

CS515: Computer System Lab 2

Date: 31st Mar 2022

Submission Filename: (assign09.tgz)

Assignment 09

Due Date: 06th Apr 2022

1 Overview

In this assignment, you need to develop a framework for comparing a set of the approximation algorithms whose objectives are same. After implementing those algorithms, you need to develop a comparative framework using shell script. Comparison of various algorithms will be done based on

- final output
- execution time

2 The algorithms that need to be considered

We are given a household H , which contains a set of appliances A that run for a total of T hours, $\{1...T\}$. The appliances can start working from the first till the T^{th} time slot. There are 4 modes of operation for each appliance $a \in A$:

- **Slow:** At every time slot for slow mode of operation, energy consumed in this mode=2 KWh and price paid by the household=1Rs.
- **Moderate:** At every time slot for moderate mode of operation, energy consumed in this mode=4 KWh and price paid by the household=2Rs.
- **Fast:** At every time slot for fast mode of operation, energy consumed in this mode=6 KWh and price paid by the household=4Rs.
- **OFF:** At every time slot when the appliance is OFF, energy consumed in this mode=0 KWh and price paid by the household=0.75Rs.

Due to grid load constraints, at any time, a maximum of 2 appliances are allowed to operate in a single mode of operation. For example, if there are 7 appliances, maximum 2 can operate in slow mode, 2 in moderate, 2 in fast mode, and minimum one of them will have to be OFF. However, there is a minimum requirement of the household R_{min} KWh per day, where the total energy consumed is the sum of the energy consumed by each appliance per time slot.

The total money paid by the household is based on the usage of these appliances in the entire day. Our objective is to schedule these appliances in a certain manner that the total money paid is minimum and R_{min} is respected.

A list of algorithms is provided. For each of the following, consider two broad disciplines of scheduling: Queue Based (QB) and Random Based (RB). The approaches for any scheduling can be considered as follows.

1. **Fast charging first (FCF):** Here, a Fast mode is assigned to two appliances, Moderate mode to the next two and so on, per time slot. The priority given is, Fast > Moderate > Slow.
2. **Moderate charging first (MCF):** Here, a Moderate mode is assigned to two appliances, Slow mode to the next two and so on, per time slot. The priority given is, Moderate > Slow > Fast.
3. **Slow charging first (SCF):** Here, a Slow mode is assigned to two appliances, Moderate mode to the next two and so on, per time slot. The priority given is, Slow > Moderate > Fast.

Note: Any appliance can be switched OFF as and when necessary. In all the approaches, the priorities work, iff the appliance is ON.

We have now 6 algorithms: QB_FCF, QB_MCF, QB_SCF, RB_FCF, RB_MCS, RB_SCF. In case of QB, the ordering of the above three approaches should be based on the placement of the appliances in a queue. As a convention, we can assume the appliances are in the order $\{a_1, a_2, a_3, a_4, a_5\}$ when $|A|=5$, with a_1 being the front and a_5 being the rear end. For RB, an appliance can be picked in any order. An example test case is provided, for your reference.

3 Sample Example

In the Fig. 1, the energy consumed in both the examples is 66KWh. However, the price paid in QB_FCF is 45.75Rs, whereas in RB_FCF is 43Rs.

Please note that the above examples are only for the explanation purpose and need not necessarily be optimal.

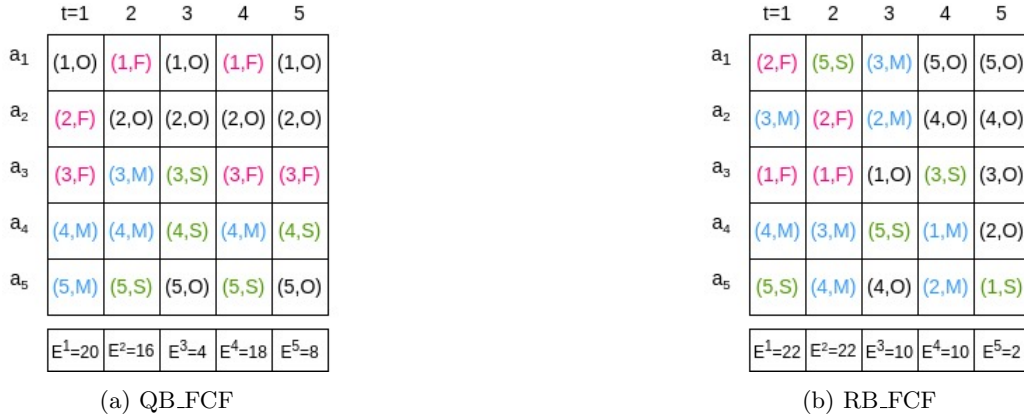


Figure 1: An example figure for FCF using (a) the QB schedule, and (b) using RB schedule, when $R_{min} = 60$.

4 Developing the Comparative Framework

These algorithms need to be run on various input sizes. On the same input data, all the algorithms must be run a fixed number of times. For all these different runs on the same input data, the output Price paid and execution time need to be recorded. The average of these will have to be used for comparison purpose. For better visualization, the average number of comparisons and execution times will have to be plotted using *gnuplot* tool.

One will use the following command in a linux shell to analyze the performance of various algorithms

```
$ ./testhouse -n <A_size> -t <T> -r <R_min> -s <step_size> -a <avg> -m [QB_FCF, QB_MCF, QB_SCF, RB_FCF, RB_MCS, RB_SCF] -o <out_filename> -g <plot_script_file_name>
```

- **n** **<A_size>** : To specify minimum number of appliances. Minimum value is 5.
- **t** **<T>** : To specify the total time horizon in a day. Minimum value is 5.
- **r** **<R_min>** : To specify the minimum value of R_{min} . Minimum value should be 50, and maximum value should be 300.
- **s** **<step_size>** : To specify the step size of the value of R_{min} . Default value is 10.
- **a** **<avg>** : An algorithm will be run avg number of times on the same input set and finally average will be taken as an estimate for the performance measure. Default value of avg is 1.
- **o** **<out_filename>** : Performance measure will be stored in this file. Default filename is output.txt.
- **g** **<plot_script_file_name>** : You need to generate *gnuplot* script to plot the output that you have obtained in output.txt. Default value is plot.gnu.
- **m** **[]** : User can choose any number of methods out of the six methods - QB_FCF, QB_MCF, QB_SCF, RB_FCF, RB_MCS, RB_SCF. Default values are QB_FCF and RB_FCF.

Please note the followings-

1. You need to use the same input for all the algorithms specified by **m** option.
2. For your comparison graph for the execution time, the independent variable should be the values of R_{min} and the dependent variable should be the average time taken.
3. For your comparison graph for the Price paid, the independent variable should be the values of R_{min} and the dependent variable should be the corresponding average values obtained for Price paid.
4. For a given value of R_{min} , any schedule that results in the total energy consumption between R_{min} and $R_{min} + \text{step_size}$ is a valid schedule.
5. Please provide *makefile* to build executable(s) for your C/C++ code

5 Submission

Prepare a **README.txt** file containing the steps need to be followed.

All the necessary source files along with the README.txt should be archived to **assign09.tgz**. You can use `tar -cvzf assign09.tgz folderName` to create such *tgz* file.

Reference

1. GNUPLOT

http://www.gnuplot.info/docs_4.0/gpcard.pdf

2. Makefile

<http://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/>