Maximizing Wind Energy Value: Optimization Under Dynamic Pricing & Constraints

Rina Forristal - 2025/01/25

OUTLINE

The Problem: A Timing Mismatch

Proposed Solution: Energy Optimization

Solution Outline: Algorithm Basics

Decision Making Process: Linear Optimization

Overview of Linear Programming

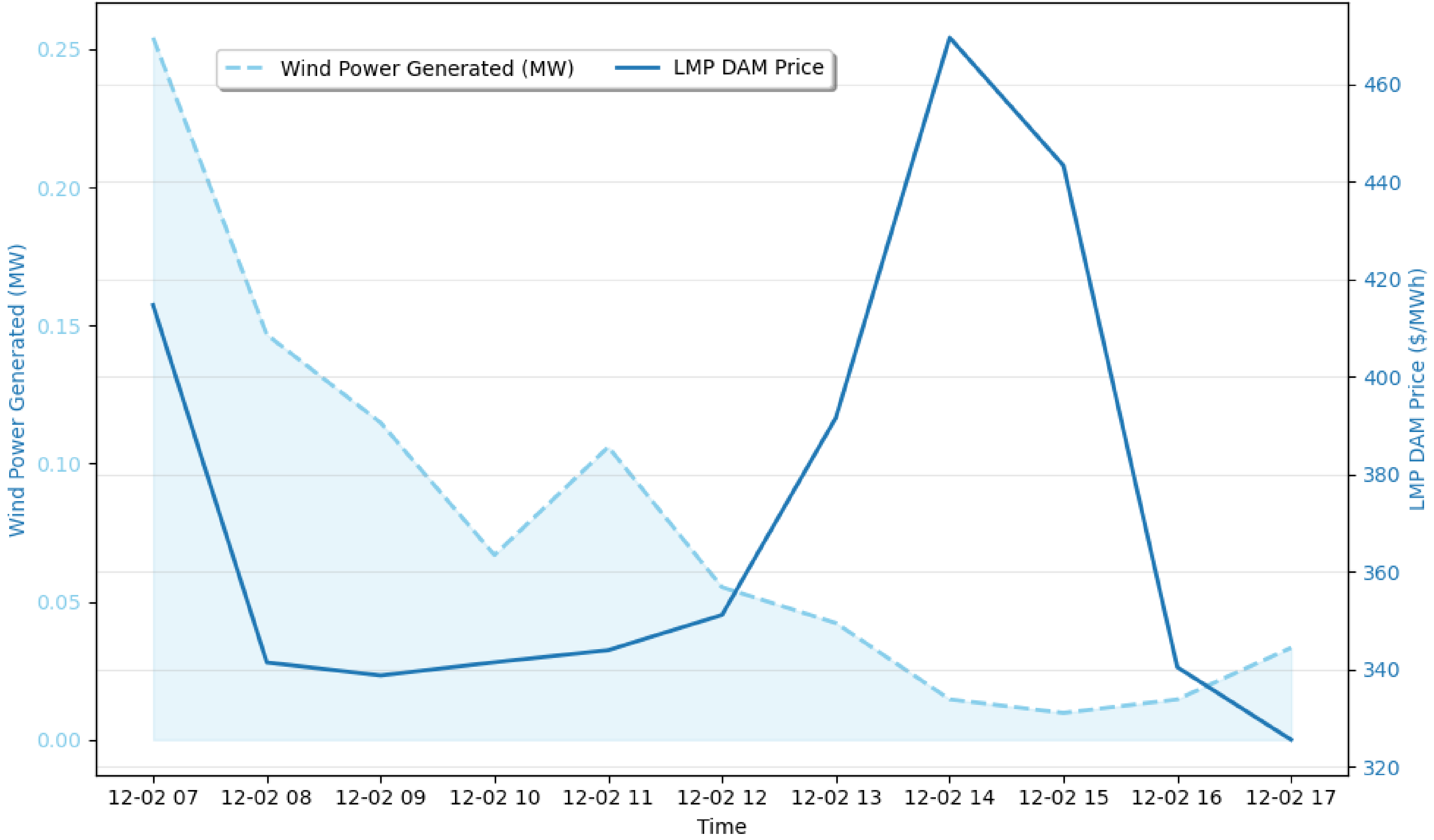
Peak Demand Shifting In Action

Battery Storage Optimization

Practical Applications

THE PROBLEM: A TIMING MISMATCH

Peak energy prices and production levels frequently do not align.



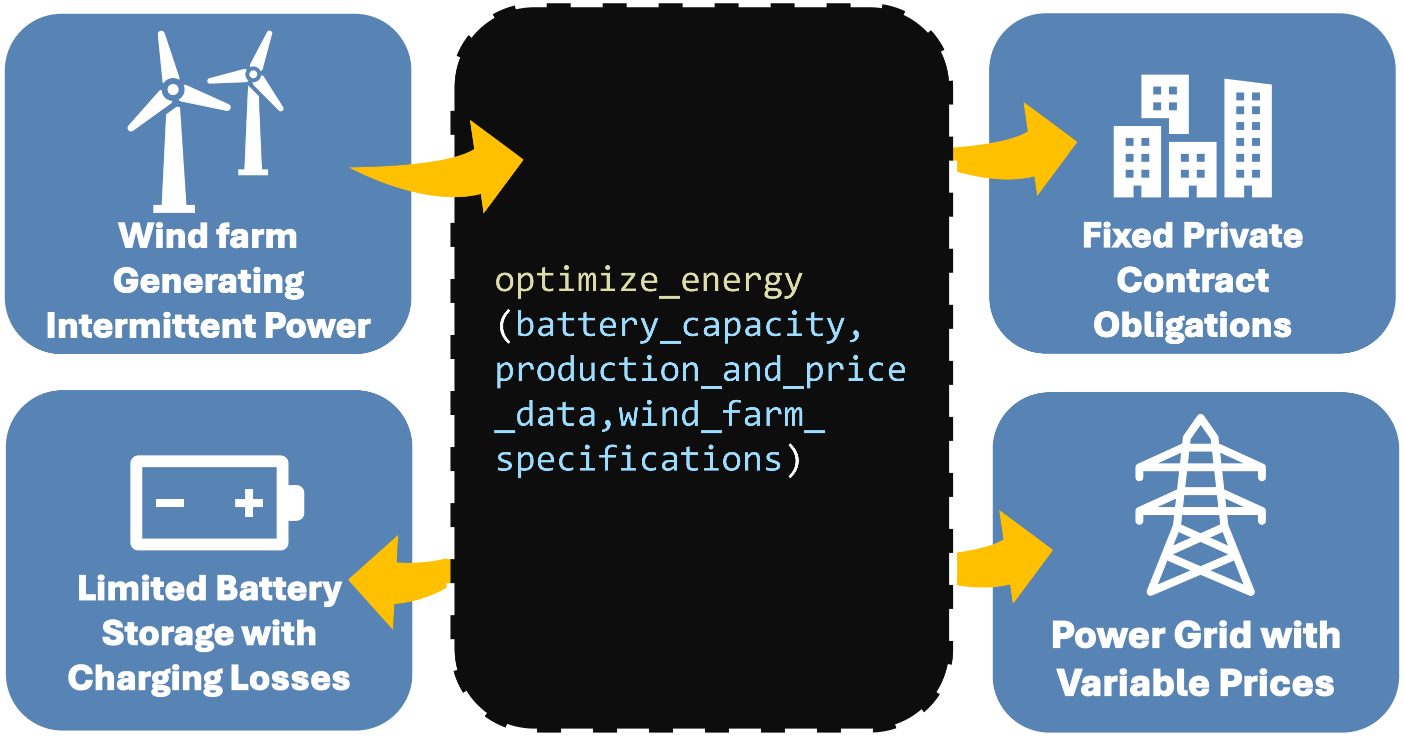
OBJECTIVE AND OVERVIEW

Large fluctuations in Locational Marginal Prices (LMP) on the Day Ahead Market (DAM) creates opportunities for arbitrage.

Objective

Our solution automates energy management and enables peak demand shifting by using linear programming to optimize storage utilization and energy sales. It ensures efficient charging, discharging, and sales strategies, while adhering to capacity limits, charging constraints, and private contract obligations. Data from California ISO's (CAISO) Open Access Same-time Information System (OASIS) and NREL's Wind Resource Database accounts for the variability in power generation and energy prices.

Model Dynamics



1) http://oasis.caiso.com/mrioasis/logon.do
2) https://wrdb.nrel.gov/data-sets/us-data

THREE STEP SOLUTION

At the heart of every successful energy storage strategy is a well-tuned decision-making engine.

Inputs: A data frame containing hourly wind power and price

```python
weather_df['power at 100m (MW)'] = (
0.5 * Cp * weather_df['air density at 100m (kg/m³)'] * A * (weather_df['wind speed at 100m (m/s)']**3) / 1e6)
merged_df = pd.merge(weather_df, LMPDAM_prices, on='datetime', how='inner')
input_data=merged_df[['datetime','MW','power at 300m (MW)','power at 200m (MW)','power at 100m (MW)']]
input_data[0:1]
```

[369] ✓ 0.0s Python

...

	datetime	power at 300m (MW)	power at 200m (MW)	power at 100m (MW)	Price Per MW
0	2022-01-01 07:00:00	2.45	2.47	2.38	66.65

Optimization: Automated!

```python
# Extract parameters
wind_power = merged_df[power_at_height].values * params['n']…
# Define decision variables
charge = pulp.LpVariable.dicts("Charge", hours, lowBound=0, upBound=charge_C_rate*storage_capacity, cat='Continuous')…
# Define the optimization problem
prob = LpProblem("Energy_Optimization", LpMaximize)
prob += pulp.lpSum((dam_prices[h] * sell_grid[h] +  private_price * sell_private[h] + dam_prices[h] * discharge_grid[h]
+ private_price * discharge_private[h] - charge[h] * storage_efficiency) for h in hours
# Constraints
for h in hours:prob += charge[h] <= wind_power[h] - sell_private[h] - sell_grid[h] prob += discharge_grid[h]…
```

Actionable Outputs: Variables, neatly translated

```python
pulp.LpVariable.dicts("SellPrivate", hours, lowBound=0, cat='Continuous')
pulp.LpVariable.dicts("SellGrid", hours, lowBound=0, cat='Continuous')
pulp.LpVariable.dicts("Charge", hours, lowBound=0, upBound=charge_C_rate*storage_capacity, cat='Continuous')
pulp.LpVariable.dicts("DischargeGrid", hours, lowBound=0, upBound=discharge_C_rate*storage_capacity, cat='Continuous')
pulp.LpVariable.dicts("StoredEnergy", hours, lowBound=0, upBound=storage_capacity, cat='Continuous')
pulp.LpVariable.dicts("DischargePrivate", hours, lowBound=0, upBound=discharge_C_rate*storage_capacity,cat='Continuous')
results_df = pd.DataFrame(results)
```

DECISION MAKING PROCESS

```
# Objective Function

pulp.lpSum()
```

We're working to find:
Maximum annual profits of variable-priced and fixed-contract electricity sales while minimizing energy losses during storage and charging.

$$\text{Maximize } Z = \sum_{h=1}^{H} \left(P_{\text{grid},h} \cdot E_{\text{sold, grid},h} + P_{\text{private}} \cdot E_{\text{sold, private},h} + P_{\text{grid},h} \cdot E_{\text{discharged, grid},h} + P_{\text{private}} \cdot E_{\text{discharged, private},h} - C_{\text{charge},h} \cdot C_{\text{efficiency}} \right)$$

```
#Constraints

for t in time:
prob +=
```

Subject to Key Constraints:
Power balances, charging/discharging constraints, storage dynamics and capacity.

$$E_{\text{sold, grid},h} + E_{\text{sold, private},h} + C_{\text{charge},h} = E_{\text{wind},h} \quad \text{(Wind generation)} \quad 0 \leq E_{\text{discharged, grid},h}, E_{\text{discharged, private},h} \leq C_{\text{rate,discharge}} \cdot S_{\text{capacity}}$$

$$S_{\text{stored},h} = S_{\text{stored},h-1} + C_{\text{charge},h} \cdot C_{\text{efficiency}} - E_{\text{discharged, grid},h} - E_{\text{discharged, private},h} \qquad 0 \leq C_{\text{charge},h} \leq C_{\text{rate,charge}} \cdot S_{\text{capacity}}$$

```
#Store Results

return
pd.DataFrame
```

So that we get:
Calculated and neatly stored optimized decision results for every hour in a year and calculate financial summary statistics. The mess from the previous slide, now nice and organized:

Energy Sold to Private Market	Energy Discharged to Private Market	Energy Sold to Grid	Energy Charged	Energy Discharged to Grid	Energy Stored	Wind Power Generated	LMP DAM Price

HOW ITS DONE: LINEAR PROGRAMMING WITH PULP

Matrix Representation

The objective function (Revenue -Efficiency) is represented as a vector multiplication:

$$Z = \mathbf{c}^T \mathbf{x}$$

Constraints (real world dynamics) are represented as matrices:

$$\begin{aligned} &\mathbf{Ax} \leq \mathbf{b} \quad \text{(Inequality constraints)} \\ &\mathbf{Cx} = \mathbf{d} \quad \text{(Equality constraints)} \\ &\mathbf{x} \geq 0 \quad \text{(Non-negativity constraints)} \end{aligned}$$

Optimization Solver

Feasibility: Optimization begins by identifying an initial solution that satisfies all constraints.

Iterative Improvement:
Through iterative steps, the solver refines the solution, navigating the edges of the feasible region to find better values.

Achieving Optimality: The process concludes when the maximum value of the objective function under the given constraints is identified.

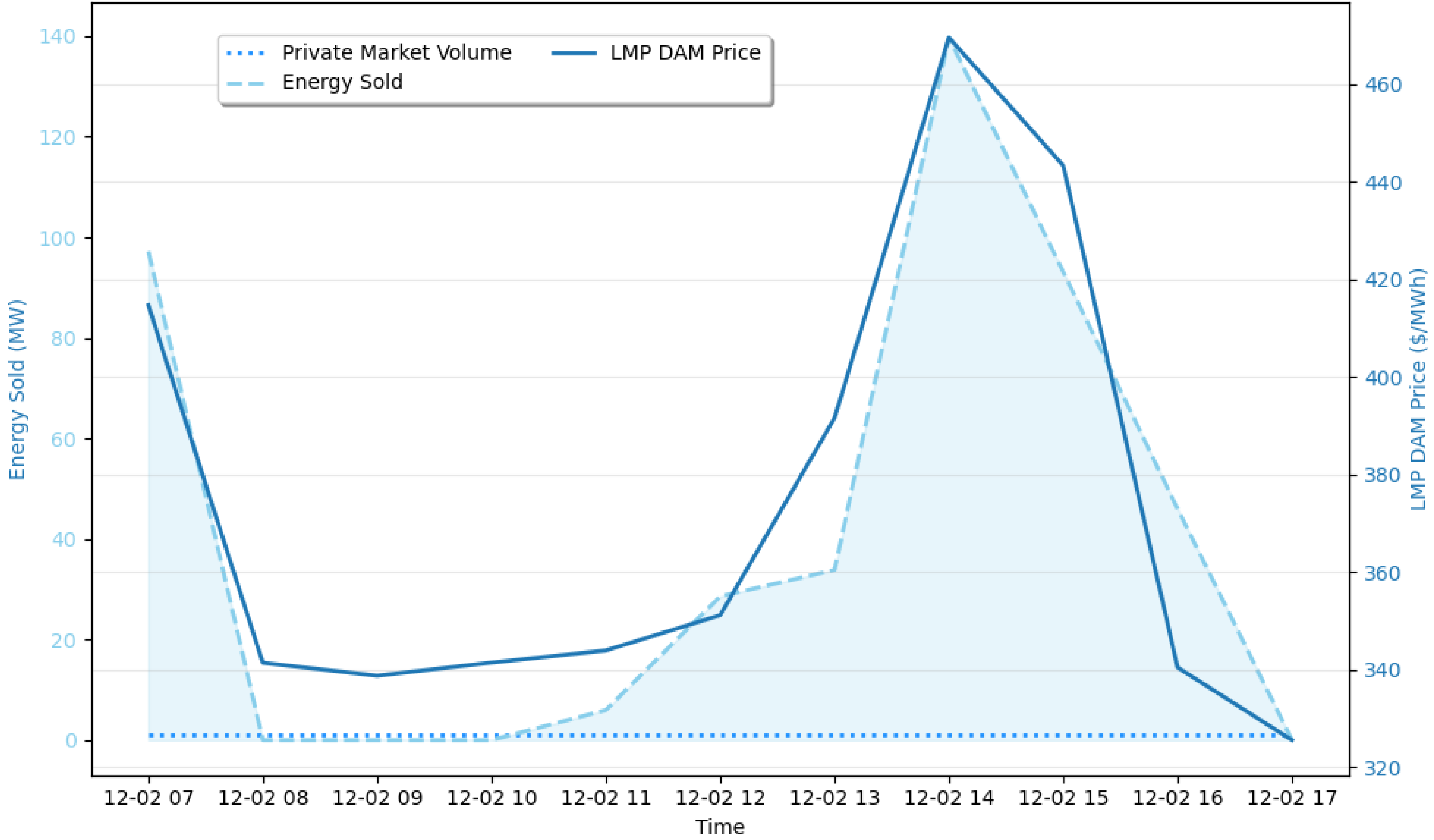
Constraint Compliance

Loop Through Each Hour:
The testing process checks each hour of the optimization results to ensure all constraints are met. Key values such as energy sold, charge, discharge, and stored energy are compared against their constraints.

Error Collection:
If a constraint is violated, an error message is logged with the violated constraint and hour to keep the model within limits.

PEAK DEMAND SHIFING

Optimizing energy storage and timing ensures energy is sold when its most valuable.



OPTIMAL BATTERY STORAGE

Storage capacity is key to unlocking the full potential of demand-shifting strategies

The Solution:
An algorithm that models a customizable wind farm and battery setup, evaluating a range of capacities to pinpoint the optimal size for maximizing profit growth.

Inputs

```
parameters = {
'charge_C_rate',
'discharge_C_rate',
'number_of_windmills',
'battery_power_cost_per_kw',
'battery_energy_cost_per_kwh',
'discount_rate',
'degradation_cost_per_cycle',
'battery_capacity'
}
```

```
def
calculate_unoptimized_revenue(results_df
, parameters)
"Calculate revenue without optimization"
```

```
def calculate_degradation_cost(
results_df, capacity, parameters):
"""Calculate degradation cost based on
cycles used."""
```

```
def npv_calculation(net_profit,
discount_rate, periods):
    """Calculate Net Present Value
(NPV)."""
```

```
def
find_best_storage_capacity(merged_df,
parameters, storage_range: step_size):
```

Output

Metric	Value
Optimal Storage Capacity (MWh)	30.00
Profit ($)	219,571,954.48
NPV ($)	0.00
Payback Period (years)	inf
Increase in profit ($)	0.00

Energy storage is key to demand-shifting strategies, but today's battery systems remain too costly to be economically viable. This creates a bottleneck for grid optimization despite recent cost declines.

PRACTICAL APPLICATIONS

Forecasting & Back Testing

The algorithm can optimize allocation of forecasted wind and LMPs to maximize value of energy. Back testing and optimization helps in mitigating risks by identifying potential flaws and weaknesses in a trading strategy before implementing them. This analysis leads to more informed decision-making, aligning strategies with real market conditions and improving revenue predictability.

Location Evaluations

Choosing the right site and configuration is critical to maximizing the potential of wind energy projects. Using the NREL's Wind Resource Database this algorithm can simulate energy production to assess wind farm performance across various locations, turbine heights, and configurations to identify the most cost-effective site setups.

https://wrdb.nrel.gov/data-viewer

Specifications & Investments

What role can this play as a decision-making tool for energy projects? By simulating energy production and profitability under various scenarios, it provides insights into the potential impact of adopting advanced technologies or expanding capacity. By evaluating financial benefits of upgrades and it enables companies to allocate resources, prioritize projects with the highest return on investment, and confidently scale operations.

MY EXPERIENCE

4th Year student at the University of British Columbia Bachelor's of International Economics (BIE) graduating Spring of 2025. Educational concentration in Finance, Statistics and Computer Science . Experience in in financial modeling, asset valuation, and algorithm development.

SG STEWART GROUP

- Conducted asset valuations for projects over CAD $2.05B, delivering forecasts to guide executive decisions to address a CAD $600M funding deficit.
- Produced feasibility assessments and criteria for acquisitions.
- Developed negotiation frameworks and funding strategies for successful agreement execution.
- Prepared executive-ready materials for TransLink's C-suite.

Beach Algorithms

- Developed an automated stock trading algorithm and interactive analytics platform to track and trade NYSE-listed companies, integrating news sentiment, template portfolios, financials, dividends, technical indicators, options, ratings, etc.
- Achieved 95.4% back testing accuracy and 26.36% annualized return over the past year.

UBC THE UNIVERSITY OF BRITISH COLUMBIA

- Reviewed and graded over 350 assignments, exceeding average TA outputs by 73% while receiving great student feedback.
- Identified curriculum gaps and independently developed supplementary materials, including lecture slides and speaking notes for CPSC 430: Ethics for Software Engineering.

MY CONTACT INFO

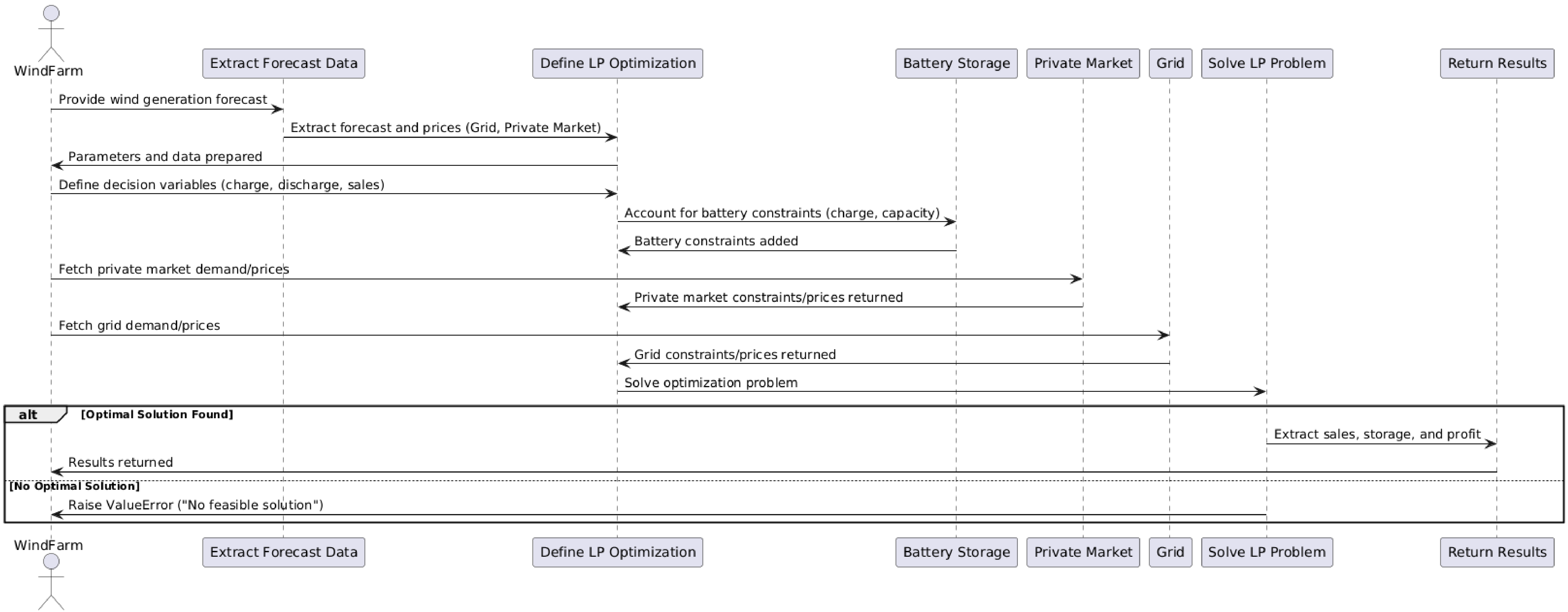
Rina.smf@gmail.com

+1 (778) 938 9763

https://github.com/rinaforristal/WindEnergyOptimization/tree/master

APPENDIX 1

Energy Management Optimization Sequence Diagram



https://plantuml.com/ascii-math

APPENDIX 2

Sample Energy Optimization Output: Financial Summary

```
params = {
    'power_at_height’:'power at 100m (MW)',
    'private_market_demand': 1,
    'private_price': 75,
    'storage_efficiency': 0.9,
    'charge_C_rate': 0.5,
    'discharge_C_rate': 1,
    'n': 100,
    'battery_power_cost_per_kw': 100,
    'battery_energy_cost_per_kwh': 100,
    'discount_rate': 0.05,
    'degradation_cost_per_cycle': 20,
}

max_capacity=200
```

Metric	Value
Total Production (MWh)	2,475,141.95
Max Production (MWh)	3,025.82
Min Production (MWh)	0.00
Max Price ($/MWh)	1,336.97
Min Price ($/MWh)	-4.70
Revenue Change from Baseline ($)	5,244,171.38
Profit ($)	226,992,347.76

APPENDIX 3

Sample Energy Optimization Output: Decision Variables for Every Hour of the Year

	Energy Sold to Private Market	Energy Discharged to Private Market	Energy Sold to Grid	Energy Charged	Energy Discharged to Grid	Energy Stored	Wind Power Generated	LMP DAM Price
0	1.00	0.00	237.07	0.00	0.00	0.00	238.07	66.65
1	1.00	0.00	128.24	50.16	0.00	45.15	179.41	62.66
2	1.00	0.00	141.83	0.00	0.00	45.15	142.83	64.72
3	1.00	0.00	0.00	99.34	0.00	134.55	100.34	60.83
4	1.00	0.00	0.00	72.72	0.00	200.00	73.72	61.59
5	1.00	0.00	55.26	0.00	0.00	200.00	56.26	63.78
6	1.00	0.00	52.97	0.00	0.00	200.00	53.97	79.10
7	1.00	0.00	61.05	0.00	41.14	158.86	62.05	87.81
8	1.00	0.00	34.41	0.00	0.00	158.86	35.41	81.11
9	1.00	0.00	0.00	17.25	0.00	174.39	18.25	59.49
10	1.00	0.00	0.00	15.15	0.00	188.03	16.15	44.93
11	1.00	0.00	0.00	11.01	0.00	197.94	12.01	43.41
12	1.00	0.00	0.00	2.29	0.00	200.00	3.29	31.80
13	0.53	0.47	0.00	0.00	0.00	199.53	0.53	30.73
14	0.15	0.85	0.00	0.00	0.00	198.68	0.15	32.92
15	0.01	0.99	0.00	0.00	0.00	197.69	0.01	35.75
16	0.01	0.99	0.00	0.00	0.00	196.70	0.01	58.79
17	0.66	0.34	0.00	0.00	0.00	196.36	0.66	87.53
18	1.00	0.00	1.39	0.00	98.18	98.18	2.39	100.26
19	1.00	0.00	2.30	0.00	49.09	49.09	3.30	102.41
20	1.00	0.00	8.31	0.00	24.55	24.55	9.31	96.37

APPENDIX 4

Sample Battery Capacity Optimization Output

	Storage Capacity (MWh)	Profit ($)	NPV ($)	Payback Period (years)	Unoptimized Earnings ($)	Increase in Profit ($)	Battery Cost ($)	Total Revenue ($)	Degradation Cost ($)
0	10	221,011,963.39	0.00	inf	221,112,851.38	-100,888.00	1001000	222,036,308.02	23,344.64
1	20	220,294,144.68	0.00	inf	221,112,851.38	-818,706.70	2002000	222,319,032.62	22,887.94
2	30	219,571,954.48	0.00	inf	221,112,851.38	-1,540,896.90	3003000	222,597,516.84	22,562.36
3	40	218,845,780.03	0.00	inf	221,112,851.38	-2,267,071.35	4004000	222,872,123.81	22,343.78
4	50	218,117,151.31	0.00	inf	221,112,851.38	-2,995,700.08	5005000	223,144,309.15	22,157.85
5	60	217,385,855.98	0.00	inf	221,112,851.38	-3,726,995.41	6006000	223,413,843.44	21,987.47
6	70	216,652,273.51	0.00	inf	221,112,851.38	-4,460,577.87	7007000	223,681,106.22	21,832.71
7	80	215,916,181.10	0.00	inf	221,112,851.38	-5,196,670.29	8008000	223,945,873.05	21,691.95
8	90	215,177,529.80	0.00	inf	221,112,851.38	-5,935,321.59	9009000	224,208,092.18	21,562.38
9	100	214,436,848.49	0.00	inf	221,112,851.38	-6,676,002.89	10010000	224,468,260.62	21,412.13
10	110	213,694,458.91	0.00	inf	221,112,851.38	-7,418,392.48	11011000	224,726,718.95	21,260.04
11	120	212,950,556.54	0.00	inf	221,112,851.38	-8,162,294.84	12012000	224,983,676.13	21,119.59
12	130	212,205,252.26	0.00	inf	221,112,851.38	-8,907,599.12	13013000	225,239,237.57	20,985.31
13	140	211,458,687.58	0.00	inf	221,112,851.38	-9,654,163.80	14014000	225,493,539.30	20,851.72
14	150	210,710,820.43	0.00	inf	221,112,851.38	-10,402,030.96	15015000	225,746,549.30	20,728.87
15	160	209,961,935.30	0.00	inf	221,112,851.38	-11,150,916.09	16016000	225,998,547.46	20,612.16
16	170	209,211,564.84	0.00	inf	221,112,851.38	-11,901,286.54	17017000	226,249,062.65	20,497.81
17	180	208,459,827.91	0.00	inf	221,112,851.38	-12,653,023.47	18018000	226,498,208.63	20,380.71
18	190	207,706,885.08	0.00	inf	221,112,851.38	-13,405,966.30	19019000	226,746,147.14	20,262.05
19	200	206,952,202.09	0.00	inf	221,112,851.38	-14,160,649.29	20020000	226,992,347.76	20,145.67
20	210	206,196,733.55	0.00	inf	221,112,851.38	-14,916,117.84	21021000	227,237,789.22	20,055.68