
```python
#Average Annual Profit
profit = nsp['npv_lsm']

#Volatility in profit
vol = nsp['npv_lst_std']

#Sharpe
sharpe = profit / vol

print(f"Average Profit: ${profit:.2f}")
print(f"Volatility: {vol:.0f}")
print(f"Sharpe Ratio: {sharpe:.2f}")
```

```
Average Profit: $967442.67
Volatility: 849025
Sharpe Ratio: 1.14
```

Intraday Analysis

Assumptions:

- Intraday Period = 15M

[58]:
```
intra = sim.run_sim(seed=None, intraday=True, analysis = True,␣
 ↪plot_vol_surface=True, Node='NSP')
```

```
LSMC optimal NPV : $540,677
```

[74]:
```
#Print Summary

print(f"Always operating ${intra['Operate all ($)'].values[0]}")
print(f"Least Squares Monte Carlo Payout ${intra['Net payout ($)'].values[0]} ")
print(f"Curtail Value ${intra['Curtail value ($)'].values[0]}")
```

```
Always operating $518440
Least Squares Monte Carlo Payout $540677
Curtail Value $22237
```

This 3D surface plot illustrates how the net payout from a BTC mining operation varies with changes in BTC price volatility and electricity price volatility. Generally, we observe that higher BTC volatility tends to increase the expected payout. This is unexpected as we would think greater volatility in BTC would decrease the net payouts or when it is optimal to exercise. In contrast, increased electricity volatility introduces more uncertainty in operating costs, often leading to lower or more erratic payouts due to the risk of unexpected price spikes. However the surface shows several sharp peaks, indicating that certain combinations of volatilities can lead to significantly higher payouts, though these can be driven by random shocks in the model. Although the randomization and shocks may oppose the explaination

New Node: ODEL

[54]:
```
odel = sim.run_sim(seed=None, intraday=False, analysis = False,␣
 ↪plot_vol_surface=False, Node='ODEL')
```

```
LSMC optimal NPV : $796,504
```

[61]:
```
#Average Annual Profit
profit = odel['npv_lsm']

#Volatility in profit
vol = odel['npv_lst_std']

#Sharpe
sharpe = profit / vol

print(f"Average Profit: ${profit:.2f}")
print(f"Volatility: {vol:.0f}")
print(f"Sharpe Ratio: {sharpe:.2f}")
```

```
Average Profit: $796503.78
Volatility: 646729
```

```
Sharpe Ratio: 1.23
```

“ “ “What node did you choose and why? How does this investment compare to doubling the size of the installation at your uncle’s original location (assuming the same capital expenditure in each case)? What can you learn from this difference” “ ”