MATH6184 coursework

ID: 35621168

Contents

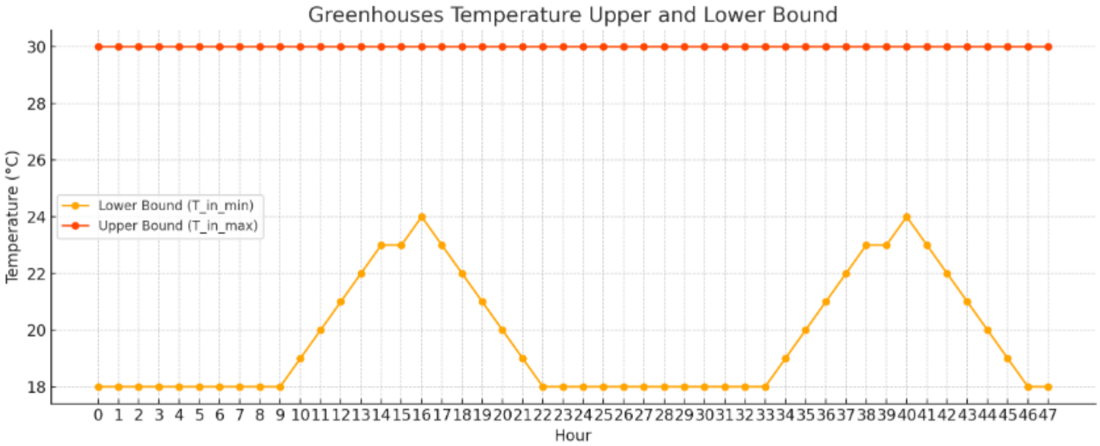
1 Introduction

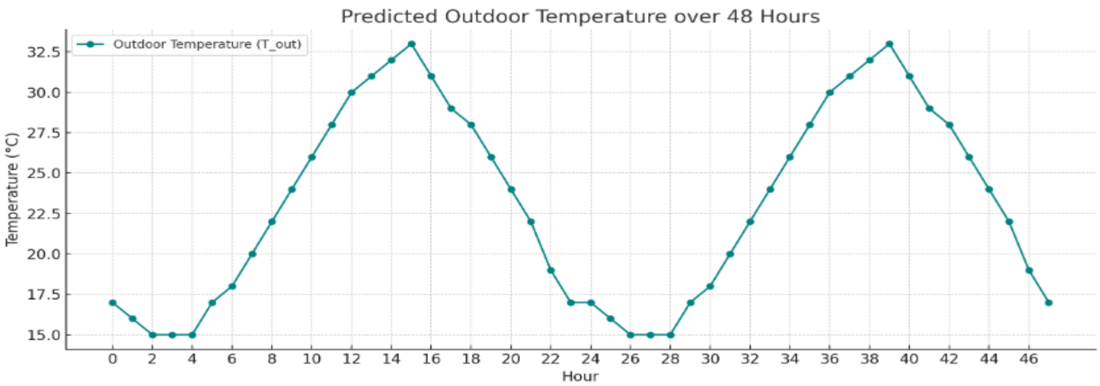
1.1 Background and Purpose

This project aims to analyze the cost of ventilation control in smart greenhouse systems through nonlinear optimization models, explore the minimum cost of greenhouse operation, the ideal greenhouse temperature cost, and the balance decision under comprehensive consideration of cost and temperature factors.

The purpose of this project is to ensure that the temperature in the greenhouse meets the requirements of crop growth, optimize the ventilation rate to minimize the total electricity cost, and balance the relationship between temperature and energy consumption.

1.2 Problem Description





At hour i, the cost to replace q% of air in the greenhouses with air from outside is

$$\mathbf{Cost}(q, i) = \frac{a_i q^3}{1000} + b_i q \text{ £}$$

The upper and lower limits of the greenhouse temperature, outdoor temperature, and ventilation cost function for 48 hours are as above. It is necessary to calculate the minimum cost of greenhouse operation, the ideal greenhouse temperature cost, and the balance of cost and temperature factors.

2.Assumptions and formulations of the models

Assumptions:

1. The greenhouse is a closed single-zone model and there will be no heat loss.
2. The hourly ventilation rate q(t) determines the mixing ratio of indoor and outdoor temperatures during that hour and is fixed and will not change during that hour. (q represents the ratio of outdoor air to the total greenhouse space).
3. The indoor temperature is adjusted by the indoor temperature of the previous hour and the outside temperature of this hour. The temperature formula can be expressed as:

$$\mathrm{T_{in}[t]} = \left(1 - \frac{q[t]}{100}\right) \cdot T_{\mathrm{in}}[t-1] + \frac{q[t]}{100} \cdot T_{\mathrm{out}}[t]$$

4. Independent decision every hour, time step is 1 hour, optimization period is 48 hours. The model makes ventilation rate decisions once every hour, optimizing a total of 48 time points.
5. Energy cost as a nonlinear function of ventilation rate. where a[t], b[t] are the nonlinear and linear cost coefficients given for each hour.

$$\mathrm{Cost}(q, i) = \frac{a_t q^3}{1000} + b_t q \text{ £}$$

Formulations of the Models:

1. T as a set, indicate the steps of hours in 2 days.

$$T = \{0,1,2,\ldots,47\}$$

2. Parameters table

$\mathrm{T_{in\_start}}$	Initial indoor temperature at hour 0.
$\mathrm{T_{out}}[\mathrm{t}]$	Outdoor temperature at hour t
T_{inmin}, T_{inmax}	Minimum and maximum allowed indoor temperature at hour t
$a[t], b[t]$	Nonlinear and linear electricity cost coefficients at hour t
T_{target}	Target indoor temperature (e.g., 27°C)
Penalty	Penalty weight for deviation from the target temperature

3. Decision Variables

$q[t] \in [0,100]$	Ventilation rate at t t hour (in %)
$T_{in}[t]$	Indoor temperature at hour t(t≠0)
deviation[t]≥0	Absolute deviation of room temperature from target temperature at hour t

3. Tasks and Results

3.1 Task 1： Cost minimization model

Object: Minimize the electricity cost

$$\min \text{Total\_Cost} = \sum_{t\in T}\left(\frac{a[t]\cdot q[t]^3}{1000} + b[t]\cdot q[t]\right)$$

Constrain: Indoor Temperature(in hour t)

$$T_{\text{in}}[t] = \left(1-\frac{q[t]}{100}\right)\cdot T_{\text{in\_start}} + \frac{q[t]}{100}\cdot T_{\text{out}}[t], \quad \text{if } t=1$$

$$T_{\text{in}}[t] = \left(1-\frac{q[t]}{100}\right)\cdot T_{\text{in}}[t-1] + \frac{q[t]}{100}\cdot T_{\text{out}}[t], \quad \text{if } t>1$$

Indoor Temperature Bounds

$$T_{\text{in\_min}}[t] \le T_{\text{in}}[t] \le T_{\text{in\_max}}[t], \quad \forall t\in T,\ t\ne t_0$$

Result:

1. The Minimal Total electricity cost： 90.6115£
2. The air exchange rate q[t] is 0 for most of the time, and ventilation occurs only from the 11th to the 15th hour ($q \approx 10$)
3. The indoor temperature T_in[t] starts to rise gradually at the 11th hour and then stabilizes at 24°C

This optimization result indicates that in order to save costs, the ventilation rate will be reduced as much as possible, and ventilation will only be performed when the room temperature approaches the lower limit of the temperature.

3.2 Task 2： Minimize temperature deviation

Object: Minimize the total deviation

$$\min \text{Total\_Deviation} = \sum_{t\in T,\ t\ne t_0} \text{deviation}[t]$$

Constrain: Indoor Temperature

$$T_{\text{in}}[t] = \left(1-\frac{q[t]}{100}\right)\cdot T_{\text{in\_start}} + \frac{q[t]}{100}\cdot T_{\text{out}}[t], \quad \text{if } t=1$$

$$T_{\text{in}}[t] = \left(1-\frac{q[t]}{100}\right)\cdot T_{\text{in}}[t-1] + \frac{q[t]}{100}\cdot T_{\text{out}}[t], \quad \text{if } t>1$$

Indoor Temperature Bounds

$$T_{\text{in\_min}}[t] \le T_{\text{in}}[t] \le T_{\text{in\_max}}[t], \quad \forall t\in T,\ t\ne t_0$$

Deviation

$$\text{deviation}[t] \ge T_{\text{in}}[t] - T_{\text{target}}, \qquad \forall t\in T,\ t\ne t_0$$

$$\text{deviation}[t] \ge T_{\text{target}} - T_{\text{in}}[t], \qquad \forall t\in T,\ t\ne t_0$$

Result:

1. Total temperature deviation (Total_Deviation): 64
2. Total electricity cost: 1607.5

3. The ventilation rate q[t] reaches 100% from the 7th to the 11th hour, quickly raising the room temperature to 27°C
4. After the 12th hour, the ventilation rate approaches 0, and the room temperature is stabilized at the target value

This method starts 100% ventilation once the outside temperature is higher than the greenhouse temperature, and the electricity cost is very high.

3.3 Task 3: Tradeoff optimization between cost and temperature deviation

Object

$$\min \text{Total\_Objective} = \sum_{t \in T} \left(\frac{a[t] \cdot q[t]^3}{1000} + b[t] \cdot q[t] \right) + \sum_{t \in T,\, t \neq t_0} \text{penalty} \cdot \text{deviation}[t]$$

Constrain

$$T_{\text{in}}[t] = \left(1 - \frac{q[t]}{100}\right) \cdot T_{\text{in\_start}} + \frac{q[t]}{100} \cdot T_{\text{out}}[t], \quad \text{if } t = 1$$

$$T_{\text{in}}[t] = \left(1 - \frac{q[t]}{100}\right) \cdot T_{\text{in}}[t-1] + \frac{q[t]}{100} \cdot T_{\text{out}}[t], \quad \text{if } t > 1$$

Indoor Temperature Bounds

$$T_{\text{in\_min}}[t] \leq T_{\text{in}}[t] \leq T_{\text{in\_max}}[t], \quad \forall t \in T,\ t \neq t_0$$

Deviation

$$\text{deviation}[t] \geq T_{\text{in}}[t] - T_{\text{target}}, \quad \forall t \in T,\ t \neq t_0$$

$$\text{deviation}[t] \geq T_{\text{target}} - T_{\text{in}}[t], \quad \forall t \in T,\ t \neq t_0$$

Result:

Penalty	**Total Cost**	**Average temperature deviation**	**Total Objective after penalty**
0.10	90.61	4.756	110.7115226
0.50	90.61	4.756	191.1115226
0.70	125.90	4.159	224.8169308
1.00	157.70	4.021	258.8330121
1.50	157.94	4.018	309.3511797
2.00	160.04	3.986	359.5771221
10.00	254.70	3.633	1081.296475

As the penalty increases, the total cost after adding the penalty increases linearly. Therefore, for temperature-sensitive crops, the difference between benefits and costs should be calculated in advance; for temperature-insensitive crops, the ventilation volume can be appropriately reduced.

3.4Task 4: Solver Performance Comparison

λ=1

Solver	Object	Literations	Times
baron	258.83	13	4.422
conopt	258.83	11	0.05
copt	0	Fail	
gurobi	0	Fail	
ilogcp	0	Fail	
knitro	258.83	11	0.17
lgo	0	Fail	
loqo	0	Fail	
minos	0	Fail	
octeract	258.83	25	6.02
snopt	258.83	213	0.31
xpress	0	Fail	

Found that: "conopt" and "knitro" performed best, taking less than 0.25 seconds and less than 15 iterations

Knotro, octeract, snopt can work out this model but octeract take 6 seconds, snopt take 213 literations.

3.5 Task 5:

In this greenhouse ventilation optimization problem, the best performing solvers are CONOPT and KNITRO. Their features and advantages are summarized as follows:

In this greenhouse ventilation optimization problem, the best performing solvers are BARON, CONOPT and KNITRO. Their features and advantages are summarized as follows:

This feature makes the nonlinear optimization model in this project perform well.

- CONOPT:

CONOPT is a solver specifically for large-scale continuous nonlinear optimization problems, using the sequential quadratic programming (SQP) method. For models involving continuous variables and smooth nonlinear constraints, CONOPT can quickly and efficiently obtain high-quality local optimal solutions. (Neos-server.org, 2025)

In this project, CONOPT can give full play to its advantages.

- KNITRO:

KNITRO is an advanced solver for large-scale nonlinear optimization, supporting multiple algorithm strategies such as interior point method and active set method. Its flexibility and ability to handle complex constraints (Neos-server.org, 2025)

In this project, KNITRO was able to converge quickly with fewer iterations and obtain high-quality solutions, taking into account both solution speed and accuracy.

3.6 Task 6： Further advice

1.In the temperature model, in addition to volume mixing, it is also necessary to consider the temperature rise under sunlight.

2.Consider the impact of temperature difference caused by external temperature changes on greenhouse heat loss. For example, in addition to the air mixing model, a temperature conduction model is added at the same time. After comprehensively calculating the impact of the two, a new optimization model is constructed.

3.Consider the impact of sudden factors such as power outages.

4Conclusion

This assignment systematically analyzes the impact of greenhouse ventilation strategies on energy consumption and temperature control effects by constructing a nonlinear optimization model.

The results show that only pursuing the lowest energy consumption will lead to a reduction in ventilation operations, which may cause the indoor temperature to deviate from the ideal growth range of crops; after introducing the temperature deviation penalty term, a reasonable balance can be achieved between energy consumption and temperature control effects.

In the solver comparison, CONOPT and KNITRO show excellent solution performance for nonlinear optimization problems. This study not only reflects the application of optimization algorithms in intelligent agricultural control, but also provides an important reference.

Appendix

Model1：

```
#question1.mod


# Sets & Parameters
set T ordered;                     # Time steps: 0 to 47

param t0;                          # Initial time step (e.g., 0)
param T_in_start;                  # Initial indoor temperature
param T_out {T};                   # Outdoor temperature
param T_in_min {T};                   # Minimum allowed indoor temperature
param T_in_max {T};                   # Maximum allowed indoor temperature
param a {T};                       # Non-linear electricity cost coefficient
param b {T};                       # Linear electricity cost coefficient

# Decision Variables
var q {T} >= 0, <= 100;            # Air exchange ratio (%)
var T_in {t in T: t != t0};        # Indoor temperature at time t (t ≠ t0)

# Objective: Minimize electricity cost
minimize Total_Cost:
    sum {t in T} (a[t] * (q[t])^3 / 1000 + b[t] * q[t]);

# Constraints
# Recursive indoor temperature update (using T_in_start at t0)
subject to Temperature_Update {t in T: t != t0}:
    T_in[t] = (1 - q[t]/100) * (if prev(t) = t0 then T_in_start else T_in[prev(t)])
              + (q[t]/100) * T_out[t];

# Temperature bounds for indoor temperature
subject to Temperature_Limits {t in T: t != t0}:
    T_in_min[t] <= T_in[t] <= T_in_max[t];
```

Data1：

```
# question1.dat
# (Hours from day1 to day2,from 0 to 47)
set T := 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
          24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47;
#Initial temperature
param t0 := 0;
param T_in_start := 19;

#Temperature outside
param T_out :=
0 17 1 16 2 15 3 15 4 15 5 17 6 18 7 20 8 22 9 24
10 26 11 28 12 30 13 31 14 32 15 33 16 31 17 29 18 28 19 26
20 24 21 22 22 19 23 17 24 17 25 16 26 15 27 15 28 15 29 17
30 18 31 20 32 22 33 24 34 26 35 28 36 30 37 31 38 32 39 33
40 31 41 29 42 28 43 26 44 24 45 22 46 19 47 17;
#Minimum room temperature
param T_in_min :=
0 18 1 18 2 18 3 18 4 18 5 18 6 18 7 18 8 18 9 18
10 19 11 20 12 21 13 22 14 23 15 23 16 24 17 23 18 22 19 21
20 20 21 19 22 18 23 18 24 18 25 18 26 18 27 18 28 18 29 18
30 18 31 18 32 18 33 18 34 19 35 20 36 21 37 22 38 23 39 23
40 24 41 23 42 22 43 21 44 20 45 19 46 18 47 18;
#Maximum room temperature
param T_in_max :=
0 30 1 30 2 30 3 30 4 30 5 30 6 30 7 30 8 30 9 30
10 30 11 30 12 30 13 30 14 30 15 30 16 30 17 30 18 30 19 30
20 30 21 30 22 30 23 30 24 30 25 30 26 30 27 30 28 30 29 30
30 30 31 30 32 30 33 30 34 30 35 30 36 30 37 30 38 30 39 30
40 30 41 30 42 30 43 30 44 30 45 30 46 30 47 30;
#Non-linear electricity cost coefficient
param a :=
0 0.1 1 0.1 2 0.1 3 0.1 4 0.1 5 0.1 6 0.2 7 0.2 8 0.2 9 0.2
10 0.2 11 0.3 12 0.3 13 0.3 14 0.3 15 0.3 16 0.3 17 0.3 18 0.3 19 0.2
20 0.2 21 0.1 22 0.1 23 0.1 24 0.1 25 0.1 26 0.1 27 0.1 28 0.1 29 0.1
30 0.2 31 0.2 32 0.2 33 0.2 34 0.2 35 0.3 36 0.3 37 0.3 38 0.3 39 0.3
40 0.3 41 0.3 42 0.3 43 0.2 44 0.2 45 0.1 46 0.1 47 0.1;
#Linear electricity cost coefficient
param b :=
0 1 1 1 2 1 3 1 4 1 5 1.5 6 1.6 7 1.6 8 1.6 9 1.8
10 1.8 11 1.8 12 1.8 13 1.8 14 1.7 15 1.6 16 1.6 17 1.6 18 1.4 19 1.2
20 1 21 1 22 1 23 1 24 1 25 1 26 1 27 1 28 1 29 1.5
30 1.6 31 1.6 32 1.6 33 1.8 34 1.8 35 1.8 36 1.8 37 1.8 38 1.7 39 1.6
40 1.6 41 1.6 42 1.4 43 1.2 44 1 45 1 46 1 47 1;
```

```
ampl: model question1.mod;
ampl: data question1.dat;
ampl: option solver conopt;
ampl: solve;
CONOPT 4.36: Locally optimal; objective 90.61152263
7 iterations; evals: nf = 11, ng = 11, nc = 193, nJ = 26, nH =
ampl: display Total_Cost;
Total_Cost = 90.6115
```

Model2:

```
#question2.mod

# Sets & Parameters
set T ordered;                        # Time steps: 0 to 47

param t0;                             # Initial time step (e.g., 0)
param T_in_start;                     # Initial indoor temperature
param T_out {T};                      # Outdoor temperature
param T_in_min {T};                  # Minimum allowed indoor temperature
param T_in_max {T};                  # Maximum allowed indoor temperature
param a {T};                          # Non-linear electricity cost coefficient
param b {T};                          # Linear electricity cost coefficient
param T_target;                       # Target temperature

# Decision Variables
var q {T} >= 0, <= 100;               # Air exchange ratio (%)
var T_in {t in T: t != t0};           # Indoor temperature at time t (t ≠ t0)

# Auxiliary Variables
var deviation {t in T: t != t0} >= 0; # Absolute deviation from T_target
#Total Cost
var Total_Cost >= 0;

# Objective: Minimize total deviation from target temperature
minimize Total_Deviation:
    sum {t in T: t != t0} deviation[t];

subject to Total_Cost_:
    Total_Cost = sum {t in T} (a[t] * (q[t])^3 / 1000 + b[t] * q[t]);
# Constraints

# Recursive indoor temperature update (using T_in_start at t0)
subject to Temperature_Update {t in T: t != t0}:
    T_in[t] = (1 - q[t]/100) *
              (if prev(t) = t0 then T_in_start else T_in[prev(t)])
              + (q[t]/100) * T_out[t];

# Temperature bounds
subject to Temperature_Limits {t in T: t != t0}:
    T_in_min[t] <= T_in[t] <= T_in_max[t];

# Deviation constraints (absolute value handling)
subject to Dev_Positive {t in T: t != t0}:
    deviation[t] >= T_in[t] - T_target;

subject to Dev_Negative {t in T: t != t0}:
    deviation[t] >= T_target - T_in[t];
```

Data2：

```
#question2.dat
# (Hours from day1 to day2, from 0 to 47)
set T := 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
          24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47;

# Initial temperature
param t0 := 0;
param T_in_start := 19;

# Temperature outside
param T_out :=
0 17 1 16 2 15 3 15 4 15 5 17 6 18 7 20 8 22 9 24
10 26 11 28 12 30 13 31 14 32 15 33 16 31 17 29 18 28 19 26
20 24 21 22 22 19 23 17 24 17 25 16 26 15 27 15 28 15 29 17
30 18 31 20 32 22 33 24 34 26 35 28 36 30 37 31 38 32 39 33
40 31 41 29 42 28 43 26 44 24 45 22 46 19 47 17;

# Minimum room temperature
param T_in_min :=
0 18 1 18 2 18 3 18 4 18 5 18 6 18 7 18 8 18 9 18
10 19 11 20 12 21 13 22 14 23 15 23 16 24 17 23 18 22 19 21
20 20 21 19 22 18 23 18 24 18 25 18 26 18 27 18 28 18 29 18
30 18 31 18 32 18 33 18 34 19 35 20 36 21 37 22 38 23 39 23
40 24 41 23 42 22 43 21 44 20 45 19 46 18 47 18;

# Maximum room temperature
param T_in_max :=
0 30 1 30 2 30 3 30 4 30 5 30 6 30 7 30 8 30 9 30
10 30 11 30 12 30 13 30 14 30 15 30 16 30 17 30 18 30 19 30
20 30 21 30 22 30 23 30 24 30 25 30 26 30 27 30 28 30 29 30
30 30 31 30 32 30 33 30 34 30 35 30 36 30 37 30 38 30 39 30
40 30 41 30 42 30 43 30 44 30 45 30 46 30 47 30;

# Non-linear electricity cost coefficient
param a :=
0 0.1 1 0.1 2 0.1 3 0.1 4 0.1 5 0.1 6 0.2 7 0.2 8 0.2 9 0.2
10 0.2 11 0.3 12 0.3 13 0.3 14 0.3 15 0.3 16 0.3 17 0.3 18 0.3 19 0.2
20 0.2 21 0.1 22 0.1 23 0.1 24 0.1 25 0.1 26 0.1 27 0.1 28 0.1 29 0.1
30 0.2 31 0.2 32 0.2 33 0.2 34 0.2 35 0.3 36 0.3 37 0.3 38 0.3 39 0.3
40 0.3 41 0.3 42 0.3 43 0.2 44 0.2 45 0.1 46 0.1 47 0.1;

# Linear electricity cost coefficient
param b :=
0 1 1 1 2 1 3 1 4 1 5 1.5 6 1.6 7 1.6 8 1.6 9 1.8
10 1.8 11 1.8 12 1.8 13 1.8 14 1.7 15 1.6 16 1.6 17 1.6 18 1.4 19 1.2
20 1 21 1 22 1 23 1 24 1 25 1 26 1 27 1 28 1 29 1.5
30 1.6 31 1.6 32 1.6 33 1.8 34 1.8 35 1.8 36 1.8 37 1.8 38 1.7 39 1.6
40 1.6 41 1.6 42 1.4 43 1.2 44 1 45 1 46 1 47 1;

# Target temperature and penalty weight
param T_target := 27;
```

```
ampl: display Total_Deviation;
Total_Deviation = 64

ampl: display Total_Cost;
Total_Cost = 1607.5
```

Model3:

```
#question3.mod

# Sets & Parameters
set T ordered;                        # Time steps: 0 to 47
set SOLVERS;
# solver solve_time
param obj_value {SOLVERS};


param t0;                             # Initial time step (e.g., 0)
param T_in_start;                     # Initial indoor temperature
param T_out {T};                      # Outdoor temperature
param T_in_min {T};                  # Minimum allowed indoor temperature
param T_in_max {T};                  # Maximum allowed indoor temperature
param a {T};                          # Non-linear electricity cost coefficient
param b {T};                          # Linear electricity cost coefficient
param T_target;                       # Ideal internal temperature (e.g. 27°C)
param penalty;                         # Penalty parameter on deviation from T_target

# Decision Variables
var q {T} >= 0, <= 100;               # Air exchange ratio (%)
var T_in {t in T: t != t0};           # Indoor temperature at time t (t ≠ t0)

# Auxiliary Variables
var deviation {t in T: t != t0} >= 0; # Absolute deviation from T_target

# Objective: Minimize electricity cost + penalty on temperature deviation
minimize Total_Objective:
    sum {t in T} (
        a[t] * q[t]^3 / 1000 + b[t] * q[t]
    )
    +
    sum {t in T: t != t0} (
        penalty * deviation[t]
    );

# Constraints

# Recursive indoor temperature update
subject to Temperature_Update {t in T: t != t0}:
    T_in[t] = (1 - q[t]/100) *
              (if prev(t) = t0 then T_in_start else T_in[prev(t)])
              + (q[t]/100) * T_out[t];

# Temperature bounds
subject to Temperature_Limits {t in T: t != t0}:
    T_in_min[t] <= T_in[t] <= T_in_max[t];

# Absolute deviation from T_target
subject to Dev_Positive {t in T: t != t0}:
    deviation[t] >= T_in[t] - T_target;

subject to Dev_Negative {t in T: t != t0}:
    deviation[t] >= T_target - T_in[t];
```

Data3：

```
#question.dat
# (Hours from day1 to day2, from 0 to 47)
set T := 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
          24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47;

# Initial temperature
param t0 := 0;
param T_in_start := 19;
param T_target := 27;
# Temperature outside
param T_out :=
0 17 1 16 2 15 3 15 4 15 5 17 6 18 7 20 8 22 9 24
10 26 11 28 12 30 13 31 14 32 15 33 16 31 17 29 18 28 19 26
20 24 21 22 22 19 23 17 24 17 25 16 26 15 27 15 28 15 29 17
30 18 31 20 32 22 33 24 34 26 35 28 36 30 37 31 38 32 39 33
40 31 41 29 42 28 43 26 44 24 45 22 46 19 47 17;

# Minimum room temperature
param T_in_min :=
0 18 1 18 2 18 3 18 4 18 5 18 6 18 7 18 8 18 9 18
10 19 11 20 12 21 13 22 14 23 15 23 16 24 17 23 18 22 19 21
20 20 21 19 22 18 23 18 24 18 25 18 26 18 27 18 28 18 29 18
30 18 31 18 32 18 33 18 34 19 35 20 36 21 37 22 38 23 39 23
40 24 41 23 42 22 43 21 44 20 45 19 46 18 47 18;

# Maximum room temperature
param T_in_max :=
0 30 1 30 2 30 3 30 4 30 5 30 6 30 7 30 8 30 9 30
10 30 11 30 12 30 13 30 14 30 15 30 16 30 17 30 18 30 19 30
20 30 21 30 22 30 23 30 24 30 25 30 26 30 27 30 28 30 29 30
30 30 31 30 32 30 33 30 34 30 35 30 36 30 37 30 38 30 39 30
40 30 41 30 42 30 43 30 44 30 45 30 46 30 47 30;

# Non-linear electricity cost coefficient
param a :=
0 0.1 1 0.1 2 0.1 3 0.1 4 0.1 5 0.1 6 0.2 7 0.2 8 0.2 9 0.2
10 0.2 11 0.3 12 0.3 13 0.3 14 0.3 15 0.3 16 0.3 17 0.3 18 0.3 19 0.2
20 0.2 21 0.1 22 0.1 23 0.1 24 0.1 25 0.1 26 0.1 27 0.1 28 0.1 29 0.1
30 0.2 31 0.2 32 0.2 33 0.2 34 0.2 35 0.3 36 0.3 37 0.3 38 0.3 39 0.3
40 0.3 41 0.3 42 0.3 43 0.2 44 0.2 45 0.1 46 0.1 47 0.1;

# Linear electricity cost coefficient
param b :=
0 1 1 1 2 1 3 1 4 1 5 1.5 6 1.6 7 1.6 8 1.6 9 1.8
10 1.8 11 1.8 12 1.8 13 1.8 14 1.7 15 1.6 16 1.6 17 1.6 18 1.4 19 1.2
20 1 21 1 22 1 23 1 24 1 25 1 26 1 27 1 28 1 29 1.5
30 1.6 31 1.6 32 1.6 33 1.8 34 1.8 35 1.8 36 1.8 37 1.8 38 1.7 39 1.6
40 1.6 41 1.6 42 1.4 43 1.2 44 1 45 1 46 1 47 1;
```

Run3:

```
reset;
model question3.mod;
data question3.dat;
set PENALTIES := {0.1, 0.5, 0.7, 1, 1.5, 2, 10};

printf "\n%-10s %-15s %-20s %-20s\n",
       "Penalty", "Total Cost", "Avg Temp Deviation", "Max Deviation";

for {p in PENALTIES} {
    let penalty := p;
    solve;

    printf "%-10.2f %-15.2f %-20.3f %-20.3f\n",
        p,
        sum {t in T} (a[t] * q[t]^3 / 1000 + b[t] * q[t]),
        sqrt(sum {t in T: t != t0} (T_in[t] - T_target)^2 / (card(T) - 1)),
        max {t in T: t != t0} abs(T_in[t] - T_target);
};
```

```
ampl: include question3.run;

Penalty     Total Cost       Avg Temp Deviation    Max Deviation
CONOPT 4.36: Locally optimal; objective 110.7115226
6 iterations; evals: nf = 16, ng = 11, nc = 225, nJ = 25, nH = 0, nHv = 0
0.10        90.61            4.756                 8.000
CONOPT 4.36: Locally optimal; objective 191.1115226
1 iterations; evals: nf = 1, ng = 1, nc = 1, nJ = 1, nH = 0, nHv = 0
0.50        90.61            4.756                 8.000
CONOPT 4.36: Locally optimal; objective 224.8169308
5 iterations; evals: nf = 6, ng = 5, nc = 6, nJ = 5, nH = 1, nHv = 7
0.70        125.90           4.159                 8.000
CONOPT 4.36: Locally optimal; objective 258.8330121
7 iterations; evals: nf = 13, ng = 7, nc = 92, nJ = 7, nH = 1, nHv = 4
1.00        157.70           4.021                 8.000
CONOPT 4.36: Locally optimal; objective 309.3511797
5 iterations; evals: nf = 6, ng = 5, nc = 31, nJ = 5, nH = 1, nHv = 9
1.50        157.94           4.018                 8.000
CONOPT 4.36: Locally optimal; objective 359.5771221
5 iterations; evals: nf = 6, ng = 5, nc = 35, nJ = 5, nH = 1, nHv = 9
2.00        160.04           3.986                 8.000
CONOPT 4.36: Locally optimal; objective 1081.296475
20 iterations; evals: nf = 24, ng = 17, nc = 203, nJ = 18, nH = 6, nHv = 33
10.00       254.70           3.633                 8.000
ampl:
```

Run4:

```
# question.run

# use model and data 3
model question3.mod;
data question3.dat;
#penalty=1
let penalty := 1;

# define all solver
let SOLVERS := { 'baron','conopt', 'copt', 'gurobi', 'ilogcp', 'knitro', 'lgo', 'loqo', 'minos', 'o


for {s in SOLVERS} {

    printf "Now solving with %s...\n", s;

    # solver
    option solver s;

    solve;

    # Save objective function value and solution time
    let obj_value[s] := Total_Objective;
}

# print
printf "\n========= Summary Table =========\n";
display obj_value, solve_time;
```

```
Objective 258.8330119
baron       258.83      4.41        ampl:

11 iterations; evals: nf = 24, ng = 17, nc = 311, nJ = 31, nH = 1, nHv = 12
conopt      258.83      0.05        ampl:
```

```
objective 258.8330121; feasibility error 4.39e-09
11 iterations; 16 function evaluations

suffix feaserror OUT;
suffix opterror OUT;
suffix numfcevals OUT;
suffix numiters OUT;
knitro     258.83     0.17       ampl:
```

```
-------------------------------------------------------------------------------------------
 Iteration            GAP                 LLB          BUB            Pool        Time
-------------------------------------------------------------------------------------------
         0     1.438e+01 (  5.88%)    2.445e+02    2.588e+02           1          0.5s
         3     8.689e+00 (  3.47%)    2.501e+02    2.588e+02           4          1.1s
         8     4.583e+00 (  1.80%)    2.542e+02    2.588e+02           5          2.2s
        12     2.265e+00 (  0.88%)    2.566e+02    2.588e+02           5          3.1s
        17     7.061e-01 (  0.27%)    2.581e+02    2.588e+02           6          4.2s
        22     3.397e-01 (  0.13%)    2.585e+02    2.588e+02           5          5.2s
        25     2.134e-01 (  0.08%)    2.586e+02    2.588e+02           5          5.6s

----------------------------------
Solved to global optimality. Just goated.
Objective value at global solution :  2.588e+02
----------------------------------

Solution file written to: C:\Users\mac\AppData\Local\Temp\\at19676.octsol

Time spent on:
-------------------------------------------
          IPOPT Solver:       0.062s
     Primal Heuristics:       0.063s
    Solving Relaxation:       0.222s
 Constraint Propagation:      0.070s
 Feasibility Tightening:      2.300s
  Optimality Tightening:      2.531s
-------------------------------------------
For more information, visit: https://octeract.gg/docs/

Solved_To_Global_Optimality
octeract   258.83     6.02       ampl:
```

```
SNOPT 7.5-1.2 : Optimal solution found.
213 iterations, objective 258.8330121
Nonlin evals: obj = 13, grad = 12, constrs = 13, Jac = 12.
snopt      258.83     0.31       ampl:
```

Reference

Neos-server.org. (2025b). *NEOS Server: CONOPT*. [online] Available at: https://neos-server.org/neos/solvers/nco:CONOPT/AMPL.html [Accessed 27 Apr. 2025].

Neos-server.org. (2025c). *NEOS Server: Knitro*. [online] Available at: https://neos-server.org/neos/solvers/minco:Knitro/AMPL.html [Accessed 27 Apr. 2025].