

基于“XXX”的低碳建筑研究

摘要

关键词

1 问题重述

1.1 问题背景

“双碳”即碳达峰与碳中和的简称，我国力争 2030 年前实现碳达峰，2060 年前实现碳中和。“双碳”战略倡导绿色、环保、低碳的生活方式。我国加快降低碳排放步伐，大力推进绿色低碳科技创新，以提高产业和经济的全球竞争力。低碳建筑是指在建筑材料与设备制造、施工建造和建筑物使用的整个生命周期内，减少化石能源的使用，提高能效，降低二氧化碳排放量。

1.2 目标任务

问题一：计算给定建筑通过空调调节温度的年碳排放量。

问题二：建立综合评价模型，找出易于量化的指标，评估居住建筑整个生命周期的碳排放。

问题三：基于问题二，考虑建筑生命周期三个阶段的碳排放问题，对江苏省 13 个地级市的居住建筑进行评价，验证模型的有效性。

问题四：建立碳排放预测模型，基于江苏省建筑全过程碳排放的历史数据，对 2023 年江苏省建筑全过程的碳排放量进行预测。

问题五：结合前面的讨论给出江苏省建筑碳减排的政策建议。

2 问题分析

2.1 问题一

问题一要求计算通过空题调节温度产生的年碳排放量。我们需先求出空调制热和制冷的热量，借此通过 COP 和 EER 求出空调消耗的电量，最后转换成碳排放。其中 COP 和 EER 的定义分别为

$$COP = \frac{Q_{heat}}{W}, \quad EER = \frac{Q_{cold}}{W} \quad (2.1)$$

Q_{heat}/Q_{cold} 指的是单位时间内的制热/制冷量，单位为 J ，公式中 W 指的是单位为时间内空调消耗的功率，单位为 W

首先计算出建筑物各个月的能量需求量。设室内温度要维持的温度为 t_{in} ，室外温度为 t_{out} ，当月该地区平均温度为 t_{ave} ，方便起见，我们规定

$$t_{out} = t_{ave}$$

$$t_{in} = \begin{cases} 18^{\circ}C & t_{out} < 18^{\circ}C \\ t_{out} & t_{out} \in [18^{\circ}C, 26^{\circ}C] \\ 26^{\circ}C & t_{out} > 26^{\circ}C \end{cases}$$

我们使用热传导方程计算用来需要制热/制冷的热量，其形式为

$$\Phi = \frac{\lambda \cdot A \cdot |\Delta T|}{\delta} \quad (2.2)$$

其中 Φ 表示传热速率， λ 为导热系数， A 为传热面积， ΔT 是室内外温度差，即 $t_{in} - t_{out}$ ， δ 表示材料厚度。

将建筑分成墙、门窗、房顶、地面四个部分，分别计算并累加即可得到需要制热/制冷的热量，设为 Q_{make} ，由 COP 和 EER 的定义可得到需电量 Q_{elec} 和热量 Q_{make} 的转化关系

$$Q_{elec} = \begin{cases} \frac{Q_{make}}{EER} & \Delta t < 0 \\ 0 & \Delta t = 0 \\ \frac{Q_{make}}{COP} & \Delta t > 0 \end{cases} \quad (2.3)$$

最后根据需电量与碳排放的换算关系 $m = Q_{elec} \cdot 0.28$ 求出每月碳排放后累加，即得到年度碳排放量。

3 模型假设

4 符号说明

5 模型的建立与求解

6 结果检验与误差分析

7 模型评价

8 模型推广与改进

9 参考文献

10 附录

附录 A 问题一

Listing 1: testModel1.py

```
1 # -*- coding:utf-8 -*-
2 """
3 Project: MathematicsModeling-2023.5
4 Written by: Evan Wong
5 DATE: 2023/5/9
6 TIME: 17:09
7 """
8
9 import matplotlib.pyplot as plt
10
11 monthdays = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
12 temperatures = [-1, 2, 6, 12, 22, 28, 31, 32, 26, 23, 15, 2]
13
14 COP = 3.5
15 EER = 2.7
```

```

16
17 length = 4
18 width = 3
19 height = 3
20
21 roof_thickness = 0.3
22 roof_thermal = 0.2
23 roof_square = length * width
24 windowAndDoor_thickness = 0.1
25 windowAndDoor_thermal = 1.6
26 windowAndDoor_square = 5
27 wall_thickness = 0.3
28 wall_thermal = 0.3
29 wall_square = (length + width) * height * 2 - windowAndDoor_square
30 ground_thickness = 1.0
31 ground_thermal = 0.25
32 ground_square = length * width
33
34
35 def get_delta_t(tout):
36     if tout < 18:
37         return 18 - tout
38     elif tout > 26:
39         return 26 - tout
40     else:
41         return 0
42
43
44 def get_qmake(k, a, t, d):
45     return (k * a * t) / d
46
47
48 def get_day_qmake(dt):
49     if dt == 0:
50         return 0.0
51
52     day_qmake = \
53         get_qmake(wall_thermal, wall_square, dt, wall_thickness) + \
54         get_qmake(windowAndDoor_thermal, windowAndDoor_square, dt,
55                     windowAndDoor_thickness) + \
56         get_qmake(roof_thermal, roof_square, dt, roof_thickness) + \
57         get_qmake(ground_thermal, ground_square, dt, ground_thickness)

```

```

57     return day_qmake * 86400
58
59
60 def get_month_qmake(month):
61     return get_day_qmake( abs(get_delta_t(temperatures[month]))) * monthdays[month]
62
63
64 def get_qelec(qmake, dt):
65     if dt < 0:
66         return qmake / EER
67     elif dt == 0:
68         return 0
69     else:
70         return qmake / COP
71
72
73 def get_carbon_emission(qelec):
74     return (qelec / 3600000) * 0.28
75
76
77 carbonEmissions = []
78 for i in range(0, 12):
79     qelec = get_qelec(get_month_qmake(i), get_delta_t(temperatures[i]))
80     carbonEmissions.append(get_carbon_emission(qelec))
81
82 print("Sum of carbon emissions: " + str( sum(carbonEmissions)) + "kg")
83
84 plt.figure(figsize=(10, 6))
85 plt.bar( range(1, len(carbonEmissions) + 1), carbonEmissions, fc='g')
86 plt.title("Carbon Emissions")
87 plt.xlabel("Month")
88 plt.ylabel("kg")
89
90 plt.show()

```

附录 B 问题二

附录 C 问题三

附录 D 问题四

附录 E 问题五