Indian Institute of Information Technology Vadodara (Gandhinagar Campus)

Research Internship Report-2022

on

Efficient Scheduling of Non-Preemptive Appliances for Peak Load Optimization

Submitted by

Name	Roll Number
Hritik Mahajan	**201951071**
Sudhanshu Kumar	**201951154**

under the supervision of

Dr. Sunandita Debnath

***Abstract*—Due to the current electrical grid system, overloading, peak-load consumption, and other issues are a problem in the modern world. This is a result of the limited ability of energy creation to be monitored and controlled in real-time. We have worked on load scheduling and controlling algorithms [1] during our research internship to schedule controllable appliances with the goal of reducing peak load consumption. To do this, we represent the issue as the strip packing issue. Then, in our problem setup, we worked on the pre-existing heuristics. Additionally, we have contrasted various methods with the Min peak algorithm and talked about how much peak load each technique consumes.**

We have also talked about the time complexity of various heuristic methods.

I. Introduction

A. Overview

In the present global environment of rising demands to reduce greenhouse gas emissions and provide cleaner energy, it is critical to use existing energy infrastructure considerably more efficiently than ever before. According to the newly released International Energy Outlook 2016 (IEO) **[2]** study, worldwide energy consumption is anticipated to increase by **48 percent** from **549 quadrillion British thermal units (Btu) in 2012 to 815 quadrillion Btu by 2040**. Even though renewable energy sources such as wind, hydro, and solar have gradually been integrated into existing infrastructure, the current grid system requires a paradigm shift in overall energy production and efficient consumption structure, mitigating problems such as inefficient energy usage, grid instability, power outages, and so on.**[3]**

Peak load usage also raises system operational costs, which subsequently raises customers' power bills, particularly when users are paid based on peak load consumption as well as overall energy consumption, a frequently used dynamic pricing mechanism **[4], [5]**.

In the past, two methods for reducing peak load consumption have been reported: reduce consumption and shift consumption**[6]**. Whereas lowering power consumption necessitates the development of energy-efficient equipment, shifting consumption is a simple and cost-effective approach that alters users energy consumption habits and transfers consumption to off-peak hours to balance total load consumption patterns.

B. Problem definition

1) Goals: Given a collection of deferrable appliances, as well as their unique power consumption and needed execution time or running time, the task is to schedule all of them inside a specified time horizon so that the peak load consumption within that time horizon is as low as possible.

All of the deferrable appliances in this section are non-preemptive in nature. Non-preemptive Scheduling is a CPU scheduling strategy in which the process takes a resource (CPU time) and retains it until the process is terminated or moved to the waiting state. No process is interrupted until it is completed, at which point the processor switches to another**[7]**.

2) Objectives:

- **Implemented and discussed MinPeak, a new two-dimensional strip packing-based offline algorithm designed exclusively for scheduling electrical appliances to reduce peak power usage.**
- Compared our Minpeak algorithm with existing heuristic algorithms like BL, BLDH, etc.
- Compared our Minpeak algorithm with respect to time complexities and maximum peak load value.
- Provided the implemented algorithm's worst-case approximation bound and analyze its time complexity.

II. RELATED WORK

Recently, several methods to reduce peak load consumption have been proposed, including load scheduling programs, fuel substitution programs, conservation and energy efficiency programs, and programs for energy efficiency**[5]**. The scheduling for peak minimization problem has been handled in a number of articles as a two-dimensional strip packing (2SP) problem, where the appliances can be thought of as tasks, with execution times as widths and power consumption as heights.The issue has previously been addressed by a number of heuristic approaches.

Bottom left (BL), put out by Baker et al.**[8]**, is one of the most popular heuristic approaches for strip packing. The BL technique involves processing and scheduling each item one at a time, first into the smallest time periods, and then leaving adjustments. As an arbitrary sequence of input can perform very poorly in relation to an optimization problem, the input sequence of the items can also be pre-processed in order to enhance the performance of the fundamental BL technique.

Best fit decreasing height (BLDH) is one of the most used algorithms in this the appliance with maximum power consumption is scheduled first. And if the device to be scheduled has time period less then that left in the strip then we will schedule other appliance with greater power consumption. Best fit decreasing width (BLDW) same as that of BLDH instead of maximum power consumption it uses maximum time period. In Bottom-left fill decreasing height(BLFDH) is similar to the BLDH but if there is some space created or some unfilled time slots then we can fill that time slot with the appliance.

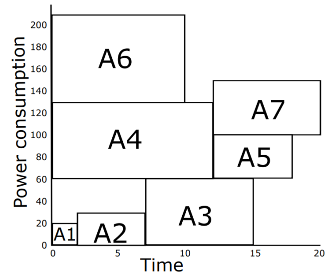
III. MOTIVATING EXAMPLE

Let's have a look at seven deferrable non-interruptible appliances (Ai, I = 1, 2,..., 7), each of which is defined by an ordered pair (Ai = (*Ei*,*Pi*)) that represents the execution time (*Ei*) and power consumption (*Pi*) of appliance *i* shown in Table 1. Execution time and power usage are taken to be in the correct units. In this instance, let the scheduling horizon T(scheduling deadline) be 20. The challenge is to schedule all the appliances by the deadline while minimising the peak load demand given these inputs.

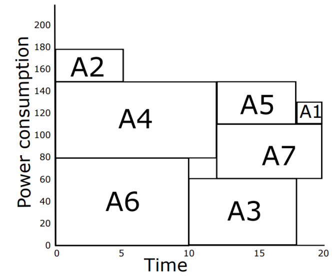
TABLE 1: Input specifications for seven appliances

Appliance	*Execution time*	*Power consumption*
A1	2	20
A2	5	30
A3	8	60
A4	12	70
A5	6	40
A6	10	80
A7	8	50

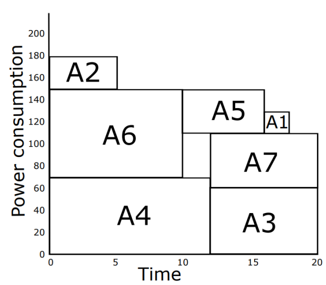
For this example the peak load of BL is 210, for the algorithm BLDH it is 180, for BLDW it is 180, for BLFDH it is 180 and for the **MinPeak it is 180**.



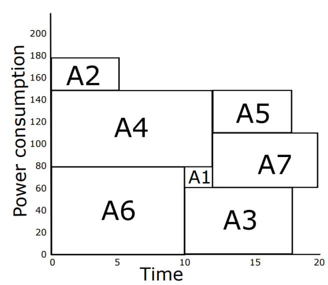
(a) Scheduling using *BL*



(b) Scheduling using *BLDH*

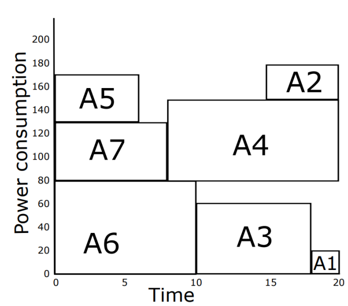


(c) Scheduling using *BLDW*



(d) Scheduling using *BLFDH*

Scheduling using existing heuristic algorithms



Scheduling with MinPeak

IV. OFFLINE SCHEDULING MECHANISM

A. Introduction

In this part, we describe and implemented MinPeak, a 2SP-based load regulating and offline scheduling method used to schedule appliances that attempt to optimize peak power usage. *Algorithm 1* shows the pseudo code for the same.

The implemented technique takes as inputs a collection of appliances with their individual power consumption (*Pa*) and execution time (*Ea*), as well as the scheduling time horizon *T*. It also accepts a load array (L) of size *T* as input, which comprises the overall power consumption at various time instants. As a result, L[*t*] represents total load consumption at time *t*. When all appliances are of the deferrable type, all L indices are set to 0. When there are both deferrable and non-deferrable appliances, the relevant indices of L are changed to reflect the power and temporal needs of the non-deferrable appliances, and *Algorithm 1* is used to schedule the deferrable appliances.

B. Methodology

The above mentioned method processes and schedules each device from the pool of deferrable appliances one by one. We determine the best interval for scheduling an appliance based on the load array. The process is repeated until all of the appliances have been scheduled. The algorithm is divided into three steps, each of which is detailed below:

Phase 1: The suggested MinPeak method first checks the feasibility of the input requirements by comparing the execution times of all the appliances. If *Ea* is greater than the time horizon for any appliance *a*, the appliance cannot be scheduled, and the algorithm terminates claiming in-feasibility. The scheduling procedure begins only once the input parameters have been verified and determined to be viable.

Phase 2: After determining that the input parameters are viable in Phase 1, the algorithm calculates the best potential time slots for the appliances to be scheduled. The optimal time slots for a scheduled appliance are defined as continuous slots of length equal to *Ea* in the load array L with the lowest overall load consumption. This is accomplished by identifying the shortest load sub-array of length *Ea* inside L in order to schedule appliance *a*.

The matching time slots are calculated by determining the minimum value of a sub-array whose length is precisely equal to the execution duration of appliance *a*. We select the smallest feasible sub-array for the *a*th appliance after computing for all potential sub-arrays. If there are numerous sub-arrays, we consider the one with the lowest peak, and ties are broken arbitrarily if multiple have the same peak. The value obtained is the optimal start time frame for which the appliance *i* is scheduled. This process is performed for each appliance in the set.

Phase 3: Following the conclusion of phase 2, which schedules all of the appliances, the proposed method computes peak load consumption by calculating the greatest value in the L array.

Algorithm 1 Implemented MinPeak Algorithm

Input: Consider *m* deferrable appliances (*Ad*) with the following specifications: power consumption *(P = P1,P2,...,Pm)*, execution time *(E = E1, E2,..., Em)*, scheduling time horizon (T), and the initialised load array (L). A unit length time interval is considered. We have taken input parameters in pair of power consumption and execution time in vector A.

Output: Schedule each device one by one with optimized peak load consumption.

- We check the feasibility of input appliances. For every device, if execution time(*Ea*) is less than or equal to time horizon(T), then input devices are feasible for scheduling. Else, input appliances is infeasible for MinPeak algorithm. EXIT
- First sort the input appliances w.r.t power consumption.
- Now, starting with the first appliance, perform the following for each appliance*(i)* in the list *Ad*:
- We find the first strip for which remaining time of that strips should be greater than or equal to *Ei*. If found such strip, then schedule device *Ai* in that strip. Else create new strip and schedule it at last of that strip.
- All appliances are scheduled from left to right for particular strip inorder to minimize Peak Load except for the first strip.
- Remove *i*th appliance from *Ad* and schedule the next device following the same algorithm till *Ad* doesn't gets empty.
- After *Ad* gets empty, find the maximum peak load by adding all values of strips column-wise.
- Maximum value among them is our Peak load value.

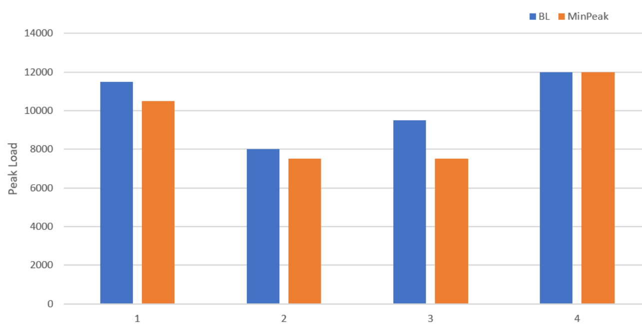
V. TIME COMPLEXITY OF ALGORITHMS

The time complexity of the Bottom left (BL) algorithm is O(k),because the scheduling of the appliances is done upon the sequence of the element provided Whereas in Best fit decreasing height (BLDH) it requires the appliances in the decreasing order of the power consumption therefore its time complexity is O(klogk), in case of Best fit decreasing width (BFDW) it is similar of the BLDH because in this algorithm it requires appliances decreasing order of their execution time. Whereas in minpeak algorithm the time complexity is O(k^2).

Algorithms	*Time Complexity*
BL	*O(k)*
BLDH	*O(k^2)*
BLDW	*O(k^2)*
BLFDH	*O(k^3)*
MinPeak	*O(k^2)*

VI. Result and Discussion

We have implemented BL(in java) and Minpeak(in cpp) algorithm and compared the peak load value of BL and Minpeak on the same data. From the result of the diagram we can see that Minpeak is able to minimize the peak load value. And we conclude that minpeak is better algorithm to control the peak load.



Comparision of BL and MinPeak Algorigthm

Acknowledgments

We would like to convey our sincere gratitude to Dr. Sunandita Debnath for her insightful advice and constructive criticism during the conception and planning of this research project. Her willingness to so kindly donate her time has been much appreciated.

References

[1] Chakraborty, Nilotpal, Arijit Mondal, and Samrat Mondal. ”Efficient scheduling of nonpreemptive appliances for peak load optimization in smart grid.” IEEE Transactions on Industrial Informatics 14, no. 8 (2017): 3447-3458.

[2] Conti, John, Paul Holtberg, Jim Diefenderfer, Angelina LaRose, James T. Turnure, and Lynn Westfall. International energy outlook 2016 with projections to 2040. No. DOE/EIA-0484 (2016). USDOE Energy Information Administration (EIA), Washington, DC (United States). Office of Energy Analysis, 2016.

[3] Palensky, Peter, and Dietmar Dietrich. ”Demand side management: Demand response, intelligent energy systems, and smart loads.” IEEE transactions on industrial informatics 7, no. 3 (2011): 381-388.

[4] Conti, John, Paul Holtberg, Jim Diefenderfer, Angelina LaRose, James T. Turnure, and Lynn Westfall. International energy outlook 2016 with projections to 2040. No. DOE/EIA-0484 (2016). USDOE Energy Information Administration (EIA), Washington, DC (United States). Office of Energy Analysis, 2016.

[5] Mohsenian-Rad, Amir-Hamed, Vincent WS Wong, Juri Jatskevich, Robert Schober, and Alberto Leon-Garcia. ”Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid.” IEEE transactions on Smart Grid 1, no. 3 (2010): 320-331.

[6] Tang, Shaojie, Qiuyuan Huang, Xiang-Yang Li, and Dapeng Wu. ”Smoothing the energy consumption: Peak demand reduction in smart grid.” In 2013 Proceedings IEEE INFOCOM, pp. 1133-1141. IEEE, 2013.

[7] Preemptive and Non-Preemptive Scheduling, Available online: https://www.tutorialspoint.com/preemptive-and-non-preemptive-scheduling

[8] Baker, Brenda S., Edward G. Coffman, Jr, and Ronald L. Rivest. ”Orthogonal packings in two dimensions.” SIAM Journal on computing 9, no. 4 (1980): 846-855.