Thermal System

COMP 4106 Final object

FAN ZHANG 100839154

Introduction

Automated planning and scheduling, in the relevant literature often denoted as simply planning, is a branch of artificial intelligence that concerns the realization of strategies or action sequences, typically for execution by intelligent agents, autonomous robots and unmanned vehicles. Unlike classical control and classification problems, the solutions are complex and must be discovered and optimized in multidimensional space. Planning is also related to decision theory.

For my object, I would like to introduce a thermal system, which is really used in China. In Canada, most people are living in house, and each house has their own central air conditioning. In northern China are as cold as in Canada. But unlike in Canada, most Chinese people are living in apartments without air condition. They even do not have own central air conditioning for each apartment. In order to meet China's national conditions, the Chinese government set up some special facilities to ensure that people warm in winter. These facilities use coal to heat water, hot water will be transported through special pipelines to every household, using the dissipate heat of hot water from the pipeline to keep the room warm. Therefore, the problem is between user needs and plan production time.

The motivation for the problem

If we treat each apartment or building as a reservoir, we will have multiple reservoirs, they all need use hot water continuously to keep temperature. For the huge demand for hot water, one generator is obviously not enough. Thus, we need use a lot generators to product hot water. To using those generators, we also have to cost money. Therefore, the important aspects in the daily operation of a thermal system are to meet the forecasted load demand at a minimum generation cost, and schedule a time sheet.

In my object, I would like to simulate the thermal system. I designed an algorithm to schedule a time sheet for 24-hour, which is in an ideal environment. Ideal environment means there is no heat loss during the transit, do not consider holidays and weekends, and no deviation. My purpose for the object are optimizing water allocation of generators for reservoirs in 24 hours to achieve the minimum cost, and give a scheduled timesheet and simulate it in Real-time status.

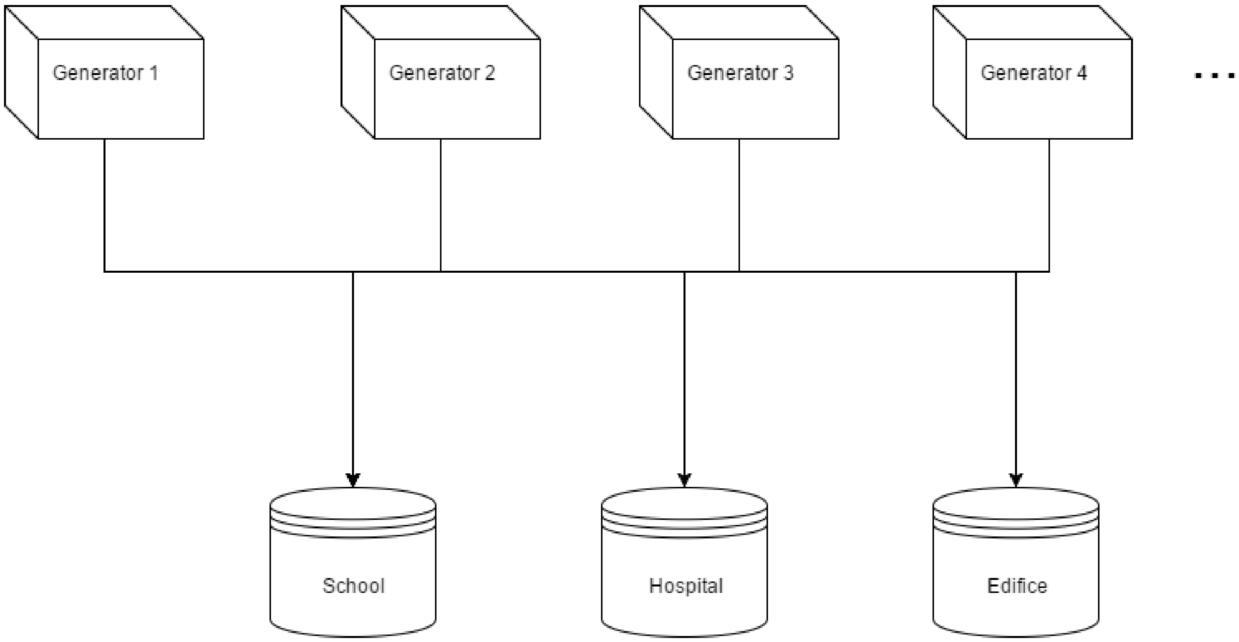
The AI techniques/algorithms used. The design choices used

Because there is no any famous algorithm to figure out this problem. I designed own algorithm for it. First, I'd like to introduce some rules I set.

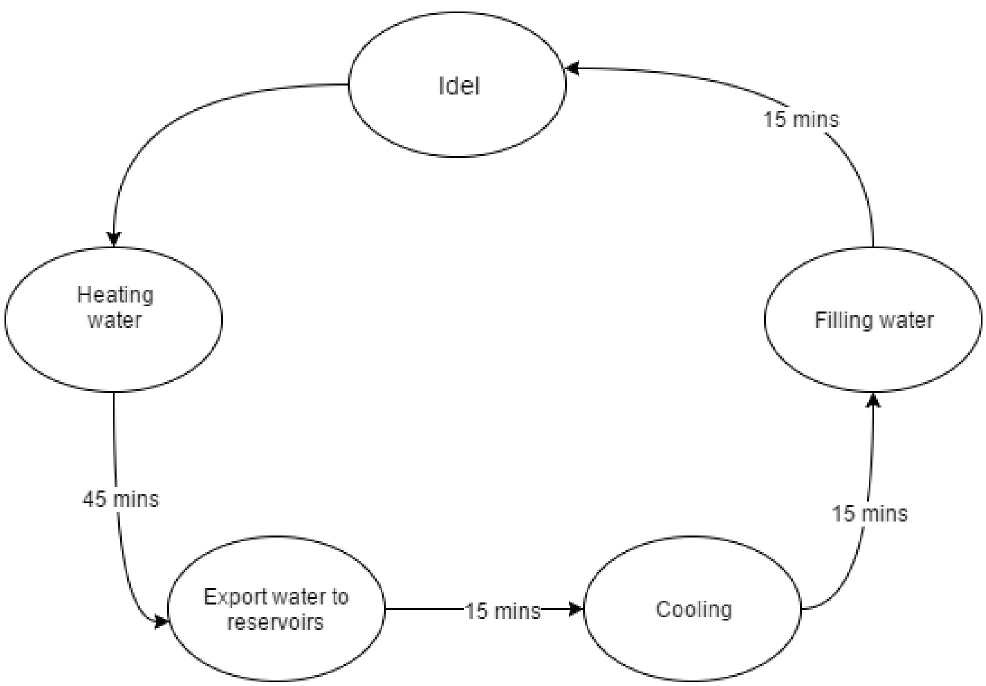
Constructive Heuristics:

1. Every generator are same volumes, this volumes is a parameter, which can be changed flexibly.
2. The number of hot water demand for every reservoirs is loaded from a text file.
3. For each reservoir, they are same volumes, and cordon of low water storage are same.
4. Ensure adequate water supply, which means the storage of water cannot lower than cordon.
5. Keep balance of each reservoir, let the volumes of storage water similar at every moment.

There is an example for my algorithm. In order to simulate the system, I assumed there are 3 reservoirs, which named by school, hospital, and edifice. And the number of hot water demand for every reservoirs are loaded from text files. Each generator has same volumes, we assume the volumes equal to 300 (this parameter can change). Because the water is continuously consumed, we need multiple generators supply water for them, and the number of generators is our goal to calculate.



A generator has its own production process, the process is a cycle. It has 5 status, idle, heating, output water to reservoirs, cooling, and filling water, after it filling water, it back to idle status. They cost different time when one state to another status. The cycle and time cost shown by following.



Due to the different needs of each reservoir, we need to inject a different amount of water in each of the different time periods to meet demand. Here I divided one hour into 4 parts, each part is 15 mins. Because we doing 24 hours algorithm, we totally have 96 parts in one day. I design the algorithm as following

```
1)  Loading water demand from text files for each reservoir
2)  Initialize N generators
3)          N = total demand for all reservoirs/volumes of one generator
4)  Calculate water consumption
5)          Remaining water = Current water − demand
6)  If Remaining water < warning of low water storage
7)          Calculate M generator need to early start.
8)                  M = total need of injection/volumes of one generator
9)          Time Jump back 3 time slots to pre schedule idle generator action, if no idle generator, open a new

10) Distribute the generator's water, which calculated by weight
11)                 Depth = 1,  injection = demand for current time slot;
12)                 Depth = 3, total demand = D1 + D2 + D3;
13)                         //V = volumes of generator, which is 300 in the example.
14)                         Injection  L1 = (D1/ total demand)*(V*M);
15)                         Injection  L2 = (D2/ total demand)*(V*M);
16)                         Injection  L3 = (D3/ total demand)*(V*M);
17) Remaining water = Current water in reservoir − demand + injection;
```

Line 2) : N is the number of generators at full load condition, N is the minimum number of generators need use today.

Line 6) : If remaining water lower than cordon, we have to injection.

Line 7) : Because each generator need 45 mins to heat the water, we need early start the generator.

Line 8) : Generators need to export all water into reservoir, and we don't want one reservoir has a lot of water, but another reservoir close to the cordon. So we need to calculate the balance for how to distribute water are reasonable.

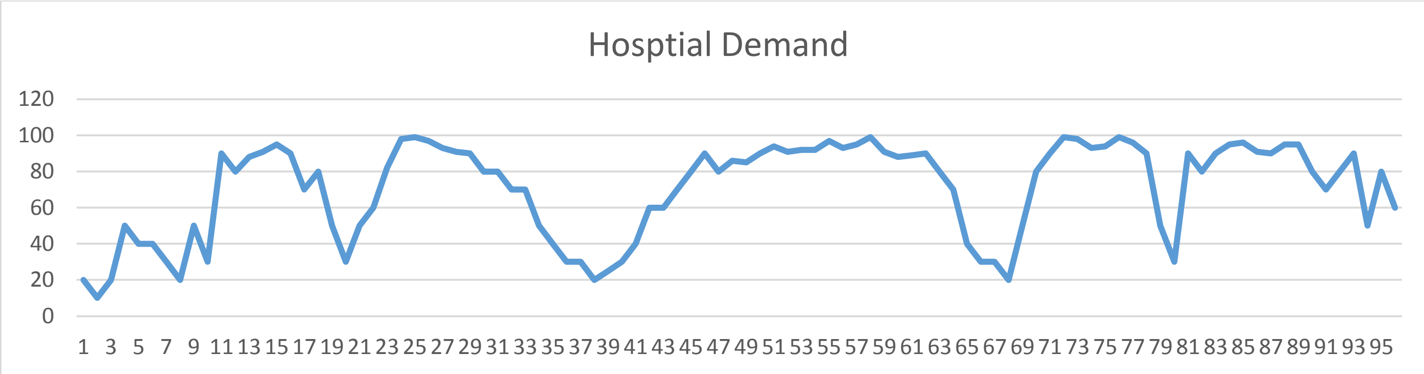
Line 11) 12) : In order to distribute the water balance, we need calculate the number of volumes to injection in each reservoir. D1 means demand for current time slot, D2 means demand for next time slot, and so on.

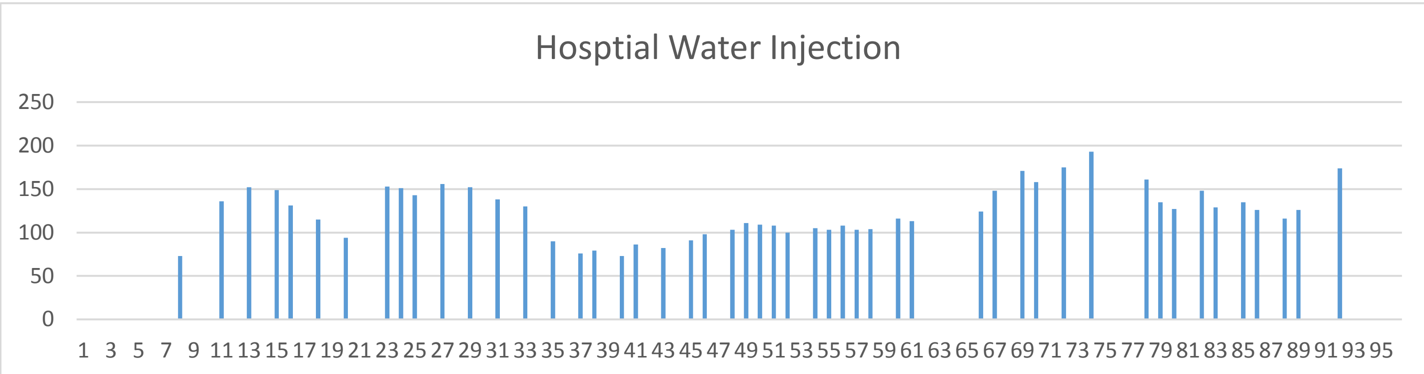
Line 14) 15) 16) : We calculate 3 depth water injection to avoid wasting. If we only look at 1 depth, we may be allocated in accordance with the existing ratio of water, but ignore the next peak of demand. Thus, we observe 3 depth of water demand in advance, in order to distribute water more reasonable.

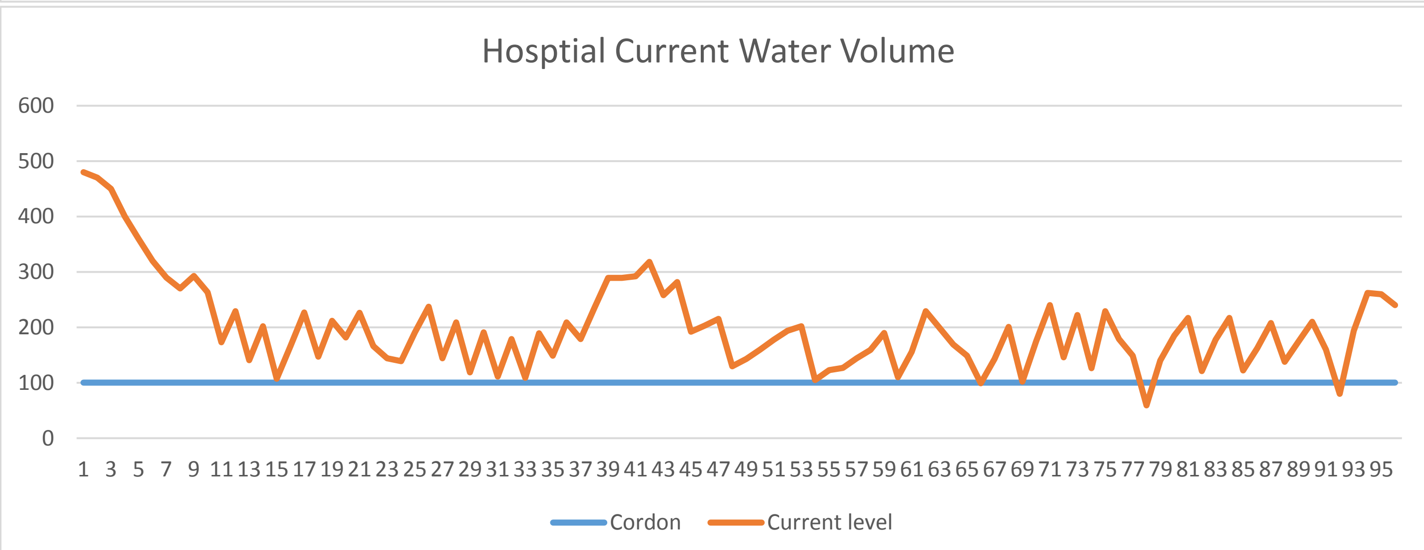
The results of applying the above-mentioned techniques to the problem

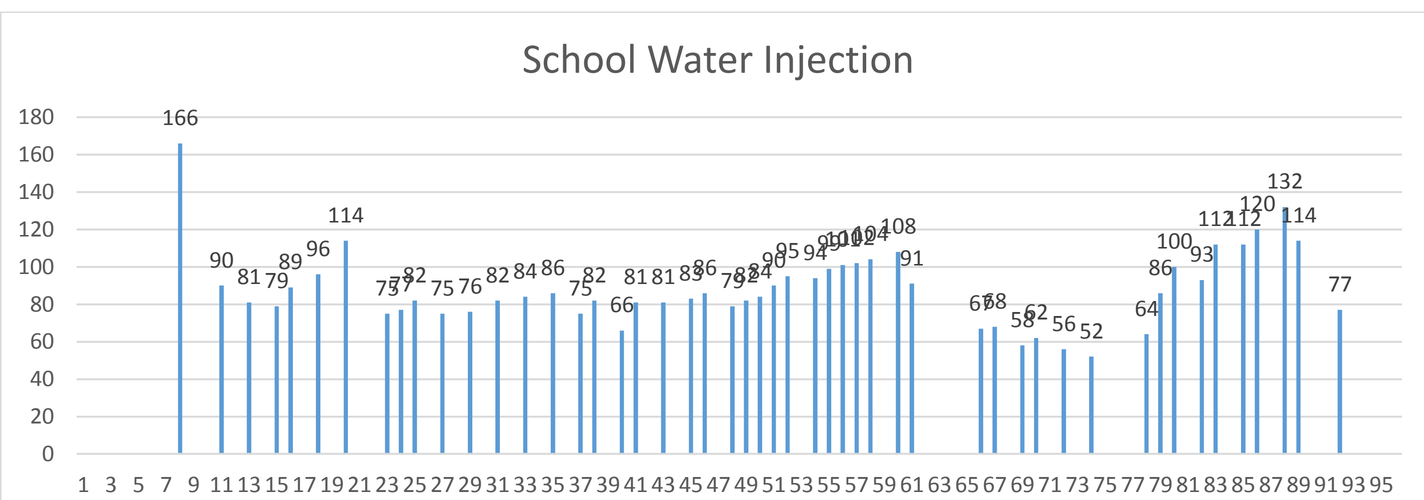
The result of my object is the schedule timesheet for multiple generators in 24 hours. In the timesheet, it give the exact time for when the generator should start. And also show how many generators should start at that time slot. Show the status for each time slot for all the generators. We also can calculate the cost, if we know how much for one generator.

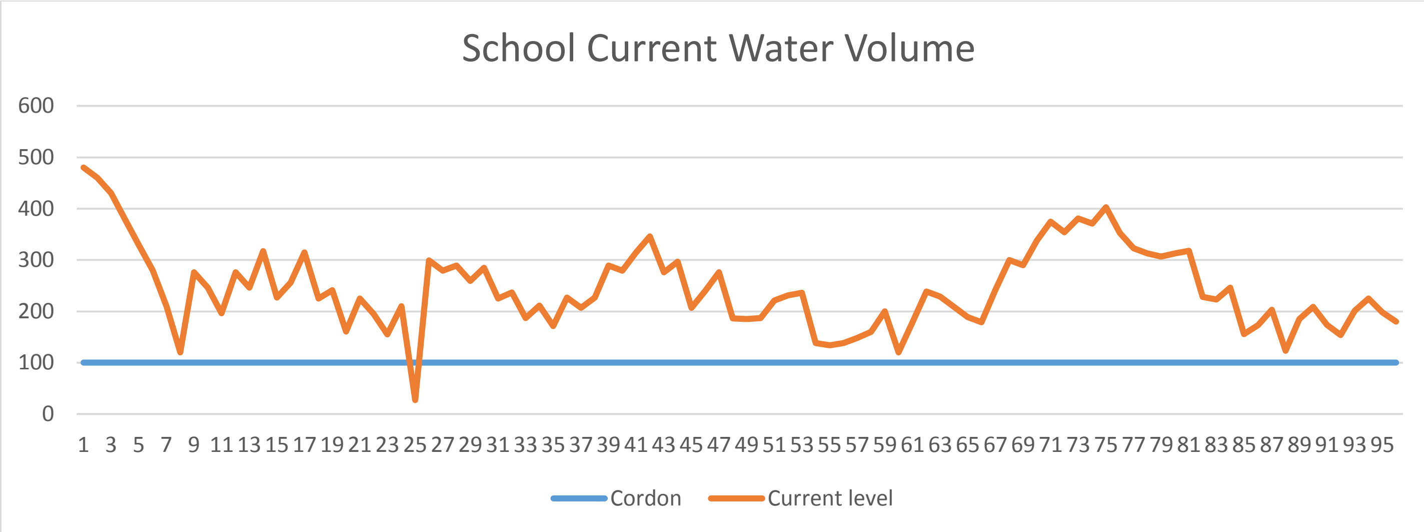
The following figures are the result of my object shown by charts. As you can see, each peak of water demand corresponds to a decreasing of current water level in reservoir, and an increase of water injection.

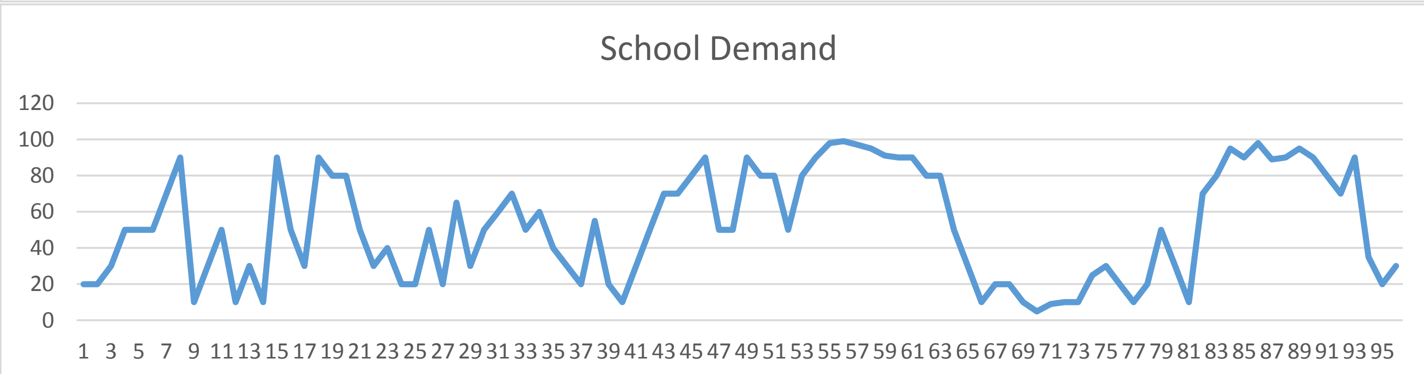

Hosptial Demand
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95

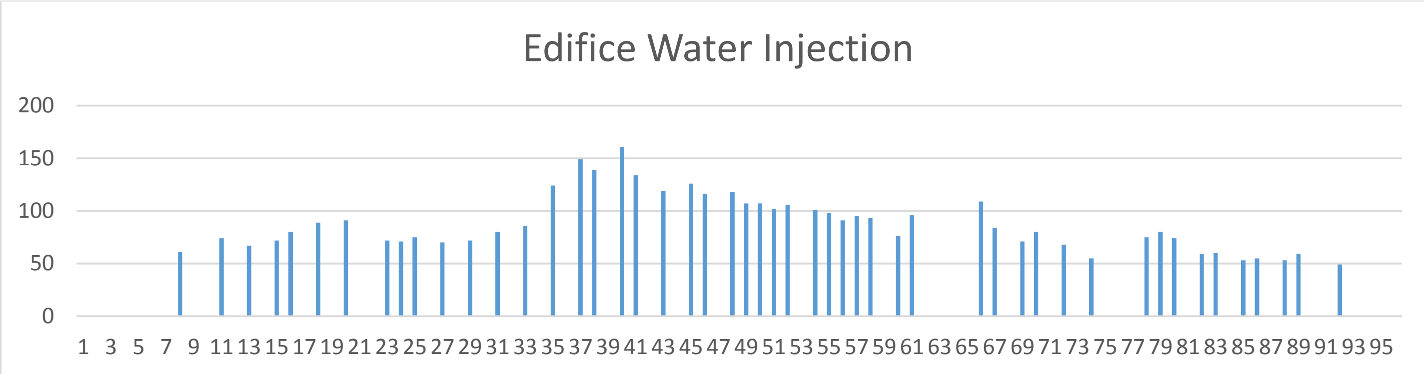

Hosptial Water Injection
250
200
150
100
50
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95

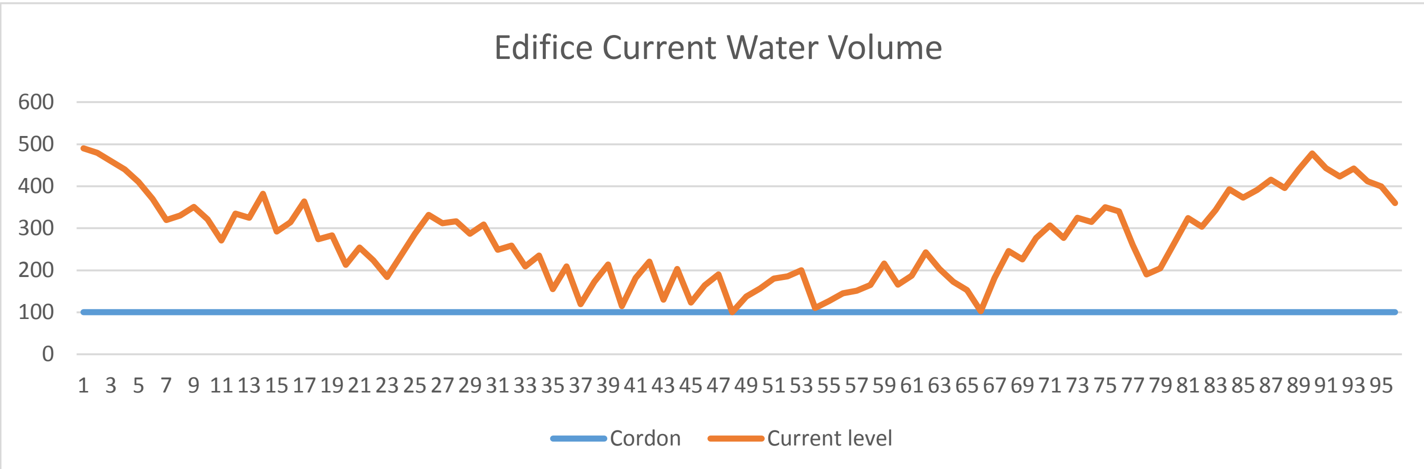

Hosptial Current Water Volume
600
500
400
300
200
100
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95
Cordon
Current level

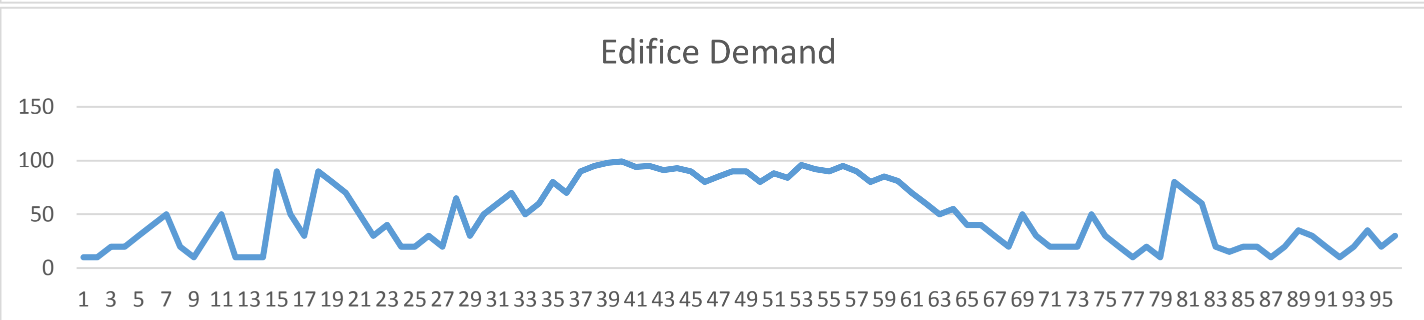

School Water Injection
180
160
140
120
100
80
60
40
20
0
166
90
81
79
89
96
114
75 76
82
84
86
66
81 81
108
91
56
52
64
86
100
93
132
120
114
77
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95


School Current Water Volume
600
500
400
300
200
100
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95
Cordon
Current level


School Demand
120
100
80
60
40
20
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95







The possible enhancements to study

There are several aspects to enhancements:

1) Deviation handling capacity. For example, time for generator to repair. Demand for holidays and weekend. Emergency events etc.
2) Generator has different capacity.
3) Consumption of heat loss during transit

Appendix

Operating environment for program is JAVA in Eclipse. Click “run”. At end of the console, choose the pattern you want. There are two pattern to display. “1. Print Schedule Sheet. 2. Run System Simulator”. Please enter 1 or 2 to see the result.