Optimal Load Assignment while Ensuring Fair Usage and User Satisfaction via Entropy Maximization

Aman Madaan[1]
Under the guidance of Prof. Ganesh Ramakrishnan[1]

IIT Bombay, Mumbai, India

Abstract. We study the problem of finding optimal switch on time of Fans and ACs in a building to maximize the satisfaction of inhabitants under given power constraints. We pose the problem as the classical convex optimization problem of entropy maximization under certain assumptions. This novel formulation helps in optimizing both inter room and intra room satisfaction of the inhabitants, at the same time enforcing fair allocation.

Keywords: Convex Optimization

1 Introduction

Section 2 introduces the problem statement, Section 3 defines satisfaction criteria, Section 4 discusses the optimization constraints, Section 5 presents the complete formulation and finally Section 6 discusses the code and results.

2 Problem Statement

The problem is described in detail in the appendix. A summarized version of the problem statement follows. KreSit building consists of a number of rooms, each with certain number of ACs, Fans and Computers. Given the number of users who sit in the building at different times, we want to find the optimal load (the number of ACs and Fans) switched on that will maximize the satisfaction of the inhabitants of the building, where we define satisfaction in section and when the total power (energy per unit time) that can be spent is a constant.

We assume that the computer systems have to be turned on at all times, since there is nothing that can be done about that. Thus, whatever powe that is left for us to spend is discounting the power that is spent by the computer systems.

Before proceeding, we discuss the notations used throughout the paper.

2.1 Notations

- n_1, The number of rooms in Kresit Building.
- n_2, The number of inhabitants.
- a, A vector of size n_1 x 1 specifying the number of ACs in each room.
- f, A vector of size n_1 x 1 specifying the number of fans in each room.
- O, A binary matrix of size $n_2 x n_1$. $O(i,j) = 1$ if i^{th} user sits in the j^{th} room.
- U, A binary matrix of size $n_2 x 24$. $O(i,j) = 1$ if i^{th} user sits in allotted room at the j^{th} hour.

2.2 The Total User Matrix

In addition to the given constraint, we use the Total User Matrix, T, throughout the paper. It is defined as follows:
$T = O^T * U$, a matrix of size $n_1 \times 24$.
$T(i,j) = K, K \geq 0$ if K users sit in the i^{th} room at j^{th} hour.

3 Defining User Satisfaction

In this section, we discuss the satisfaction criteria, and the constraints and objective that it leads to. We note that there are two different kind of satisfaction criteria, and start by discussing them. We then disuss them below.

3.1 Satisfaction Criteria

- **Intra Room Satisfaction** An inhabitant of the room is satisfied if the temperature of the room is not too hot. Usually, people would like the temperature to become as low as possible, till it gets to around 20C.
- **Inter Room Satisfaction** An inhabitant of the room would usually like the amount of air conditioner/fan services to be equal across the rooms. Ensuring inter room satisfaction implicitly ensures fair usage of resources. To define inter room satisfaction, we first need to define the cooling capacity of a room.

$$Cooling\ capacity\ of\ a\ room = \frac{Number of ACs^2 + F}{Number\ of\ Inhabitants} \tag{1}$$

We say that the inter-room satisfaction is optimum if cooling capacity of all the rooms is equal.

4 The Optimization Objective and the constraints

Armed with a definition of user satisfaction, we can proceed to define an optimization objective.

4.1 Variables

- F, a matrix formed using repMap of f 24 times, denoting the number of fans in each room.
- A, a matrix formed using repMap of a 24 times, denoting the number of ACs in each room.
- T, the total user matrix as defined
- Fs, As, the variables to be determined.

All the constraints are listed in the less than or equal to form to be consistent.

4.2 Constraints

Intra room satisfaction Constraints Based on personal observation, having *one fan per person* in a room is a must. Based on the definition of total matrix, we gets

$$T \leq Fs \tag{2}$$

Air conditioners start giving diminishing returns after a point, and unlike fans, are not really not mandatory. We thus make a casual constraint of the number of ACs begin at most equal to the square root of the fans in the room.

$$As \leq \sqrt{(}Fs) \tag{3}$$

Non negativity constraints These follow from the definition of Fs ad As

$$-Fs \leq 0 \tag{4}$$
$$-As \leq 0 \tag{5}$$

Resource Constraints Again follow from the definition of the variables, the number of ACs and Fans turned on in any room at a particular hour canot exceedthe number of ACs and fans that are available.

$$As \leq A \tag{6}$$
$$Fs \leq F \tag{7}$$

Power Constraints As stated, we assume that all the computers have to be turned on at every hour where someone is present in the room, and thus total power that is left to be given to the ACs and the fans is discounting that. **There is an implicit assumption here that people favor doing work over doing work in a cool environment (*turn on fans etc. only if you are left with power after giving it to the system*).**

The power left, P' is thus defined as follows:

$$P' = P - \sum_{i=1}^{n_1} \sum_{j=1}^{24} T_{ij}$$

We next assume that the ACs and the Fans consume the same amount of power irrespective of the time for which they are switched on, we relax this assumption later.

The constraint thus becomes the following:

$$\sum_{i=1}^{n_1} \sum_{j=1}^{24} As_{ij} * y_1 + \sum_{i=1}^{n_1} \sum_{j=1}^{24} Fs_{ij} * y_2 \leq P' \tag{8}$$

Where P' is as defined above.

No Wastage Constraints Having more ACs in a room than inhabitants is a clear wastage of resources, we add another constraint to prevent this situation:

$$A <= T \tag{9}$$

4.3 Motivating the optimization objective

We first argue that the fair usage policy/inter room satisfaction can be modeled using entropy of a probability distribution. We want that the

Entropy The entropy of a set x of non negative values is defined as follows:

$$-\sum_i x_i * log_2 x_i$$

As per the convention, $0 * log_2(0)$ is considered to be equal to 0.

Entropy of a probability distribution can be informally defined as a measure of how spread out are the values. For example, consider the following two set of numbers:

```
x = [0.3 0.3 0.4]
y = [0.9 0.05 0.05]
```

Entropy of the more evenly spread x is 1.57, whereas entropy of somewhat skewed up y is 0.5025.

Convexity of Entropy Maximization Follows from the fact that sum of convex functions is a convex function. Page 228 of [1] goes into the details of the dual formulation.

4.4 Optimization Objective

The set of cooling capacities We have already seen the definition of cooling capacity. Let us define C, the set of cooling capacities of different rooms, as follows.

For a given room i, at a given hour j, define the cooling capacity to be as follows:

$$C_{ij} = \frac{As_{ij}^2 + Fs_{ij}}{T_{ij}} \tag{10}$$

$$C = \{C_{ij} | i \in [1, n_1], j \in [1, 24]\} \tag{11}$$

4.5 The optimization objective

$$\begin{aligned} \underset{Fs,As}{\text{minimize}} \quad & -\mathit{entropy}(C) \\ \text{subject to} \quad & \mathit{Constraints}\ 2\ \mathit{through}\ 9 \end{aligned}$$

5 The convex optimization program

We now present the complete convex optimization program as follows. The constraints are augmented with a short description to make the program readable.

$$\underset{Fs,As}{\text{minimize}} \qquad \sum_{i=1}^{n_1}\sum_{j=1}^{24}\left(\frac{As_{ij}^2 + Fs_{ij}}{T_{ij}}\right) * log_2\left(\frac{As_{ij}^2 + Fs_{ij}}{T_{ij}}\right)$$

subject to

$$T \leq Fs \qquad \text{(Everyone gets a Fan)}$$

$$As \leq \sqrt{(}Fs) \qquad \text{(But not too many ACs)}$$

$$-Fs \leq 0 - As \leq 0 \qquad \text{(Can't have negative number of fans and ACs)}$$

$$As \leq A, Fs \leq F \qquad \text{(We can't switch on more fans and ACs than are present)}$$

$$As <= T \qquad \text{(No wastage)}$$

$$\sum_{i=1}^{n_1}\sum_{j=1}^{24} As_{ij} * y_1 + \sum_{i=1}^{n_1}\sum_{j=1}^{24} Fs_{ij} * y_2 \leq P' \qquad \text{(Must take care of the Budget)}$$

6 Experiments

The cvx program is as follows:

```
%sg
%Takes the values of fans and ACs in different rooms and then finds the optim
%This assumes that the supplied power is after discounting the power used in
function [Fs, As] = solve_elec(T, Fm, Am, P, y1, y2)
    row_t = size(T, 1);
    col_t = size(T, 2);
    cvx_begin
        variable F(row_t, col_t);
        variable A(row_t, col_t);
        variable E;
        %disp(size(E))

        maximize entr(E)
        subject to
            E == (F + A) ./ (T + ones(row_t, col_t))
            T - F <= 0
            -F <= 0
            -A <= 0
            A .* A <= F
            A <= T
            F <= Fm
            A <= Am
            sum(cumsum(F).*y1 + cumsum(A) .* y2) <= P
    cvx_end
    Fs = F
    As = A
```

6.1 Relaxing constant load assumption

We take the cost to be a product of the *ys* with cumulative sum of F and A, and thus a device switched for long times will consume more power.

6.2 Sample Run

We consider the following small example. We have 3 rooms, 4 users. Users only sit in the lab from 0900-1700. The first 2 labs have one fan each, and the last lab has 2 fans. All the labs have 1 AC each. This corresponds to the following matrices:

```
>> F

F =
```

```
  Columns 1 through 12

     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     2     2     2     2     2     2     2     2     2     2     2     2

  Columns 13 through 24

     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     2     2     2     2     2     2     2     2     2     2     2     2

>> A

A =

  Columns 1 through 12

     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1

  Columns 13 through 24

     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     1     1     1     1     1     1     1


>> o

o =

     1     0     0
     0     1     0
     0     0     1
     0     0     1

>> u

u =

  Columns 1 through 12
```

```
     0     0     0     0     0     0     0     0     0     1     1     1
     0     0     0     0     0     0     0     0     0     1     1     1
     0     0     0     0     0     0     0     0     0     1     1     1
     0     0     0     0     0     0     0     0     0     1     1     1

  Columns 13 through 24

     1     1     1     1     1     0     0     0     0     0     0     0
     1     1     1     1     1     0     0     0     0     0     0     0
     1     1     1     1     1     0     0     0     0     0     0     0
     1     1     1     1     1     0     0     0     0     0     0     0

 >> T

T =

  Columns 1 through 12

     0     0     0     0     0     0     0     0     0     1     1     1
     0     0     0     0     0     0     0     0     0     1     1     1
     0     0     0     0     0     0     0     0     0     2     2     2

  Columns 13 through 24

     1     1     1     1     1     0     0     0     0     0     0     0
     1     1     1     1     1     0     0     0     0     0     0     0
     2     2     2     2     2     0     0     0     0     0     0     0


fs =

  Columns 1 through 7

    0.6667    0.6667    0.6667    0.6667    0.6667    0.6667    0.6667
    0.6667    0.6667    0.6667    0.6667    0.6667    0.6667    0.6667
    0.6667    0.6667    0.6667    0.6667    0.6667    0.6667    0.6667

  Columns 8 through 14

    0.6667    0.6667    1.0000    1.0000    1.0000    1.0000    1.0000
    0.6667    0.6667    1.0000    1.0000    1.0000    1.0000    1.0000
    0.6667    0.6667    2.0000    2.0000    2.0000    2.0000    2.0000

  Columns 15 through 21
```

```
    1.0000    1.0000    1.0000    0.6667    0.6667    0.6667    0.6667
    1.0000    1.0000    1.0000    0.6667    0.6667    0.6667    0.6667
    2.0000    2.0000    2.0000    0.6667    0.6667    0.6667    0.6667

  Columns 22 through 24

    0.6667    0.6667    0.6667
    0.6667    0.6667    0.6667
    0.6667    0.6667    0.6667


as =

  Columns 1 through 7

    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000

  Columns 8 through 14

    0.0000   -0.0000    0.3333    0.3333    0.3333    0.3333    0.3333
    0.0000   -0.0000    0.3333    0.3333    0.3333    0.3333    0.3333
    0.0000   -0.0000    0.0000    0.0000    0.0000    0.0000    0.0000

  Columns 15 through 21

    0.3333    0.3333    0.3333   -0.0000   -0.0000   -0.0000   -0.0000
    0.3333    0.3333    0.3333   -0.0000   -0.0000   -0.0000   -0.0000
    0.0000    0.0000    0.0000   -0.0000   -0.0000   -0.0000   -0.0000

  Columns 22 through 24

   -0.0000   -0.0000   -0.0000
   -0.0000   -0.0000   -0.0000
   -0.0000   -0.0000   -0.0000
```

References

1. Boyd, Stephen, and Lieven Vandenberghe. Convex optimization. Cambridge university press, 2004.

Complete Problem Statement The KReSIT department has n_1 rooms. The vector A specifies the number of A/Cs in each room and the vector F specifies the number of fans in each room. Another vector C specifies the number of computer tables in each room, where a computer table is assigned to a specific student/staff. The power consumption of each electrical equipment is as follows (as measured in watts):

fan = $y_1 * x_1$, A/C = $y_2 * x_2$, computer = y_3.

where x_1 is any real number in the range $[0, v_1]$ and x_2 is any real number in the range $[0, v_2]$ - fans and A/Cs will consume power that is a linear function of "how much" they are "switched on" which is captured in x_1 and x_2.

You are also provided

a matrix O of user occupancies: Each row corresponds to an occupant of KReSIT (with n_2 rows) and each column corresponds to a room (with n_1 rooms). An entry is 1 if the user sits in that room and 0 otherwise.

a matrix U of user behavior: Each row corresponds to an occupant of KReSIT (with n_2 rows) and each column corresponds to an hour of the day (with 24 columns). An entry can be either 0 or 1.

Finally, there is a constraint that the total energy consumption in KReSIT within a day cannot exceed P units.

Recall the connection between power and energy: power is energy consumed per unit time.

We want to pose an optimization (maximization or minimization) problem, whose outcome is "how much" each fan and A/C should be turned on in each room at each hour of the day.