Problem Statement

Imagine you are a control engineer who is tasked with designing a controller for a chiller plant with the objective of minimizing the total chiller power consumption over a 24-hour operating period, while ensuring the building's cooling load is fully satisfied at all times. The controller will receive a 24-hour forecast of the building's cooling demand (measured in tons) and historical operational data from the plant. The plant consists of seven chillers, four 1000-Ton chillers (CH-1 to CH-4), two 500-Ton chillers (CH-6 and CH-7), and one 250-Ton chiller (CH-8). The chilled water supply temperature (CHS) from each chiller is assumed to perfectly track the controller's setpoint, which must remain within the range of 44 to 49 ℉. The condenser water supply temperature (CDS) is fixed at 82 ℉. The system also includes operational timing constraints: chillers can only be turned ON between 01:00 and 12:00, and can only be turned OFF between 12:00 and 22:00. Furthermore, once a chiller is turned ON, it must remain ON for at least 4 consecutive hours. In addition, it is required that the average CHS of 45 ℉ must be supplied to the building at each hour. Your controller's task is to determine, for each hour of the day, the ON/OFF status of each chiller and the CHS setpoint for each active chiller. The solution must adhere to all specified constraints and aim to minimize the total energy consumed by the chillers throughout the 24-hour period. You are expected to design the control strategy using your own modeling approach, based on the data provided. The controller should respect all operational constraints while minimizing total energy consumption across chillers and pumps.

Import Data

⚠ Note: Upload file "dataset_for_chiller_control.cvs"

(Open item Files at left side and drag file "dataset_for_chiller_control.cvs" in to it)

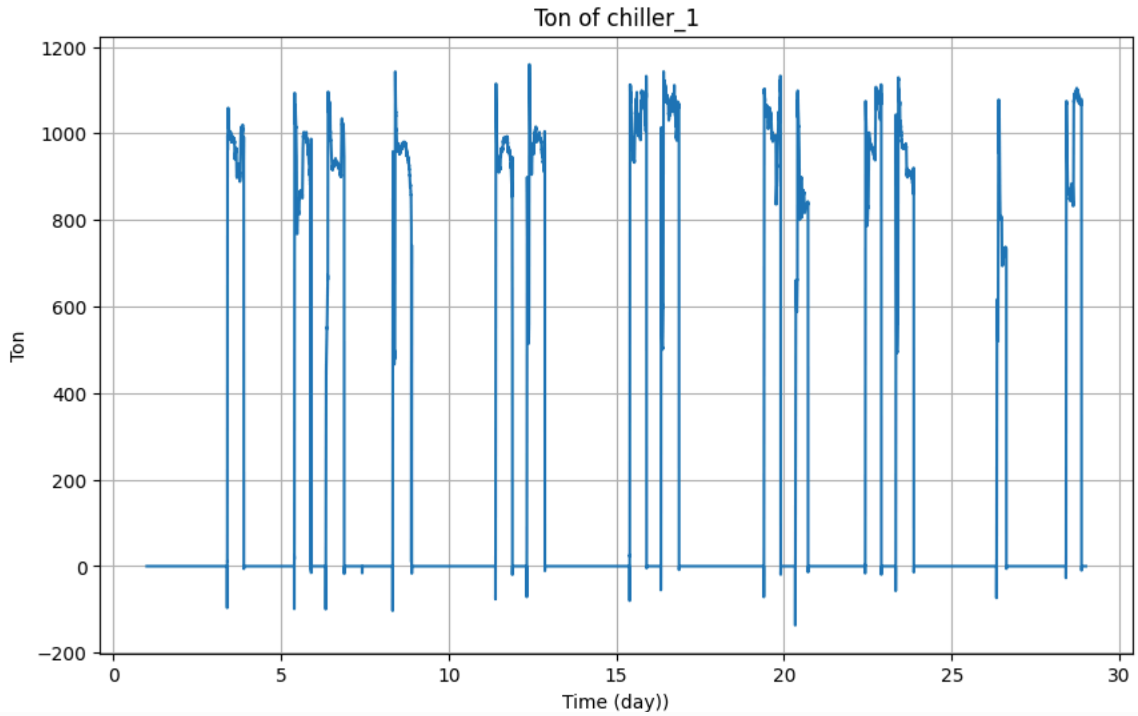
```
import csv
import numpy as np
import matplotlib.pyplot as plt
from datetime import datetime, timedelta
from scipy.optimize import minimize
import pandas as pd
import seaborn as sns
import cvxpy as cp
```

Show code

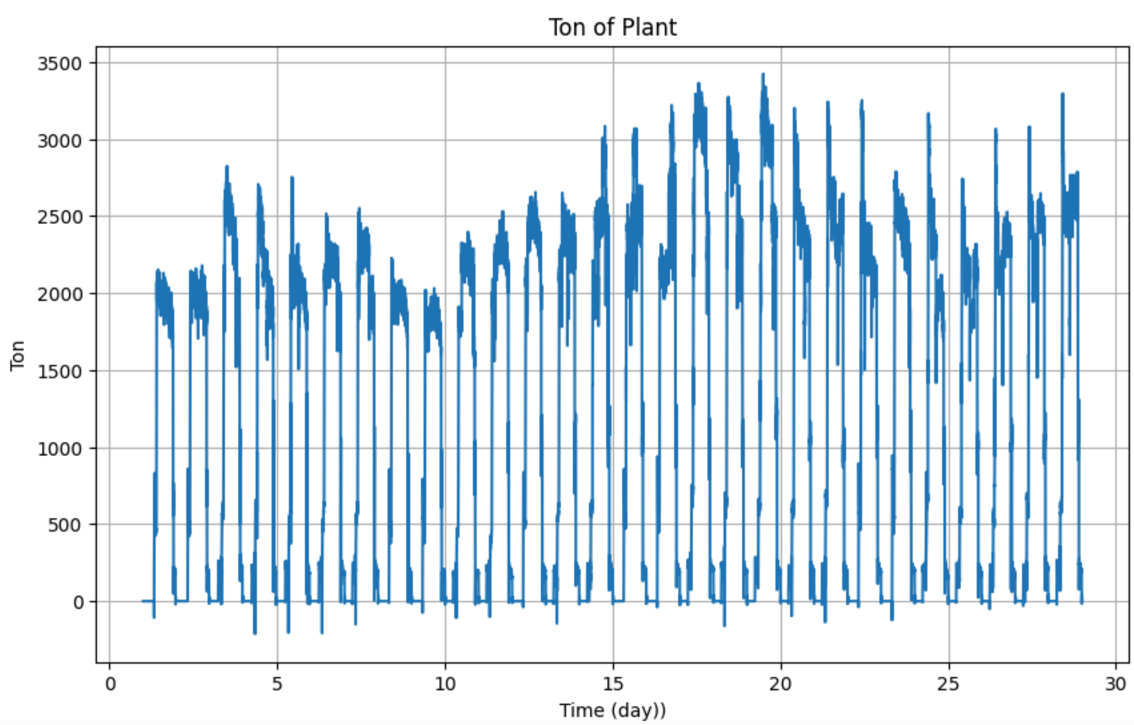
```
var_name = ['CDS',  'CDR',  'Ton',  'CHR',  'CHS',  'kW',   'Setpoint', 'On-Off']
```

Show code

Show code

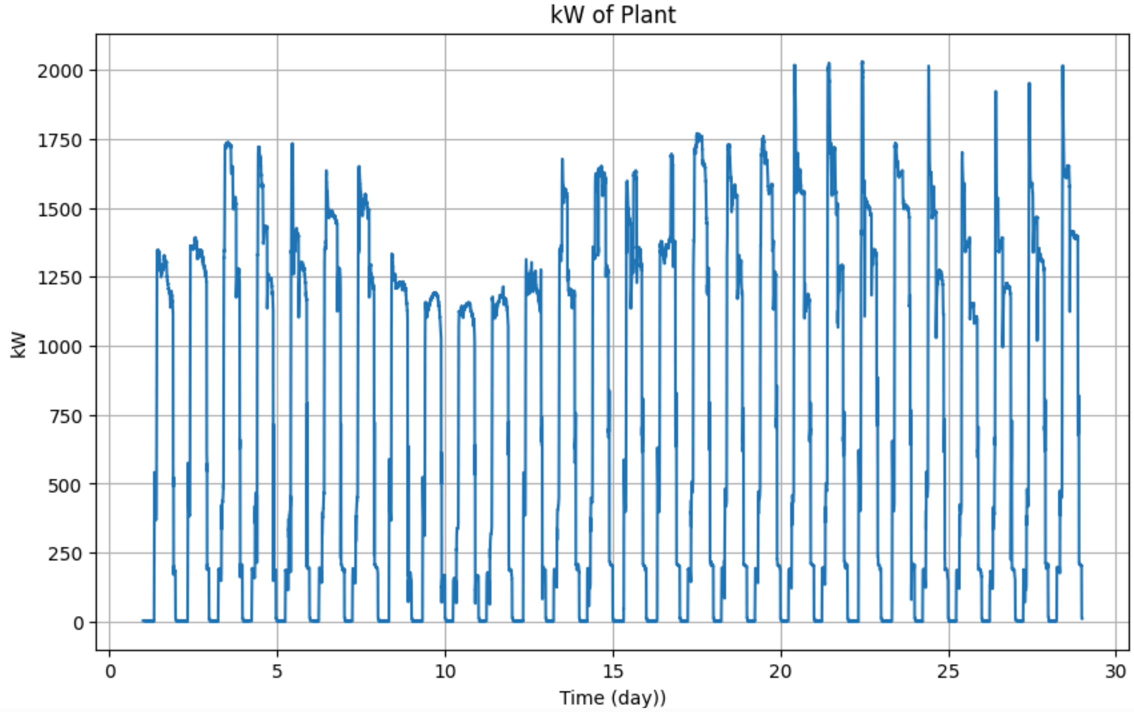



```
plant_ton = plant_data[:, 0]
if len(processed_times) == len(plant_ton):
    plt.figure(figsize=(10, 6)) # Optional: Set the figure size
    plt.plot(processed_times, plant_ton)
    plt.xlabel("Time (day))")
    plt.ylabel("Ton")
    plt.title("Ton of Plant")
    plt.grid(True) # Optional: Add a grid
    plt.show()
```




```
plant_kW = plant_data[:, 1]
if len(processed_times) == len(plant_kW):
    plt.figure(figsize=(10, 6)) # Optional: Set the figure size
    plt.plot(processed_times, plant_kW)
    plt.xlabel("Time (day))")
```

```
plt.ylabel("kW")
plt.title("kW of Plant")
plt.grid(True) # Optional: Add a grid
plt.show()
```



Section 1: Literature Review

Objective: Demonstrate your understanding of prior work and applicable methods through literature review.

Questions:

● Briefly review several research papers and select 2–5 approaches you think are the most practical when it comes to real-world implementation for energy-efficient control in large HVAC or water-cooled chiller plant systems.

● For your literature review, please make a comprehensive comparison and discussion. This may include

- Key assumptions and components
- How would they handle constraints of a chiller plant?
- Any other aspect worth discussing

General Key Assumptions

- The problem want to minimize electric cost and hardware cost of chiller
- The performance of chiller depend on itself and consistance
- Recive Demand power of chiller in each time of day

Note:

t index of time

i index of chiller

Mix Interger Linear Programing (MILP)

- **Key Assumpsion**
  - The Objective function and constant must be formulate as **Linear equation** and the system has static property (the chiller always working well as it was, COP of Chiller remain same)

$$kW_{it} = a_i + b_i P_{evap,ti} + c_i T_{o,t}$$

calcualtion in [1] show that linear appoximation of power consupsion have value $R^2 > 0.98$ (Goodness of fit)

$$\text{Objective Function} = \sum_i \sum_j kW_{it}$$

  - Operating Cost and Turn-on cost is consant
  - Recive Demand power of chiller in each time of day
- **How would they handle constraints of a chiller plant?**
  - Power evaporate (Ton) Constraint could be handled by a normal constant, and on-off chiller could handle by binary variable, with specific constraint to be linear function of power evaporate of chiller and slack variable of chiller's status.

$$g(P_{evap}, x) \leq h$$
$$g_{eq}(P_{evap}, x) = h_{eq}$$

- **Any other aspect worth discussing**
  - The MILP is the most simple method compare to other model optimizaton and its limitaion is unablity to handle complicate system and constrain. However, this simple method provide low cost of computation and less understanding of complicate system.
  - If the system has differnt coffient (COP, operating cost, Turn-on cost) as expected. we could feedback it and update to recalculation at that time or next day.

Genetic Algorithm(GA)

- **Key Assumpsion**
  - The Objective function could be formulated in a complicated function, even **nonlinear**

$$kW_{it} = f(P_{evap,it})$$

    Form [2], have relation of power consupsion as

$$kW_{it} = a_i' + b_i' P_{evap,it} + c_i' P_{evap,it}^2 + d_i' P_{evap,it}^3$$

  - The operating variable and chiller cooling power could encode in chromosome structure in GA. [2] represent the chromosome of $P_{evap,it}$ as encoded binary value.
  - The operation of genetic algorthim**(Reproduction,Crossover,Mutation)** could produce better solution in next generation
- **How would they handle constraints of a chiller plant?**
  - [2] handle constraint in form of objective fucntion

$$\text{Objective Function} = \text{Objective Function}_0 + \text{Penalty Function}$$

    where the Penalty Function is fuction that if return high value if the variable is leaked of constraint to converge the solution to feasiable space.
  - For certain constraints in the given problem, we can incorporate the specifications directly into the definition of the chromosome, leading to reduced computational complexity. For example, an on-off constraint can be handled by formulating the chromosome as a list of status values. This allows us to represent the timing of on/off events explicitly, based on assumptions about one-time turns on per day of chiller. Example:

$$S = [0\ 0\ 1\ 1\ 1\ 1\ ....\ 0\ 0]$$
$$S = [1\ 2\ 13]$$

- **Any other aspect worth discussing**
  - Genetic algorithm (GA) to solve optimal chiller loading (OCL) problem. GA overcomes the flaw that with the Lagrangian method the system may not converge at low demand.[2]
  - The disadvantage of heuristic optimization, like GA, is that it has high computation; however, it can handle complex objective functions and constraints.
  - [3] use GA as initial population for **Grey Wolf Optimizer** and **JAYA algorithm** for specificing in last minimize objective and overcoming convergence issues.

Partical Swarom Optimization(PSO)

- **Key Assumpsion**
  - The Objective function could be formulated in a complicated function, even **nonlinear**

$$kW_{it} = f(P_{evap,it})$$

    Form [4], have relation of power consupsion as

$$kW_{it} = a_i' + b_i' P_{evap,it} + c_i' P_{evap,it}^2 + d_i' P_{evap,it}^3$$

Form [5], have relation of power consupsion as

$$kW_{it} = a'_i + b'_i P_{evap,it} + c'_i P^2_{evap,it}$$

- The value of variable could be written as space of particle
- The evolution that references on global best position and the history best psotion lead to conversion of solution of problem.

- **How would they handle constraints of a chiller plant?**
  - Penaty function as GA, if the particle leak from constraints the objective fucntion would increas in large amuont for conversion. [4]
  - Re-initilaize the leaked particle. [4]

- **Any other aspect worth discussing**
  - PSO has shown effectiveness in both overcoming convergence issues and finding minimum energy solutions [4]

Section 2: Problem Formulation

Objective: Translate the control problem into a formal model-based optimization control problem.

Questions:

- Define the decision variables, models, objective function, and constraints for this problem.
- Formulate the problem as a mathematical program (e.g., mixed-integer or constrained optimization), incorporating the constraints

Optional: Suggest how your formulation might be scalable or generalizable to larger systems.

Formualte Problem

$$\min_{\vec{v},\vec{S},\vec{P},\vec{U}} \quad \mathrm{TOC}^\top \vec{v} + \frac{1}{n}\mathrm{OPC}^\top \vec{S} + \frac{1}{n}(i\mathrm{COP})^\top \vec{P}$$

$$\begin{aligned}
\text{s.t.} \quad & P_t \preceq P_{\max}, \quad \forall t && \text{Power limitaion of chiller} \\
& \sum_{i=1}^{7} P_{i,t} \geq D_t, \quad \forall t && \text{Demand Met} \\
& U_{i,t} \leq \sum_{k=1}^{T_{min}} S_{i,t+k} && \text{Minimum duration} \\
& U_t \geq 0, \quad \forall t \in [1n, 13n] && \text{Chiller turn on in 1:00-12:00} \\
& U_t \leq 0, \quad \forall t \in [13n+1, 22n] && \text{Chiller turn on in 12:00-22:00} \\
& \sum_{\tau=t}^{\tau=t+T_{min}} S_{i,\tau} = U_{i,t} * T_{min} && \text{Status and Start indicator relation} \\
& \vec{v}_i \preceq \max(S_i), \quad \forall i && \text{Status in that day and Status relation} \\
& \textstyle\sum_{t=1}^{n_t} U_i = 0, \quad \forall i && \text{Garuntee On and Off in day} \\
\text{Optional:} \\
& P_t \succeq P_{\min}, \quad \forall t && \text{Minimum Power of chiller} \\
& P_{max}^\top S_t < 1.1 D_t, \quad \forall t && \text{Reserve Chiller Power} \\
& U_t = 0, \quad \forall t \in \text{Not Avaliable} && \text{Not avaliable on-off Chiller time}
\end{aligned}$$

Not available on-off Chiller time constraint may not be required depending on Demand profile

$T_{min} = 4n$, minumum duration for 4 hours

$\overrightarrow{v}$ is used chiller status in that day, $\overrightarrow{v} \in \mathbb{B}^7$

S is on status of chiller in each time, $S \in \mathbb{B}^{7\times n_t}$

U is start indicator of chiller in each time, $S \in \{-1, 0, 1\}^{7\times n_t}$

P is power evaparate of chiller in each time, $P \in \mathbb{R_+}^{7\times n_t}$ [ton]

$i\mathrm{COP}$ is invert element wise vector of COP [kW/ton]

$P_{max} = \begin{bmatrix} 1000 & 1000 & 1000 & 1000 & 500 & 500 & 250 \end{bmatrix}^T$ [ton]

D is demand forcast in each time, $D \in \mathbb{R_+}^{n_t}$ [ton]

n is sampling time per hour

Section 3: System Identification

Objective: Derive models from the provided data that relate control inputs to power consumption. Below is the explanation of each datapoint in the dataset

- Plant, Ton: The cooling load of the whole building. Represents the cooling demand required to cool down the indoor space.
- Chiller, CHS: Chilled water supply temperature
- Chiller, CHR: Chilled water supply temperature
- Chiller, CDS: Chilled water supply temperature
- Chiller, CDR: Chilled water supply temperature Questions:
- Describe your modeling approach. You may use:
  - Thermodynamics models
  - Regression, tree-based models, neural networks, or other ML-based methods
  - Physical-inspired black-box or gray-box models
- What are the key independent variables / features you used? and how well does your model perform based on the given data?
- Does your model's behavior exhibit meaningful physical interpretations? Please discuss your findings as well. For example, you may discuss on how different parameters affect the output of the model

$\mathrm{TOC}^\top \vec{v} + \frac{1}{n}\mathrm{OPC}^\top \vec{S} + \frac{1}{n} i\mathrm{COP}^\top \vec{P}$

What are the key independent variables?

Cooling Power of Chiller (P_{evap}/ton)

All of the research we reference says that the power consumption of the chiller depend on the polynomial function of cooling power of chiller. [1], [2], [3] and [4] are spcific that the power consumpsion of chiller could be formulate as cubic equation

$$kW_{it} = a_i + b_i P_{evap,it} + c_i P^2_{evap,it} + d_i P^3_{evap,it}$$

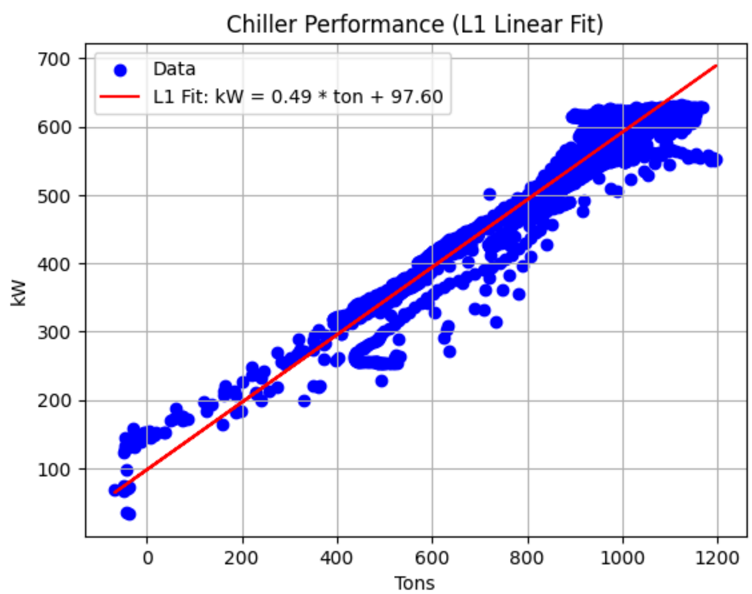
However, [1] show that with linear equation ($kW_{it} = a_i + b_i P_{evap,it}$) is enough to identify the system of chiller So we using linear programming to fit the value of the coefficients in each case:

- Linear Equation: $kW_{it} = a_i + b_i P_{evap,it}$
- Quadatic Equation: $kW_{it} = a_i + b_i P_{evap,it} + +c_i P^2_{evap}$
- Cubic Equation: $kW_{it} = a_i + b_i P_{evap,it} + c_i P^2_{evap,it} + d_i P^3_{evap,it}$

Estimate **Cofficient** form data

Show code

```
Fitted Line: kW = 0.4941 * ton + 97.5980
```



+ Code
+ Text

Linear Curve Fit

Show code

```
a_lin: [ 66.36504175  97.72185575  89.99668814  52.85261838  82.69484106
  61.77200787 107.60068989]
b_lin: [0.49499999 0.49288402 0.45626316 0.45875135 0.49717672 0.56548977
 0.17612353]
```

Quadratic Curve Fit

Show code

```
a: [ 151.38928411  -22.6296592   -57.91178992 -109.93041805  101.92442003
   15.47695686   30.15795972]
b: [0.26830008 0.80831542 0.77396087 0.79423715 0.04007997 0.81340139
 0.71839727]
c: [ 0.00014288 -0.00019547 -0.00016767 -0.00017009  0.0009418  -0.00031466
 -0.00088683]
```

Cubic Curve Fit

Show code

```
a: [137.31295867 105.40765471 118.83240772  87.25143869  98.19902781
  85.77373916 -17.74378537]
b: [0.38482348 0.50304445 0.39023374 0.47980081 0.292337   0.37993626
 0.94338862]
c: [-5.96718808e-05 -6.80862293e-05  2.82747575e-05 -1.60789996e-04
 -1.52505667e-04 -1.04597667e-04 -3.34729727e-04]
d: [ 9.97170653e-08  5.04629066e-08  8.60709140e-09  1.04376787e-07
  1.22472788e-06  8.22877258e-07 -2.66867089e-06]
```

Coefficient Table

Show code

Type	Linear		Quadratic			Cubic			
Coeff	a	b	a	b	c	a	b	c	d
Chiller1	66.365042	0.495000	151.389284	0.268300	0.000143	137.312959	0.384823	-0.000060	0.000000
Chiller2	97.721856	0.492884	-22.629659	0.808315	-0.000195	105.407655	0.503044	-0.000068	0.000000
Chiller3	89.996688	0.456263	-57.911790	0.773961	-0.000168	118.832408	0.390234	0.000028	0.000000
Chiller4	52.852618	0.458751	-109.930418	0.794237	-0.000170	87.251439	0.479801	-0.000161	0.000000
Chiller6	82.694841	0.497177	101.924420	0.040080	0.000942	98.199028	0.292337	-0.000153	0.000001
Chiller7	61.772008	0.565490	15.476957	0.813401	-0.000315	85.773739	0.379936	-0.000105	0.000001
Chiller8	107.600690	0.176124	30.157960	0.718397	-0.000887	-17.743785	0.943389	-0.000335	-0.000003

How well does your model perform based on the given data?

The overall performance of the chiller modeling can be evaluated using the R^2 (coefficient of determination) score. From the table, we observe that most chiller models perform well, with $R^2 > 0.8$, indicating a strong fit between the model predictions and actual data. However, a few chillers exhibit lower performance, suggesting that additional unmodeled variables may influence power consumption in those cases.

Among the models, the quadratic model demonstrates the best overall accuracy. Nonetheless, it underperforms in specific cases in Chiller 8. In some of these, the quadratic model yields negative coefficients or unrealistic behavior, potentially leading to non-physical predictions, such as a chiller turning on without producing cooling.

Given this, the linear model emerges as the most suitable choice for optimization. It offers a good balance between accuracy and robustness, avoids the instability observed in higher-order models, and importantly, requires significantly less computational effort than quadratic or cubic models.

R^2 Score Table

Show code

	R² Linear	R² Quadratic	R² Cubic
Chiller1	0.932360	0.940070	0.940490
Chiller2	0.919176	0.894421	0.916957
Chiller3	0.866398	0.836300	0.866366
Chiller4	0.883692	0.842671	0.880435
Chiller6	0.938722	0.959115	0.968722
Chiller7	0.913737	0.877697	0.904412
Chiller8	0.080521	0.493597	-1.517719

Next steps: Generate code with `r2_df` View recommended plots New interactive sheet

Form using linear model, we name coficient to have more physical for understanding

Estimate **Operating cost** and **COP** form data

From line fitting $kW = a \cdot ton + b$ we could see that

- b is $\mathrm{COP}^{-1} = \frac{\Delta \mathrm{kW}}{\Delta \mathrm{ton}}$ [kW/ton]
- a is operating cost or operating power consumpsion when chiller is on [kW]

Furthermore, we assume that turning on a chiller requires an initial amount of power to initiate its operation. To account for this, we define a Turn-On Cost (TOC). This cost reflects the additional energy or operational burden associated with starting a chiller. The TOC can be adjusted to provide flexibility in operation planning—for example, by discouraging frequent daily, thereby promoting more stable and efficient usage patterns.

Estimate **Turn-On Cost(TOC)** from data

$$\mathrm{TOC} = \sum (kW_t - \hat{P})/m, \quad \text{t is all time after status on for n min}$$

$$\hat{P} = ton\mathrm{COP}^{-1} + \mathrm{OPC}$$

m is status on count of chiller

Show code

```
TOC: [3.2217003  1.41742937 1.17621514 2.11068967 0.59119538 0.86668812
 0.5       ]
```

Show code

	COP[Ton/kWmin]	OPC[kW]	TOC[kWhr]
chiller_1	2.020202	66.365042	3.221700
chiller_2	2.028875	97.721856	1.417429
chiller_3	2.191718	89.996688	1.176215
chiller_4	2.179830	52.852618	2.110690
chiller_6	2.011357	82.694841	0.591195
chiller_7	1.768379	61.772008	0.866688
chiller_8	5.677833	107.600690	0.500000

Next steps: Generate code with `df` View recommended plots New interactive sheet

Section 4: Result Analysis & Baseline Comparison

Objective: Apply your model to solve for the optimal control decisions for the given dataset and compare the result with the actual controls in the dataset. Also, it would be helpful if you can estimate the potential energy savings from applying your control algorithm.

Questions:

- Using your model and controller, simulate the chiller plant operation using the given historical dataset. Show power consumption, CHS trends, and other important activities.
- Compare with a baseline strategy, as attached in the data set:
- Discuss:
  - Potential energy savings (absolute or percentage)
  - Any observed trade-offs or limitations

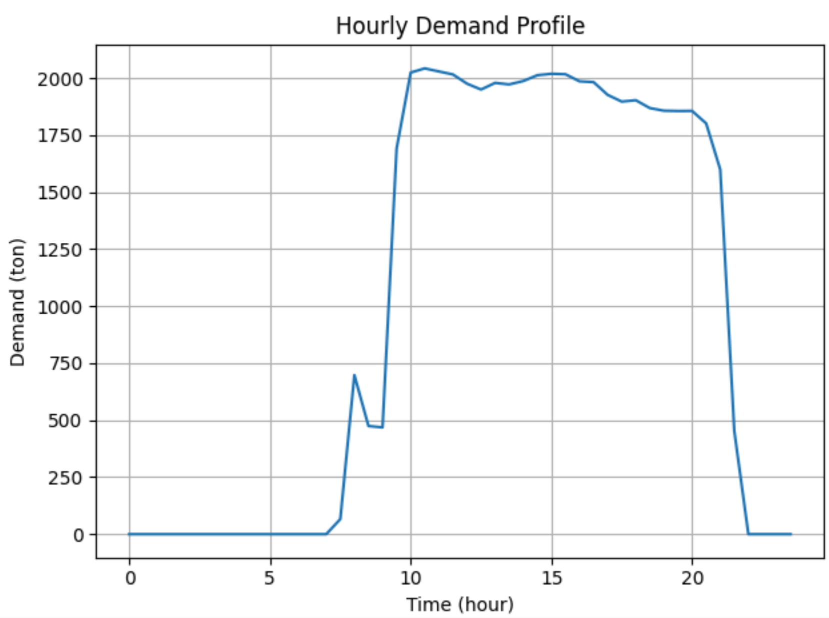
Optional: Suggest how this strategy could extend to real-time MPC deployment.

```
iCOP = 1./COP
n_chillers = 7
sam_per_hour = 2
n_t = 24 * sam_per_hour

# Example input parameters
P_max = np.array([1000, 1000, 1000, 1000, 500, 500, 250])
P_min = P_max*0.5 #Optional
T_min = 6 *sam_per_hour

S = cp.Variable((n_chillers, n_t), boolean=True)      # ON status
P = cp.Variable((n_chillers, n_t))                    # Power use
U = cp.Variable((n_chillers, n_t))                    # Start/Stop indicator
v = cp.Variable(n_chillers, boolean=True)             # Daily status
```

Simulate the chiller plant operation

Simulation with demand of day 1 data (Example)

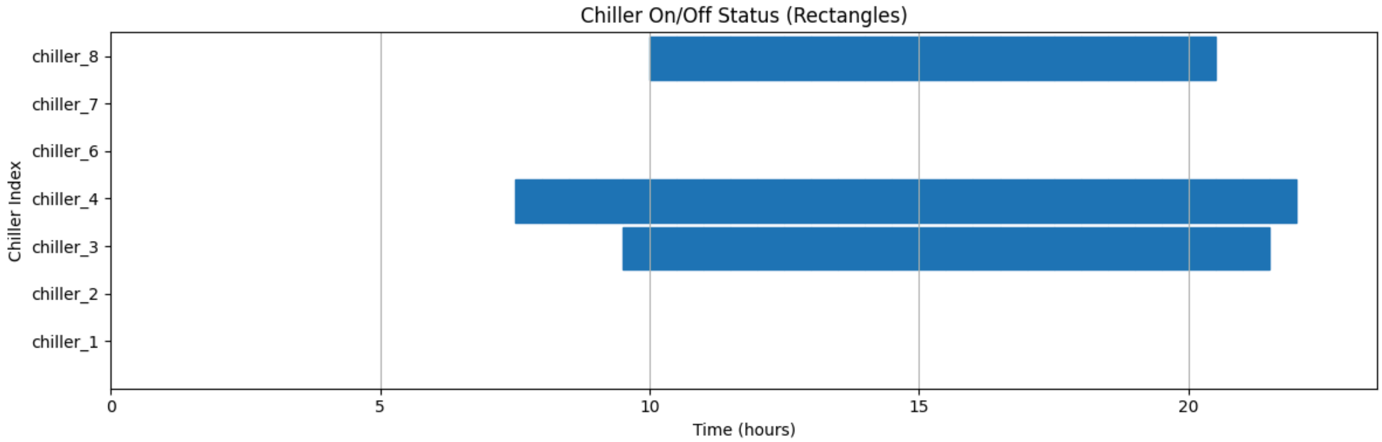
Show code

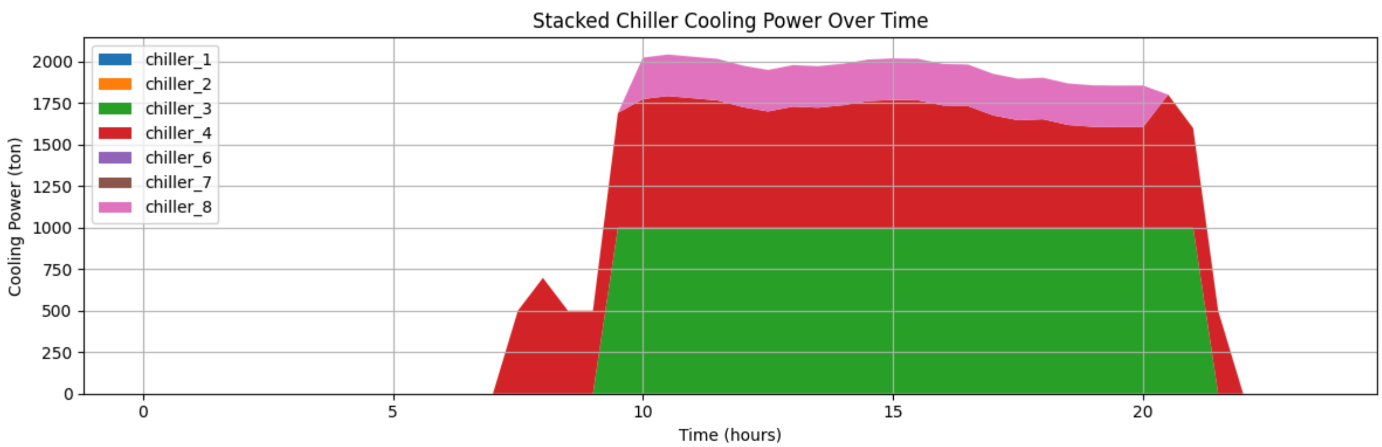


Show code

Show code

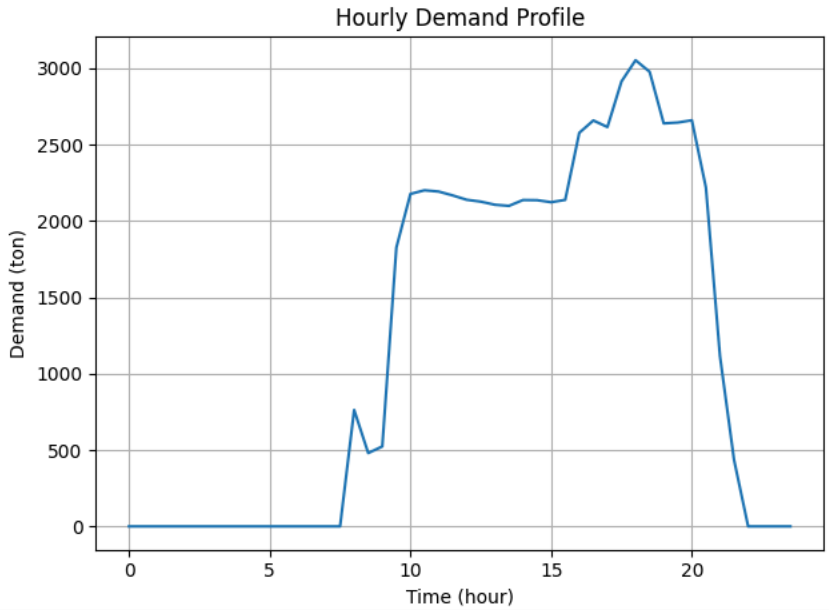
```
Status: optimal
Optimal cost (kWhr): 13436.411594406385
Chiller use (v): [0. 0. 1. 1. 0. 0. 1.]
```





Simulation with demand of day 16 data (Example)

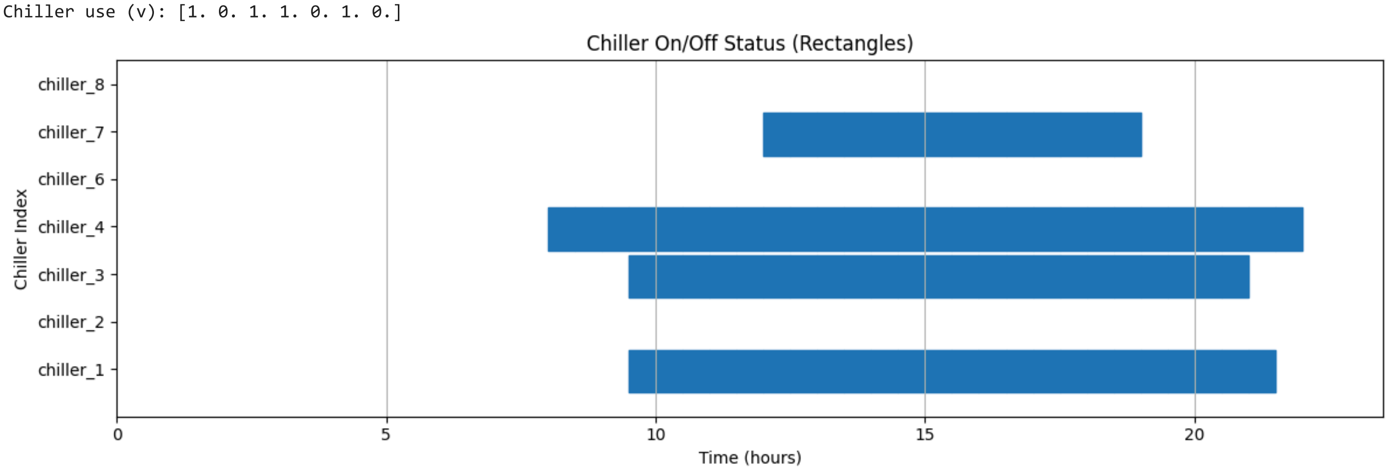
Show code

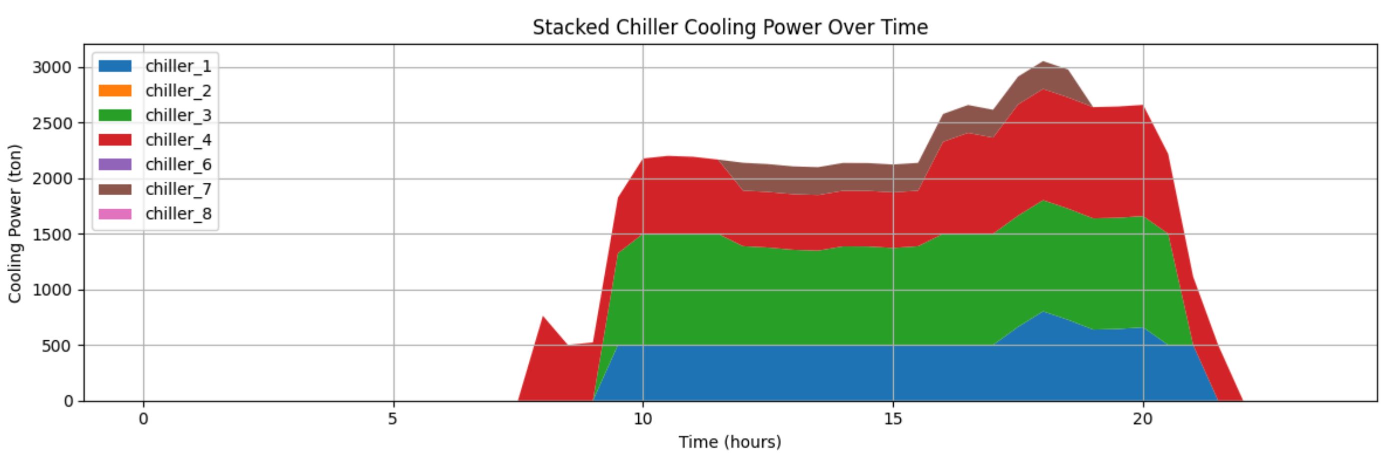


Show code

Show code

```
Status: optimal
Optimal cost (kWhr): 16692.50360436922
Chiller use (v): [1. 0. 1. 1. 0. 1. 0.]
```





Show code

Show code

```
from IPython.display import clear_output
#Short Command for calling Problem Solver
# D_sam_down is array have 24*sample_per_hour element
MILP_chiller_problem_n(D1_sam_down,sam_per_hour,TOC)
clear_output()


D_dict = dict()

for d in range(28):
  D = plant_ton[d*24*60:(d+1)*24*60]  # Ensure it's 1440 elements long
  D[22*60:24*60] = 0
  D[0*60:1*60] = 0
  D = np.array(D)        # In case it's not a NumPy array yet
  # Reshape to 24 rows of 60 columns each and take mean along axis 1
  D_sam_down = D.reshape(24*sam_per_hour, 60//sam_per_hour).mean(axis=1)
  D_sam_down = np.where(D_sam_down < 50, 0, D_sam_down)
  D =D_sam_down
  D_dict[d] = D
```

Show code

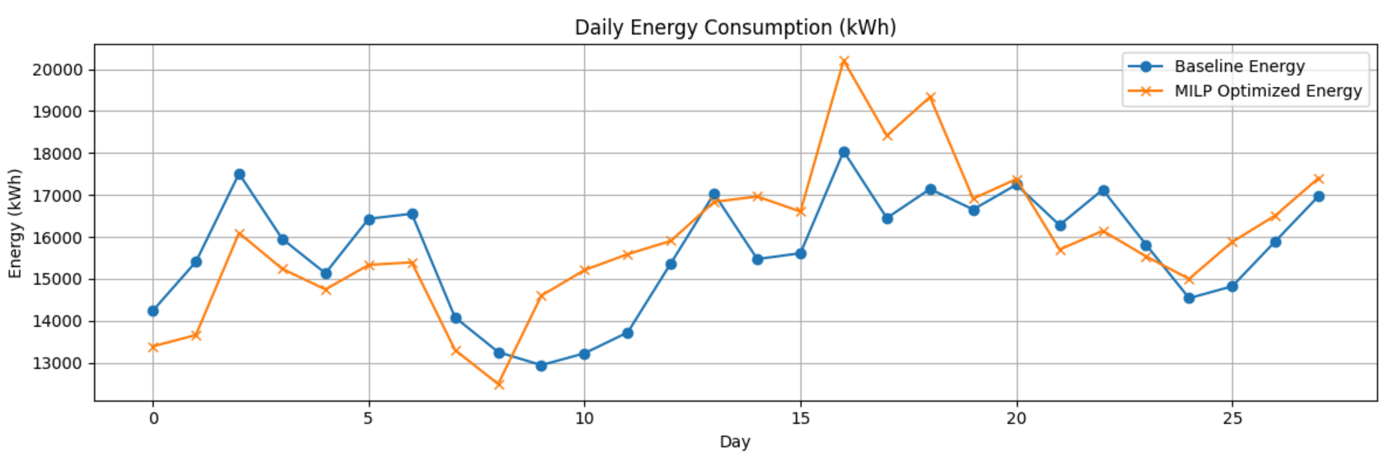
Show code

Potential energy savings (absolute or percentage)

From the energy savings graph, it can be seen that there is a portion of energy that has been successfully conserved. This is because the MILP formulation simplifies the problem in a way that tends to favor the selection of the most efficient chiller. However, there are some days when the energy consumption exceeds the baseline. This may be due to the model being overly simplified, overlooking certain advantages that arise from nonlinear characteristics. Additionally, the inclusion of the energy reserve constraint ($P_{max}^{\top} S_t < 1.1 D_t$) may sometimes result in some chillers being activated more than actually necessary.

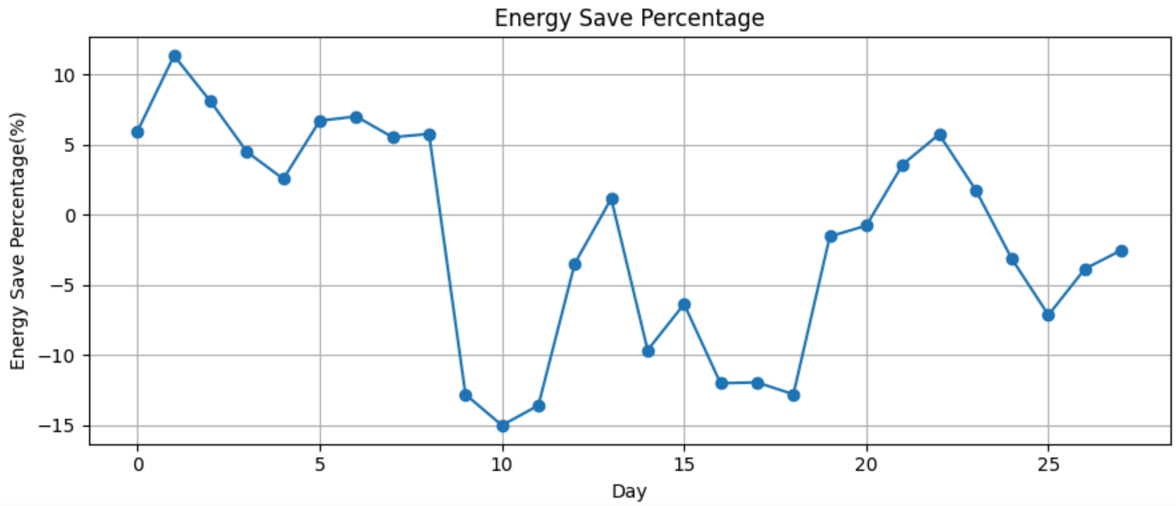
Baseline Energy vs MILP Optimized Energy

Show code



Energy Save Percentage in each day case of Demand

Show code



Any observed trade-offs or limitations

From monthly simulation, we would see the clear problem that control have tread to chosen the same chiller in eveyday continusly. this tread could casue a problem to damage to the chiller both eveyday active and non-active chiller. So for simply impliaction we could bias TOC value for balance activity in each chiller in monthly simulation.

Due to running the code on Colab, the solver we use is free and may result in slow execution, even though we use a simplified model. However, using a more suitable optimization environment such as MATLAB or other dedicated solvers could provide faster computation. Moreover, by formulating the problem as a Mixed-Integer Linear Programming (MILP) model, we can guarantee that the solution is truly globally optimal, unlike heuristic optimization methods.

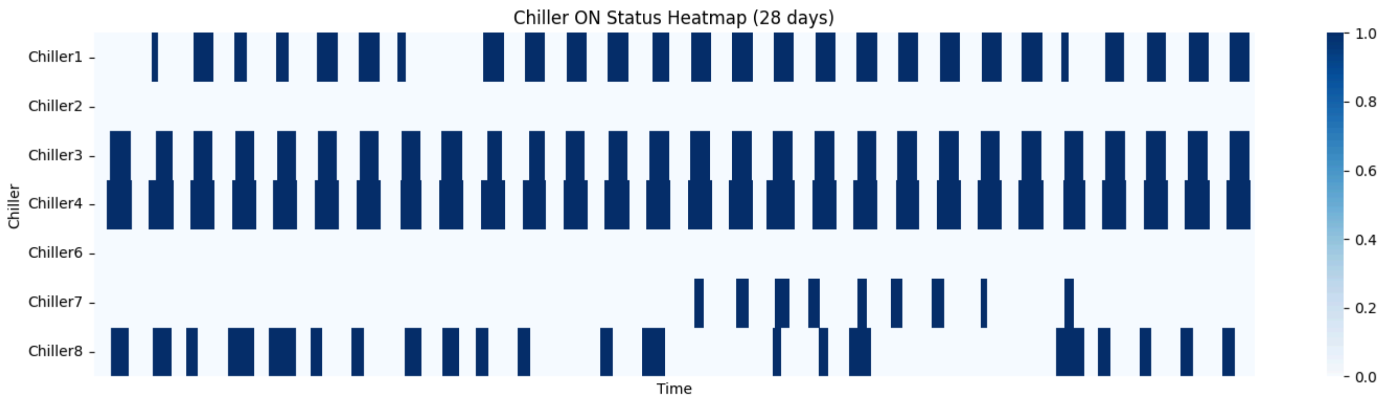
Monthly Simulation

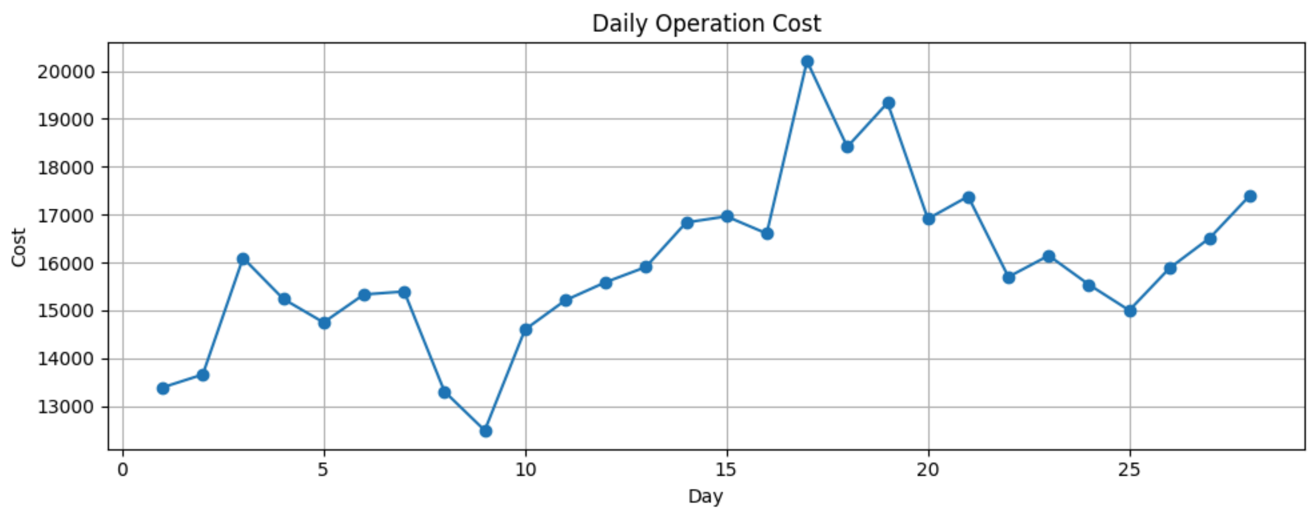
Show code

	Total On-Days	Total Energy (kWh)
Chiller1	26	127482.303209
Chiller2	0	0.000000
Chiller3	28	309441.291098
Chiller4	28	297808.631629
Chiller6	0	0.000000
Chiller7	9	11479.920123
Chiller8	21	42651.159697

Next steps: Generate code with `df_summary` View recommended plots New interactive sheet

Show code





Monthly Simulation with Bias TOC

Show code

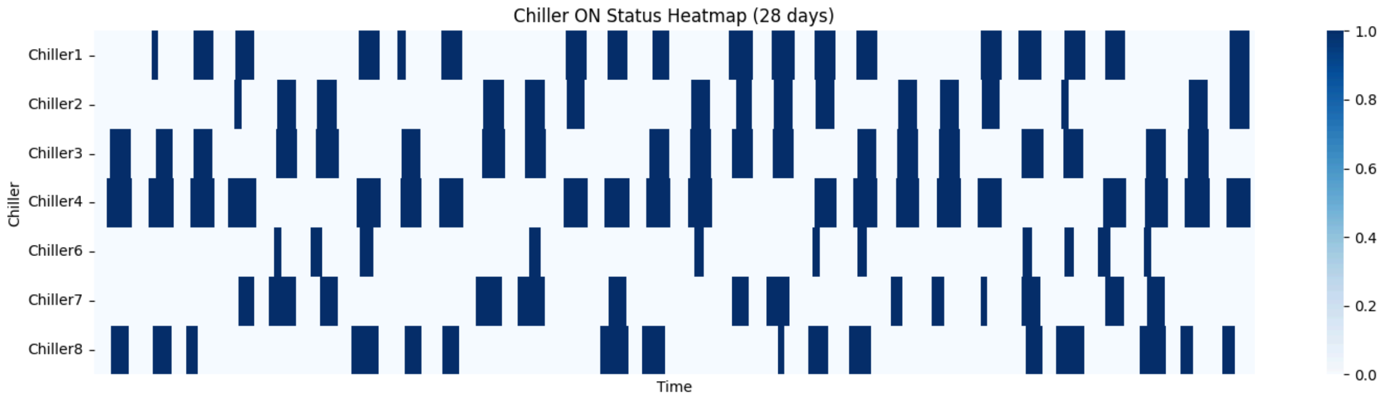
Show hidden output

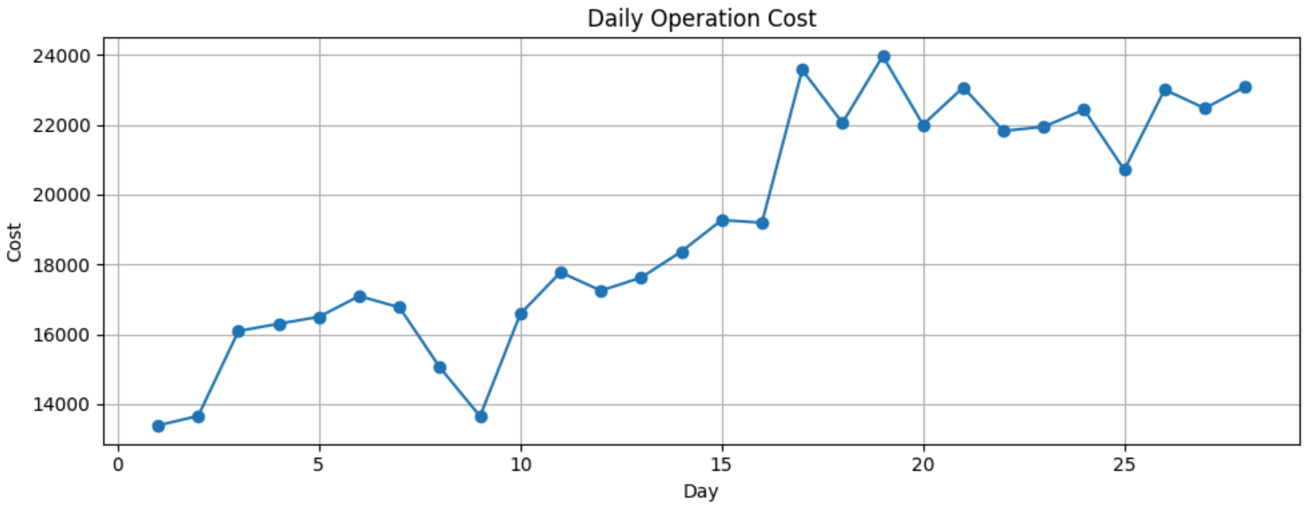
Show code

	Total On-Days	Total Energy (kWh)
Chiller1	18	125042.373059
Chiller2	16	124662.567294
Chiller3	19	221122.816134
Chiller4	20	232942.535347
Chiller6	11	13931.726353
Chiller7	14	30933.591132
Chiller8	16	40494.351802

Next steps: Generate code with `df_summary` View recommended plots New interactive sheet

Show code





Strategy could extend to real-time MPC deployment.

Solving a global solution for an MILP problem can require a long computation time. If the optimization is performed hours in advance—such as during the preparation phase the day before—it may not be an issue. However, for real-time control, this computational delay can have a significant impact. One approach to address this challenge is to adjust the sampling time: use a high sampling rate (i.e., higher accuracy) for the near-term horizon, and a lower sampling rate (i.e., lower accuracy) for time steps further in the future. After each optimization cycle and upon receiving updated data, the model should be updated with the most recent information (feedback), allowing it to adjust and improve its accuracy over time.

[1] X. Wei, G. Xu, and A. Kusiak, "Modeling and optimization of a chiller plant," Energy, vol. 73, pp. 898–907, Aug. 2014, doi: 10.1016/j.energy.2014.06.102.

[2] Y.-C. Chang, J.-K. Lin, and M.-H. Chuang, "Optimal chiller loading by genetic algorithm for reducing energy consumption," Energy and Buildings, vol. 37, no. 2, pp. 147–155, Feb. 2005, doi: 10.1016/j.enbuild.2004.06.002.

[3] K. Shejul and R. Harikrishnan, "Energy consumption optimization of chiller plants with the Genetic Algorithm based GWO and JAYA Algorithm in the dynamic pricing Demand Response," Results in Engineering, vol. 22, p. 102193, Jun. 2024, doi: 10.1016/j.rineng.2024.102193.

[4] W.-S. Lee, W.-H. Lin, C.-C. Cheng, and C.-Y. Lin, "Optimal Chiller Loading by Team Particle Swarm Algorithm for Reducing Energy Consumption," Energies, vol. 14, no. 21, p. 7066, Oct. 2021, doi: 10.3390/en14217066.

[5] J. H. Cha, J. Park, S. H. Yeon, Y. Yoon, and K. H. Lee, "Particle Swarm Optimization for multi-chiller system: Capacity configuration and load distribution," Journal of Building Engineering, vol. 98, p. 110953, Dec. 2024, doi: 10.1016/j.jobe.2024.110953.