Offline First Fit Scheduling of Power Demands in Smart Grids*

Anshu Ranjan    Pramod Khargonekar    Sartaj Sahni

Abstract

In this paper we consider the problem of scheduling energy consumption loads in the setting of smart electric grids. Each load is characterized as a “job” by a start (arrival) time and a deadline by which a certain amount of electric energy must be delivered to the load. A job may be preemptable, i.e., it can be interrupted or non-preemptable. Specifically, we focus on scheduling a mixture of preemptable and non-preemptable jobs with the same arrival time and deadline with the goal of minimizing the peak power. We study and modify the first fit decreasing height algorithm of the strip packing problem for this purpose. We derive a performance bound for the algorithm and prove its tightness. We test the performance of the algorithm extensively on a variety of datasets including real life household data.

1 Introduction and Related Work

Visions of smart electric grids are focused on bringing about a variety of improvements in the existing aging power grid infrastructure. Power system capacity needs, in terms of generation, transmission, distribution, infrastructure, are dictated largely by the need to meet peak power demand (which often occurs on summer afternoons) with required levels of reliability in the presence of naturally occurring random disturbances and failures. Since this peak demand occurs in relatively short fraction of the entire year, much of the capacity is not utilized for large parts of the year. In 2009, for example, 15% of the generation capacity in Massachusetts was utilized less than 88 hours [11, 12] or 1% of the year. Therefore, it is desirable to reduce the peak power demand as it has a direct impact on the need to invest in expensive infrastructure. Smart electric grids involve integration of a communication link between the electric energy demands, e.g., household appliances, electric vehicles, etc. The idea is to use two-way communications between demands and the utility company (or an intermediate load aggregator [4]) and exploit the inherent flexibility in electric loads to time-shift them, thereby reducing the peak power. The idea of exploiting load flexibility has attracted considerable attention in the smart grid literature as can be seen in numerous papers in various journals and conferences in power, controls, communications, and computing fields.

In this paper, we consider a sub-problem of the original scheduling problem of flexible loads in Smart Grids. Electric power loads can essentially be of three types: (a) those that have no flexibility, e.g., fans, TV, etc., (b) those that can served later but cannot be interrupted once started, e.g., industrial ovens, and (c) those that can be served later and do not need continuous power e.g. electric vehicles, washing machines, dryers, etc. We note that though the demands of type (c) tolerate intermittent power supply, minimizing the number of breaks in the supply is desirable. We consider the problem of scheduling a collection of jobs of type (b) and (c) with

---

*This research was supported, in part, by the National Science Foundation under grants CNS0905308 and CNS-1239274.
the same arrival time and deadline and the goal of minimizing the peak power. As in [1], we refer to this problem as offline cost optimal scheduling problem (OCOSP). In case the problem has only jobs of type (b) (respectively, type (c)), we refer to the problem as non-preemptive (respectively, preemptive) OCOSP. If the jobs are a mix of types (b) and (c), we call it mixed OCOSP. We note that since we do not have control over the demands of type (a), we focus simply on load types (b) and (c). If we include the demands of type (a) to OCOSP, the only difference would be that instead of having of having a flat base, we would have a time-varying base.

In recent years this class of scheduling problems in Smart Grids and its variants have attracted significant research interest. In [5] Koutsopoulos et. al. have proposed that power-cost function is convex when (preemptable) jobs have (possibly) different start times and deadlines using the load balancing algorithm of [6]. They show that the algorithm converged to the optimal solution but is expected to be inefficient. They also propose asymptotically optimal scheduling policies for infinite time horizons. Alamdari et. al. [11] propose a fully polynomial time approximation scheme for preemptive OCOSP, but since they use the ellipsoid method in their linear program of FPTAS, it is not expected to run in acceptable time. For the same problem, they have proposed approximation algorithms one with ratio 3/2 and two with ratio 5/3. In [9], Tang et. al. have proposed a 7-OPT algorithm for the Non preemptive OCOSP and 2-OPT and 4-OPT algorithms for the scheduling problem with an objective to minimize the delay of each demand. Game theoretic approaches towards power scheduling in order to achieve certain objectives like peak power minimization have been proposed in [15]-[20].

In our recent work [1], we have established a relationship between the strip packing problem and the OCOSP. The strip packing problem is to pack a set of rectangles without rotation into a rectangular strip with fixed width, with a goal of minimizing the height of the strip. This is a generalization of the bin packing problem when all the rectangles have unit height and hence is NP-hard (as can be proved with a simple reduction from the Partition problem). Also, there can be no polynomial time algorithm with an approximation ratio better than 3/2 unless \( \text{P}=\text{NP} \). This problem has been studied extensively (see [10] for more details of various algorithms on strip packing). So far the best algorithm for strip packing gives an approximation factor of \( 5/3 + \epsilon \) with \( \epsilon > 0 \) [21].

In our previous work [1], we have proposed and experimentally evaluated two 3-optimal algorithms (for non-preemptive and preemptive OCOSP) and one 2-optimal algorithm (for preemptive OCOSP). These algorithms are adaptations of the NFDH (next fit decreasing height) algorithm [8] for strip packing. We also showed that the offline scheduling problem is NP-hard when the jobs have same arrival time and deadline, for all cases: preemptables, non-preemptables or mix of the two. In this paper, we present a \( 1 + 1.7OPT \) approximation adaptation of the FFDH (first fit decreasing height) algorithm [8] of strip packing for mixed OCOSP. To the best of our knowledge, there has been no other work done related to mixed OCOSP.

The remainder of this paper is organized as follows. In Section 2, we explain the \( FFDHMix \) algorithm and prove its approximation ratio. In Section 3, we implement the algorithm and show experimental results and Section 4 is the conclusion.

## 2 First Fit Decreasing Height Scheduling

The strip packing problem is similar to OCOSP if we consider each rectangle as a demand. The width and height of the rectangle corresponds to the duration and power of the demand [1]. (We have implicitly made a simplifying assumption that each load has a constant power need. This is true, for example, for certain types of electric vehicles.) The goal of strip packing is to pack all the rectangles with minimum height which corresponds to the goal of achieving minimum peak power of OCOSP. For preemptive OCOSP, the rectangles can be sliced vertically to form
different sub-rectangles. They can then be placed at different locations with the constraint that no two or more sub-rectangles (of the same original rectangle) intersect one vertical line (stacking constraint) as illustrated in Figure 2 of [2]. For mixed OCOSP, certain rectangles can be sliced vertically and have the same constraint. The key difference between OCOSP and strip packing is that while we do not consider the empty space between the rectangles in calculating the total power in an instant for OCOSP, it is not so in strip packing. In the rest of the paper we use the terms “rectangle” and “job”, “width” and “duration” and “height” and “power” interchangeably.

First Fit Decreasing Height (FFDH) heuristic is a simple algorithm for the strip packing problem first described in [8]. In this heuristic, the rectangles are first arranged in decreasing order of height. They are then placed in this order left justified on the lowermost level of the packing where they fit. If there is no level with enough space, a new level is created. The bottom most level is the base of the strip. Each new level is created by drawing a horizontal line at the top of the leftmost rectangle (hence the tallest) of the previous level.

2.1 First Fit Decreasing Height Heuristic for Mixed OCOSP

FFDHMix is an adaptation of the FFDH heuristic for strip packing to the Mixed OCOSP problem. In FFDHMix, we first arrange the jobs in decreasing order of power demand (i.e., height). Next, the jobs are scheduled one-at-a-time in this order. If the job currently being scheduled is non-preemptable, it is scheduled on the lowest level where it fits. If there is no such level, a new level is created and the job is placed on this new level. If the job is preemptable, it is placed on levels starting from the lowest level which has available space as long as this placement does not violate the stacking constraint. If the job is not fully placed after taking into account all the existing levels, a new level is created to place the remaining portion of the job. All job placements are done left-justified on a level. The algorithm is formally written in Figure 1 and illustrated with an example in Figure 2. It is easy to see that each job is preempted at most once even though it could be sliced multiple times across different levels. Note that a preemption occurs only when there is a break in the schedule of a job. i.e., when a job is scheduled for two consecutive non contiguous time intervals.

In our proof of the tight bound for FFDHMix, we will use the notion of one job or slice of a job being scheduled/placed before another. For the case of different jobs or slices of different jobs this ordering follows from the initial sorting by height. For slices of the same job, slices at lower levels are placed before slices at higher levels.

2.2 Terminology

Our terminology is an extension of that used by Coffman et al. [8] in their analysis of the first fit decreasing height (FFDH) algorithm applied to non-preemptable jobs. This extension is needed to accommodate preemptable jobs. We remark also that our proof of the tight bound borrows heavily from the proof in [8] for the FFDH algorithm for non-preemptive strip packing.

Let $S$ be a schedule (or packing of rectangles) generated using algorithm $FFDHMix$. The rectangles in $S$ are placed on levels with level 1 being the bottommost level. We use the terms level and block interchangeably. Non-preemptable jobs are represented by exactly one rectangle in $S$ while preemptable jobs have one or more rectangles in $S$. The width of a rectangle equals the duration for which the corresponding job is scheduled and its height is the amount of power allocated. Note that the sum of the widths of the rectangles corresponding to a single job equals the duration of that job and that all of these rectangles have the same height, which equals the power requirement of the job. A preemptable rectangle is a rectangle that is part of a preemptable job while a non-preemptable rectangle corresponds to a non-preemptable job.
Algorithm \textit{FFDHMix}
\
\{

// mixed first fit decreasing height scheduling
// sort the \( n \) jobs into decreasing order of height and re-index them
// by this sorted ordering so that job 1 has maximum height;
\( i = 1, l = 0; \)
\textbf{while} \((i \leq n)\)
\{
// schedule job \( i \)
\( j = 0, \text{placedUntil} = 0; \)
\textbf{while} \((j < l)\)
\{
// check space at level \( j \)
\textbf{if} \((\text{spaceAt}[j] \geq d_i)\)
\{
\textbf{output} schedule job \( i \) from \( \text{spaceAt}[j] \) to \( \text{spaceAt}[j] + d_i; \)
\text{spaceAt}[j] - = d_i;
\textbf{break};
\}
\textbf{else}
\{
\textbf{if} \((i \text{ is preemptable})\)
\{
// slice the job and place until the stacking constraint is not violated
\textbf{output} schedule job \( i \) from \( \text{spaceAt}[j] \) to \( D - \text{placedUntil}; \)
\( d_i - = \text{spaceAt}[j] + \text{placedUntil}; \)
\text{swap}(\text{spaceAt}[j], \text{placedUntil});
\}
\}
\textbf{if} \((d_i > 0)\)
\{
// job not yet fully scheduled
// place the rest on a new level
\textbf{output} schedule job \( i \) from \( 0 \) to \( d_i; \)
\( \text{spaceAt}[l] = D - d_i \)
\textbf{l} ++
\( pMax + = p_i \)
\}
\textbf{output} Peak power is \( pMax \)
\}

Figure 1: Power scheduling using \textit{FFDHMix}
For any rectangle \( r \), its corresponding job is \( J(r) \).

A first rectangle in \( S \) is a rectangle scheduled at the leftmost position of some level of \( S \). That is, it is scheduled from time 0 to time \( \delta \) for some \( \delta > 0 \). Note that every level of \( S \) has a first rectangle. Let \( B_2 \) be the blocks (levels) of \( S \) whose first rectangle is non-preemptable and of width more than 1/2 and let \( B_1 \) be the remaining blocks of \( S \). \( H(B_1) + H(B_2) \) denotes the sum of the heights of the blocks in \( B_1 \) (\( B_2 \)). Note that the height of a block is the height of its first rectangle and the height of \( S \) is \( H(B_1) + H(B_2) \).

Example 1 Figure 2 shows the FFDHMix schedule \( S \) for a sample job set. \( S \) has 4 levels (blocks). The first rectangle of block 1 is both non-preemptable and has width more than 1/2. Also, \( B_1 = \{2, 3, 4\}, B_2 = \{1\}, H(B_1) = \frac{3}{4}, \) and \( H(B_2) = 1 \). Block 2 is the bottommost block of \( B_1 \) and block 4 is its topmost block.

For convenience, we renumber the blocks of \( B_1 \) from 1 through \( t \) bottom to top. This renumbering is used henceforth. Note that there may be one or more blocks of \( B_2 \) between two consecutive blocks \( i \) and \( i + 1 \) of \( B_1 \). As in [8] we classify some rectangles of \( B_1 \) as regular. The remaining rectangles of \( B_1 \) are fallback rectangles. Later we extend the class of fallback rectangles to include some of the rectangles in \( B_2 \). There are two types of regular rectangles in \( B_1 \)–1Reg and 2Reg and there are two types of 2Reg rectangles–2RegA and 2RegB.

Definition 1 A 1Reg rectangle is any of the following:
1. A rectangle of a non-preemptable job assigned to a block of $B_1$ when the next higher block (if any) of $B_1$ was empty.

2. The topmost rectangle in $B_1$ of a preemptable job provided that this topmost rectangle is assigned to (a) a position other than first of a block $i$ of $B_1$ when block $i+1$ of $B_1$ was empty or (b) the first position of a block $i$ of $B_1$ but not assigned to any position of block $i-1$ of $B_1$.

We observe that in case 2(b) of the preceding definition, block $i+1$ must have been empty at the time the 1Reg rectangle was assigned to block $i$ of $B_1$.

**Definition 2** A 2RegA rectangle is a rectangle of $B_1$ that satisfies the following conditions:

1. it is the topmost rectangle in $B_1$ of a preemptable job $J$
2. it is the first rectangle on some level $i$ of $B_1$
3. another rectangle of job $J$ is assigned to block $i-1$ of $B_1$.

The rectangle of job $J$ assigned to block $i-1$ is a 2RegB rectangle.

In the example of Figure 2, rectangles $r_4$, $r_8$, $r_9^{pre}$, $r_{10}$ and $r_{11}$ are 1Reg, rectangle $r_5^{pre}$ is 2RegA, rectangle $r_6^{pre}$ is 2RegB, and $r_3^{pre}$ and $r_3^{pre}$ are (preemptable) fallback rectangles.

We make the following observations with respect to regular rectangles:

1. If a job has a 2Reg rectangle, then its 2RegA rectangle is the topmost rectangle as a 2RegA rectangle is the first rectangle on its level and FFDHMix completes scheduling a job once it has assigned one of the job’s rectangles to this position. The topmost rectangle of jobs that do not have a 2Reg rectangle may be assigned to either a $B_1$ or $B_2$ block.

2. Even if the topmost rectangle of a preemptable job is assigned to a block of $B_2$, one of its rectangles in $B_1$ may be regular.

3. A job can have at most one 1Reg rectangle.

4. A block can have at most one 2RegB rectangle and no regular rectangle whether 1Reg or 2Reg can be assigned to a block after a 2RegB rectangle is assigned to it.

A regular job is one that has at least one regular rectangle. For our proofs, we extend the notion of a fallback rectangle to include some of the rectangles of $B_2$ that correspond to preemptable regular jobs. More precisely, if $r$ is a 1Reg or 2RegB preemptable regular rectangle assigned to block $i$ of $B_1$, then the rectangles of $J(r)$ assigned to blocks of $B_2$ that are below block $i$ of $B_1$ are also fallback rectangles of $J(r)$. We use the notation $FB(s)$ to denote the fallback rectangles of job $s$. Note that some of the rectangles in $FB(s)$ may be in $B_2$.

Some additional notations are illustrated below using Figure 3, which is similar to a figure in [8]. The indexes used in these notations correspond to those in the figure.

- $H_i$: Height of block $i$ of $B_1$.
- $f_i$: First rectangle in block $i$ of $B_1$. Note that $f_i$ is a regular rectangle and $H_i = h(f_i)$.
- $f_{ir}$: Remaining rectangles of $J(f_i)$. Some of these remaining rectangles of $J(f_i)$ may be in $B_1$ and the others in $B_2$. If $J(f_i)$ is non-preemptable, $f_{ir}$ is empty. In Figure 3, $f_i$ is a 2RegA rectangle. The 2RegB and a fallback rectangle of $J(f_i)$ are placed in blocks $i-1$ and $j$ of $B_1$, respectively.
- $w(f_{ir})$: The width $w(f_{ir})$ of $f_{ir}$ is the sum of the widths of the remaining rectangles of $J(f_i)$ and $w(J(f_i)) = w(f_i + f_{ir}) = w(f_i) + w(f_{ir})$.
- $R_i$: Regular rectangles in block $i$ of $B_1$. In Figure 3, the rectangles $f_i$ through $x_i$ are the regular rectangles of block $i$. 

6
$R_{i}^{non}$ Non-preemptable regular rectangles in block $i$ of $B1$. In Figure 3, $R_{i}^{non}$ is comprised of all shown rectangles other than the shaded one.

$R_{i}^{pre}$ Preamptable regular rectangles in block $i$ of $B1$. In Figure 3, the shaded rectangle in block $i$ is the only rectangle in $R_{i}^{pre}$. Note that $R_{i} = R_{i}^{non} ∪ R_{i}^{pre}$ and $R_{i}^{non} ∩ R_{i}^{pre} = \emptyset$.

$FPR_{i}$ Fallback preemptable rectangles of all jobs corresponding to 1Reg and 2RegB rectangles in block $i$ of $B1$. The rightmost shaded rectangle of block $j$ of Figure 3 is one of the elements of $FPR_{i}$.

$FPR_{ij}$ Subset of $FPR_{i}$ assigned to block $j$ of $B1$.

$FP_{ij}$ Fallback preemptable rectangles that are assigned to block $j$ of $B1$ after the last regular job assigned to block $i$ was scheduled before the scheduling of $J(f_{i})$ commenced. The leftmost shaded rectangle of block $j$ of Figure 3 is the only element of $FP_{ij}$. Note that its height must be more than that of every rectangle in block $i$.

$F_{ij}$ Non-preemptable fallback rectangles assigned to block $j$ of $B1$ after the last regular rectangle was assigned to block $i$ but before $f_{i}$ was assigned to block $i$. The fifth unshaded rectangle in block $i$ of Figure 3 is the only rectangle in this set.

$S_{ij}$ Non-preemptable fallback rectangles assigned to block $j$ of $B1$ after $f_{i}$ was assigned to block $i$ but before the assignment of the last regular rectangle in block $i$. The last unshaded rectangle in block $j$ of Figure 3 is the only rectangle in this set.

$c_{ij}$ Unused space in block $j$ of $B1$ just after the last regular rectangle was assigned to block $i$, $j < i$. Note that all of this unused space is at the right end of block $j$.

$c_{i}$ max$_{j < i}{c_{ij}}$.

**Observation 1** $FP_{i,i-1}$ and $F_{i,i-1}$ are empty, $i > 1$.

### 2.3 Supporting Lemmas

To establish the asymptotic performance bound of $1.7OPT(L) + 1$ for $FFDHMix$, we use the same weighting function as used in [8]. This function is given below:

$$W(x) = \begin{cases} 
6x/5 & 0 \leq x \leq 1/6 \\
9x/5 - 1/10 & 1/6 < x \leq 1/3 \\
6x/5 + 1/10 & 1/3 < x \leq 1/2 \\
6x/5 + 4/10 & 1/2 < x \leq 1 
\end{cases}$$

**Observation 2** $W(x) \geq 6/5w(x), 0 \leq x \leq 1$.

In the sequel, we abbreviate $W(w(r))$ by $W(r)$, where $r$ is a rectangle and $w(r)$ is its width.

**Lemma 1** Let $i$ and $j$, $j < i$ be two blocks of $B1$.

(a) $w(f_{i} + f_{i'}) = w(J(f_{i})) > c_{ij} + \sum_{r \in S_{ij} ∪ FPR_{ij}} w(r)$

(b) When $f_{i}$ is non-preemptable, $w(f_{i}) > c_{i}$.

**Proof** For (a), let $s$, $0 < s \leq 1$, be such that just following the scheduling of $J(f_{i}) - 1$ by $FFDHMix$, block $j$ has rectangles assigned in the interval $[0,s]$ and has no assigned rectangle in the interval $[s,1]$. Just after the last regular rectangle is assigned to block $i$, the empty space in block $j$ has shrunk from $[s,1]$ to $[1 - c_{ij},1]$. The rectangles assigned to block $j$ in the interval $[s,1 - c_{ij}]$, which is of width $1 - c_{ij} - s$, include those in $S_{ij} ∪ FPR_{ij}$ plus possibly one other rectangle, which is either a fallback rectangle or the corresponding $2$RegB
rectangle of \( J(f_i) \) in case \( f_i \) is a 2RegA rectangle. So, \( 1 - c_{ij} - s \geq \sum_{r \in S_{ij} \cup FPR_{ij}} w(r) \). Hence, \( 1 - s \geq c_{ij} + \sum_{r \in S_{ij} \cup FPR_{ij}} w(r) \).

Since a portion or all of \( J(f_i) \) has been assigned to block \( i \) of \( B_1 \), the width of \( J(f_i) \) must be more than the empty space available in any lower block just prior to the start of the scheduling of \( J(f_i) \) by FFDHMix. So, \( w(J(f_1)) > 1 - s \geq c_{ij} + \sum_{r \in S_{ij} \cup FPR_{ij}} w(r) \).

For part (b), we observe that when \( f_i \) is non-preemptable, \( w(f_i) = w(J(f_i)) \). Now, from part (a), we obtain \( w(f_i) > c_{ij}, j < i \). Hence, \( w(f_i) > \max_{j < i} \{c_{ij}\} = c_i \).

**Lemma 2** Let \( i \) and \( j \), \( j \leq i - 1 \) be two blocks of \( B_1 \).

\[
c_{ij} = c_{i-1,j} - \sum_{r \in S_{ij} \cup F_{P_{ij}} \cup FPR_{ij}} w(r)
\]  

(1)

**Proof** Each of the rectangles assigned to block \( j \) following the scheduling of the last regular job of block \( i - 1 \) and up to the completion of the scheduling of the last regular job of block \( i \) falls into exactly one of the following categories:

1. Fallback non-preemptable rectangles assigned to block \( j \) after the scheduling of the last regular job in \( i - 1 \) was scheduled but before the assignment of \( f_i \) to block \( i \) (\( F_{ij} \)).
2. Fallback preemptable rectangles assigned to block \( j \) after the last regular job of \( i - 1 \) was scheduled and before the scheduling of \( J(f_i) \) began (\( FP_{ij} \)).
3. Fallback non-preemptable rectangles assigned to block \( j \) after \( f_i \) is assigned to block \( i \) but before the assignment of the last regular rectangle in block \( i \) \( (S_{ij}) \).

4. Fallback preemptable rectangles of jobs corresponding to 1Reg and 2RegB rectangles of block \( i \) \( (FPR_{ij}) \).

Notice that when \( f_i \) is a 2RegA rectangle, its corresponding 2RegB rectangle is a regular rectangle of block \( i - 1 \) and so the rectangle, if any, of \( J(f_i) \) assigned to block \( j \) has been accounted for in \( c_{i-1,j} \).

**Lemma 3** Let \( r \) be a preemptable regular rectangle, other than a 2RegA rectangle, assigned to block \( i - 1, 1 < i \leq t \) of \( B_1 \).

\[
\sum_{s \in FB(J(r))} w(s) \geq c_{i-1}
\]

**Proof** Since \( c_i = 0 \), the lemma is trivially true for \( i = 2 \). Assume the \( i > 2 \). We first note that some of the rectangles in \( FB(J(r)) \) may be assigned to blocks of \( B_2 \). Regardless, before \( FFDHMix \) assigns a portion of \( J(r) \) to block \( i - 1 \), it necessarily assigns at least \( c_{i-1,j}, 1 \leq j < i - 1 \), of \( J(r) \) to blocks below \( i - 1 \) (whether in \( B_1 \) or \( B_2 \)) (if not, an additional portion of \( J(r) \) can be reassigned from block \( i - 1 \) to block \( j \) without violating the stacking constraint, which is not possible because of the way \( FFDHMix \) works). So, \( \sum_{s \in FB(J(r))} w(s) \geq c_{i-1} \). □

**Lemma 4** Let \( r \) be a 2RegB rectangle in block \( i - 1, 1 < i \leq t \) of \( B_1 \).

\[
\sum_{s \in FB(J(r))} w(s) \geq c_i
\]

**Proof** First consider the case \( c_i = c_{i,i-1} \). From the definition of \( c_{i,i-1} \), it follows that \( c_{i,i-1} \) is less than or equal to the space to the right of the 2RegB rectangle. Further, from the working of \( FFDHMix \), it follows that the fallback rectangles of \( J(r) \) must cover the space to the right of 2RegB (though on blocks of \( B_1 \) and \( B_2 \) below \( i - 1 \)) in order for \( r \) to be assigned to its position in block \( i - 1 \). So, \( \sum_{s \in FB(J(r))} w(s) \geq c_{i,i-1} = c_i \).

Next consider the case \( c_i = c_{ij} \) for some \( j < i - 1 \). From Lemma 2, we obtain \( c_{ij} \leq c_{i-1,j} \).

Using this in the result of Lemma 3, we get

\[
\sum_{s \in FB(J(r))} w(s) \geq c_{i-1} = \max_{q<i-1} \{c_{i-1,q}\} \geq c_{i-1,j} \geq c_{ij} = c_i
\]

□

**Lemma 5** \cite{Garey et al. \[7\]} If \( B \) is a set of one or more numbers \( x \) satisfying \( c < x \leq 1 \) and \( \sum_{x \in B} W(x) < 1 \), then either \( |B| = 1 \) and the single element \( x \in B \) satisfies \( x \leq 1/2 \), or else

\[
\sum_{x \in B} W(x) \geq 6/5 \sum_{x \in B} x + 6/5c - 1/5
\]

**Lemma 6** For every block \( j, 1 < j \leq t \), of \( B_1 \).

(a) When \( f_j \) is assigned to block \( j \), the blocks of \( B_1 \) below block \( j \) are more than half full (note that prior to this assignment, portions of \( J(f_j) \) may have been assigned to \( B_1 \) and \( B_2 \) blocks below block \( j \) of \( B_1 \)).

(b) \( c_j < 1/2 \)
Proof Suppose that the first part of the lemma is not true. Then, there is a least \( j \), \( j > 1 \) such that when \( f_j \) is assigned to block \( j \) one or more lower blocks of \( B_1 \) are at most half full. Let \( k \) be this least \( j \). So, when \( f_{k-1} \) was assigned to block \( k-1 \) of \( B_1 \) blocks \( 1 \cdots k-2 \) were more than half full. Hence, just before the scheduling of \( J(f_k) \) began, blocks \( 1 \cdots k-2 \) of \( B_1 \) were more than half full. So, block \( k-1 \) must be the only block of \( B_1 \) that is less than or equal to half full just before the scheduling of \( J(f_k) \) began. From this and the observation that all blocks in \( B_2 \) have a first rectangle that is non-preemptable and of width \( > 1/2 \), it follows that all blocks, other than \( k-1 \), below \( k \), whether in \( B_1 \) or \( B_2 \) are more than half full when the scheduling of \( J(f_k) \) began. Hence, there is an \( \epsilon > 0 \) such that all blocks (whether in \( B_1 \) or \( B_2 \)) other block \( k-1 \) of \( B_1 \) are full from 0 to \( 1/2 + \epsilon \) at the time \( FFDMix \) begins to schedule \( J(f_k) \).

If \( f_k \) is a non-preemptable rectangle, then its width must be \( > 1/2 \) as otherwise, it would have been assigned to block \( k-1 \) (recall that block \( k-1 \) is at most half full when \( f_k \) is assigned). But, blocks whose first rectangle is non-preemptable and of width \( > 1/2 \) are in \( B_2 \). So, \( f_k \) must be a preemptable rectangle. Suppose that block \( k-1 \) has rectangles assigned from 0 to \( q \) and is empty from \( q \) to 1, \( q \leq 1/2 \) at the time the scheduling of \( J(f_k) \) begins. The interval \( [q, 1/2 + \epsilon] \) of \( k-1 \) is free at this time. This interval is not free on any of the blocks below \( k-1 \) (including those of \( B_2 \)) as these blocks are full up to at least \( 1/2 + \epsilon \). So, \( FFDMix \) will assign a portion of \( J(f_k) \) to fill the interval \( [q, 1/2 + \epsilon] \) of \( k-1 \) before assigning any of \( J(f_k) \) to \( k \). So, just before \( f_k \) is assigned to block \( k \) of \( B_1 \), block \( k-1 \) will be more than half full. This contradicts the earlier claim that \( k-1 \) is at most half full when \( f_k \) is assigned to block \( k \) of \( B_1 \).

From the first part of this lemma, we obtain \( c_{jk} < 1/2, k < j \). So, \( c_j = \max_{k < j} \{c_{jk}\} < 1/2 \)

\[
J(f_k) \preceq J(f_{k-1}) \quad (1)
\]

Lemma 7 Let \( i, 2 \leq i \leq t \), be a block of \( B_1 \).

(a) \( \sum_{r \in R_{i-1}} w(r) + \sum_{s \in FPR_{i-1}} w(r) + c_{i-1} = 1. \)

(b) \( \sum_{r \in R_{i-1}^{\text{non}}} W(r) + 6/5 \sum_{r \in R_{i-1}^{\text{pre}}} w(r) + \sum_{s \in S_{i-1} \cup FPR_{i-1}} w(r) + c_{i-1} \geq 6/5 \)

Proof (a) follows from the observation that the left hand side includes all rectangles assigned to block \( i-1 \) plus the empty space in this block. (b) follows from Observation 2 and (a).

Lemma 8 For each block \( i \) of \( B_1, 2 \leq i \leq t \), that block \( i-1 \) has no \( 2\text{Reg} \) rectangle, there is an \( i' \), \( 0 \leq i' < i \), for which

\[
\sum_{r \in R_{i-1}^{\text{non}}} W(r) + 6/5 \sum_{r \in R_{i-1}^{\text{pre}}} w(r) \geq 1 + 6/5 c_{i-1} \tag{2}
\]

Proof

Case 1: \( \sum_{r \in R_{i-1}^{\text{non}}} W(r) + 6/5 \sum_{r \in R_{i-1}^{\text{pre}}} w(r) \geq 1. \)

If \( c_i \geq c_{i-1} \), then for some \( i' \), \( c_i = c_{i'} \geq c_{i-1} \). Hence,

\[
\text{LHS of Equation 2} \geq 1 + 6/5 c_{i'} \geq 1 + 6/5 c_{i-1}
\]

If \( c_i < c_{i-1} \), then \( c_{i-1} > 0 \) and hence \( i-1 \geq 2 \) (note that \( c_1 = 0 \)). So, for some \( i' < i-1 < i \), \( c_{i-1} = c_{i-1,i'} \). From Lemma 2, it follows that

\[
\sum_{r \in S_{i-1} \cup FPR_{i-1}} w(r) = c_{i-1,i'} - c_{i'}
\]
Hence,

\[
LHS \text{ of Equation } 2 \geq 1 + 6/5(c_{i-1,i'} - c_{i'i'}) + 6/5c_{i'i'} = 1 + 6/5c_{i-1}
\]

**Case 2:** \(\sum_{r \in R_{i-1}^{\text{pre}}} W(r) + 6/5 \sum_{r \in R_{i-1}^{\text{pre}}} w(r) < 1\) We consider two subcases (a) \(R_{i-1}^{\text{pre}} = \emptyset\) and (b) \(R_{i-1}^{\text{pre}} \neq \emptyset\).

**Case 2a:** \(R_{i-1}^{\text{pre}} = \emptyset\) We first show that block \(i - 1\) has at least two (non-preemptable) regular rectangles. Recall that the first rectangle in every block of \(B1\) is regular. Suppose that \(f_{i-1}\) is the only regular rectangle in block \(i - 1\). Then, immediately after the scheduling of job \(J(f_i) - 1\) by \(FFDHMix\) is complete and before the scheduling of \(J(f_i)\) begins, \(f_{i-1}\) is the only rectangle assigned to block \(i - 1\) as the presence of another rectangle \(r\) in this block would imply that \(r\) is regular, which contradicts the assumption that block \(i - 1\) has only regular rectangle. Since \(i - 1\) is a block of \(B1\) and \(f_{i-1}\) is a non-preemptable regular rectangle, \(w(f_{i-1}) \leq 1/2\) (otherwise, \(i - 1\) would be a block of \(B2\)). From Lemma 6, it follows that when \(f_i\) is assigned to block \(i - 1\), block \(i - 1\) is more than half full. So, some portion of \(J(f_i)\) must have been assigned to block \(i - 1\) during the scheduling of \(J(f_i)\). This portion is, by definition, a 2RegB regular rectangle (and \(f_i\) is a 2RegA rectangle), which contradicts the assumptions that block \(i - 1\) has only regular rectangle and has no preemptable regular rectangle. So, we conclude that block \(i - 1\) has at least two (non-preemptable) regular rectangles.

We next observe that the width of each of the regular rectangles in block \(i - 1\) is more than \(c_{i-1}\). To see this, note that since the regular rectangles in block \(i - 1\) are non-preemptable they cannot have a stacking constraint violation with any previously assigned rectangle in any block (whether in \(B1\) or \(B2\)). Further, if the width of any one of these regular rectangles is \(\leq c_{i-1,j}\) for any \(j < i - 1\), then \(FFDHMix\) would have assigned this rectangle to one such lower block \(j\) and not to \(i - 1\). Also, since \(R_{i-1}^{\text{pre}} = \emptyset\), the condition for Case 2 implies that \(\sum_{R_{i-1}^{\text{non}}} W(r) < 1\). Lemma 5, now yields:

\[
\sum_{r \in R_{i-1}^{\text{non}}} W(r) \geq 6/5 \sum_{r \in R_{i-1}^{\text{non}}} w(r) + 6/5c_{i-1} - 1/5
\]

Substituting into the left hand side of Equation 2 and setting \(i' = i - 1\), we get

\[
LHS \geq 6/5 \sum_{r \in R_{i-1}^{\text{pre}}} w(r) + \sum_{r \in S_{i'} \cup FPR_{i'}} w(r) + c_{i'i'} + 6/5c_{i-1} - 1/5
\]

\[
= 6/5 + 6/5c_{i-1} - 1/5 \quad (R_{i-1}^{\text{pre}} = \emptyset \text{ and Lemma 7 (a)})
\]

\[
= 1 + 6/5c_{i-1}
\]

\[
= RHS
\]

**Case 2b:** \(R_{i-1}^{\text{pre}} \neq \emptyset\) From the lemma statement, it follows that block \(i - 1\) has at least one preemptable regular rectangle \(s\) that is not 2Reg (and hence not 2RegA). So, Lemma 3 applies and \(\sum_{q \in FB(J(s))} w(q) \geq c_{i-1}\). Observe also that \(FB(J(s)) \subseteq FPR_{i-1}\). Therefore,

\[
\sum_{r \in FPR_{i-1}} w(r) \geq c_{i-1}
\]

(3)

Setting \(i' = i - 1\) in the left hand side of Equation 2, we obtain
\[ \text{LHS} \geq \frac{6}{5} \left[ \sum_{r \in \mathcal{R}_{i-1}^{\text{non}}} w(r) + \sum_{r \in \mathcal{R}_{i-1}^{\text{pre}}} w(r) + \sum_{r \in \{S_{i,i-1} \cup F_{i,i-1} \cup FP_{i,i-1} \}} w(r) + c_{i,i-1} \right] + \frac{6}{5} \sum_{r \in \{FPR_{i-1,i-1} \}} w(r) \]

Observation 2

\[ = \frac{6}{5} + \frac{6}{5} \sum_{r \in \mathcal{R}_{i-1}^{\text{pre}}} w(r) \text{ (Lemma 7(a) and } FPR_{i-1,i-1} = \emptyset) \]

\[ \geq \frac{6}{5} + \frac{6}{5} c_{i,i-1} \text{ (using Equation 3)} \]

\[ > \text{RHS} \]

Let \( I \) be the subset of \( B1 \cup \{t+1\} \) such that \( i \in I \) iff block \( i - 1 \) of \( B1 \) has a 2Reg rectangle. \( I \) may be decomposed into disjoint sets of contiguous blocks \( a, a + 1, \ldots, b \) of \( B \cup \{t + 1\} \), called 2Reg sets, such that the only 2Reg rectangle in block \( a - 1 \) is 2RegB, blocks \( a \) through \( b - 2 \) have exactly one 2RegA and one 2RegB rectangle, and the only 2Reg rectangle in block \( b - 1 \) is 2RegA (Figure 4). Note that \( b > a \) as the blocks \( a - 1, b - 1 \) have different composition. So, a 2Reg set always has two or more blocks.

![Figure 4: Blocks a − 1, a, · · ·, b − 1 corresponding to a 2RegSet. Note that the composition of block b is inconsequential and is not shown here.](image)

12
Lemma 9 \ Let R = a, a + 1, \ldots, b be a 2Reg set.
\[
\sum_{r \in \text{Reg}_n} W(r) + \frac{6}{5} \sum_{r \in \text{Reg}_n \cup \text{FPR}_{a-1}} w(r) + \frac{6}{5} \sum_{r \in S_{a-1} \cup \text{FPR}_{a-1}} w(r) + \frac{6}{5}c_{a,a-1} \\
\geq 1 + \frac{6}{5}c_{a-1} + \frac{3}{5}c_a
\]
(4)

Proof \ Let q be the 2RegB rectangle in block \(a - 1\). From Lemmas 3 and 4 and \(FB(J(q)) \subseteq \text{FPR}_{a-1}\), we obtain
\[
\frac{6}{5} \sum_{r \in \text{FPR}_{a-1}} w(r) \geq 3\frac{5}{c_{a-1}} + \frac{3}{5}c_a
\]
(5)

When \(f_{a-1}\) is a preemptable rectangle, block \(a - 1\) has at least 2 preemptable rectangles (\(f_{a-1}\) and the 2RegB rectangle). Applying Lemma 3 to \(f_{a-1}\) (note that \(f_{a-1}\) cannot be 2RegA as block \(a - 1\) has no 2RegA rectangle), Lemma 4 to the 2RegB rectangle of block \(a - 1\), and noting that both the fallback rectangles of both these regular preemptable rectangles are contained in \(\text{FPR}_{a-1}\), we obtain the stronger relation
\[
\frac{6}{5} \sum_{r \in \text{FPR}_{a-1}} w(r) \geq 6\frac{5}{c_{a-1}} + \frac{6}{5}c_a
\]
(6)

Equation 4 now follows from Equation 6 and Lemma 7(b).

When \(f_{a-1}\) is a non-preemptable rectangle, Equation 4 is established considering two cases as below.

Case 1: \(0 \leq c_{a-1} \leq 1/3\)

From Equation 5 and Lemma 7(b), we get

\[
\text{LHS of Equation 4} \geq 6 + 3\frac{5}{c_{a-1}} + \frac{3}{5}c_a \\
= 1 + 1/5 + 3/5c_{a-1} + 3/5c_a \\
\geq 1 + 3/5c_{a-1} + 3/5c_{a-1} + 3/5c_a \\
= 1 + 6/5c_{a-1} + 3/5c_a
\]

Case 2: \(1/3 < c_{a-1} < 1/2\)

Since \(f_{a-1}\) is non-preemptable, \(w(f_{a-1}) = w(J(f_{a-1}))\). From Lemma 1(b) and \(c_{a-1} \geq 1/3\), it follows that \(w(f_{a-1}) > 1/3\). Hence, \(W(f_{a-1}) = 6/5w(f_{a-1}) + 1/10\). From this and Observation 2, we get

\[
\text{LHS of Equation 4} \geq 6/5 \sum_{r \in \text{Reg}_n} w(r) + 1/10 + 6/5 \sum_{r \in \text{Reg}_n \cup \text{FPR}_{a-1}} w(r) + \frac{6}{5} \sum_{r \in S_{a-1} \cup \text{FPR}_{a-1}} w(r) + \frac{6}{5}c_{a,a-1} \\
\geq 6/5 \sum_{r \in \text{Reg}_n} w(r) + \frac{6}{5} \sum_{r \in \text{FPR}_{a-1}} w(r) + \frac{6}{5} \sum_{r \in S_{a-1} \cup \text{FPR}_{a-1}} w(r) + c_{a,a-1} \\
\geq 6/5 + 1/10 + 6/5 \sum_{r \in \text{FPR}_{a-1}} w(r) \\
\geq 6/5 + 1/10 + 3/5c_{a-1} + 3/5c_a (\text{Lemma 7(a) and Equation 5}) \\
= 1 + 3/10 + 3/5c_{a-1} + 3/5c_a \\
> 1 + 3/5c_{a-1} + 3/5c_{a-1} + 3/5c_a \\
= 1 + 6/5c_{a-1} + 3/5c_a
\]
From Lemma 6(b), we know that \(c_{j-1} < 1/2\). So, there are no more cases to consider and Equation 4 is established.

**Lemma 10** Let \(R = a, a + 1, \ldots, b\) be a 2Reg set that has more than 2 blocks (i.e., \(b > a + 1\)). Let \(j\) be such that \(a < j < b\).

\[
\sum_{r \in R_{j-1}^{\text{non}}} W(r) + \frac{6}{5} \sum_{r \in R_{j-1}^{\text{pre}} \cup FPR_{j-1}} w(r) + \frac{6}{5} \sum_{r \in S_{j-1} \cup FPR_{j-1}} w(r) + \frac{6}{5} c_{j-1} \geq 1 + 3/5 c_{j-1} + 4/5 c_j
\]

(7)

**Proof** Since \(R\) is a 2Reg set, block \(j-1\) has both a 2RegA and a 2RegB rectangle. Let \(q\) be the 2RegB rectangle in block \(j-1\). From Lemmas 3 and 4 and \(FB(J(q)) \subseteq FPR_{j-1}\), we obtain

\[
\frac{6}{5} \sum_{r \in FPR_{j-1}} w(r) = 3/5 c_{j-1} + 3/5 c_j
\]

(8)

From Equation 8 and Lemma 7(b), we get

\[
\text{LHS of Equation 7} \geq \frac{6}{5} + 3/5 c_{j-1} + 3/5 c_j = 1 + 1/5 + 3/5 c_{j-1} + 3/5 c_j > 1 + 3/5 c_{j-1} + 4/5 c_j
\]

(9)

**Lemma 11** Let \(R = a, a + 1, \ldots, b\) be a 2Reg set.

\[
\sum_{r \in R_{b-1}^{\text{non}}} W(r) + \frac{6}{5} \sum_{r \in R_{b-1}^{\text{pre}} \cup FPR_{b-1}} w(r) + \frac{6}{5} \sum_{r \in S_{b-1} \cup FPR_{b-1}} w(r) + \frac{6}{5} c_{b-1} \geq \sum_{r \in R_{b-1}^{\text{non}}} W(r) + \frac{6}{5} \sum_{r \in R_{b-1}^{\text{pre}} \cup FPR_{b-1}} w(r) + \frac{6}{5} \sum_{r \in S_{b-1} \cup FPR_{b-1}} w(r) + \frac{6}{5} c_{b-1}
\]

(10)

**Proof** First, we observe that Lemma 7(b) holds even when \(b = t + 1\) (note that \(S_{t+1, t} = FPR_{t+1, t} = \emptyset\) and \(c_{t+1, t}\) is the empty space in block \(t\) when FFDMix terminates). From Lemmas 7(b) and 6(b), we obtain

\[
\text{LHS of Equation 9} \geq \sum_{r \in R_{b-1}^{\text{non}}} W(r) + \frac{6}{5} \sum_{r \in R_{b-1}^{\text{pre}} \cup FPR_{b-1}} w(r) + \frac{6}{5} \sum_{r \in S_{b-1} \cup FPR_{b-1}} w(r) + \frac{6}{5} c_{b-1} \geq 6/5 \text{ (Lemma 7(b))} = 1 + 1/5 > 1 + 2/5 c_{b-1} \text{ (Lemma 6(b))}
\]

(11)

**Lemma 12** Let \(R = a, a + 1, \ldots, b\) be a 2Reg set that has more than 2 blocks (i.e., \(b > a + 1\)).

\[
\sum_{i=a}^{b} H_i \left( \sum_{r \in R_{i-1}^{\text{non}}} W(r) + \frac{6}{5} \sum_{r \in R_{i-1}^{\text{pre}} \cup FPR_{i-1}} w(r) + \frac{6}{5} \sum_{r \in S_{i-1} \cup FPR_{i-1}} w(r) + \frac{6}{5} c_{i-1} \right) \geq \sum_{i=a}^{b} H_i \left[ \frac{6}{5} + \frac{6}{5} \right]
\]

(10)
Proof. Since $R$ has more than two blocks, it has at least one block to which Equation 7 applies. Multiplying both sides of Equations 4, 7, and 9 by the height of the block and summing over all blocks of $R$, we get

$$
\sum_{i=a}^{b} H_i \left[ \sum_{r \in R_{i-1}^{\text{non}}} W(r) + 6/5 \sum_{r \in R_{i-1}^{\text{pre}}} w(r) + 6/5 \sum_{r \in S_{i-1}^{\text{pre}}} w(r) + 6/5c_{i-1} \right] \\
\geq H_a \left[ 1 + 6/5c_{a-1} + 3/5c_a \right] + \sum_{i=a+1}^{b-1} H_i \left[ 1 + 3/5c_{i-1} + 4/5c_i \right] + H_b \left[ 1 + 2/5c_{b-1} \right] \\
\geq H_a \left[ 1 + 6/5c_{a-1} \right] + \sum_{i=a+1}^{b-1} H_i \left[ 1 + 6/5c_{i-1} \right] + 4/5H_{b-1}c_{b-1} + H_b + 2/5H_{b-1}c_{b-1} \\
\quad \text{(since, $H_{i-1} \geq H_i$ for $1 < i \leq t$)} \\
\geq H_a \left[ 1 + 6/5c_{a-1} \right] + \sum_{i=a+1}^{b-1} H_i \left[ 1 + 6/5c_{i-1} \right] + H_b + 6/5H_{b-1}c_{b-1} \quad \text{(as $H_{b-1} \geq H_b$)} \\
= \sum_{i=a}^{b} H_i \left[ 1 + 6/5c_{i-1} \right]
$$

Lemma 13. Equation 10 holds even when $R$ has exactly two blocks.

Proof. When $R$ has only two blocks (i.e., $b = a + 1$), the proof of Lemma 12 does not work as we cannot employ Equation 7. We consider two cases depending on whether or not block $a = b - 1$ has at least 2 regular rectangles.

Case 1: Block $a = b - 1$ has at least 2 regular rectangles. $f_{b-1}$, which is $2\text{RegA}$, is one of the 2 regular rectangles in $b - 1$. Let $s$ be one of the other regular rectangles in this block. If $s$ is preemptable, then from Lemma 3 and $FB(J(s)) \subseteq FPR_{b-1}$, we obtain

$$
\sum_{r \in FPR_{b-1}} w(r) \geq c_{b-1}
$$

From this and Lemma 7(b), we get

$$
\sum_{r \in R_{b-1}^{\text{non}}} W(r) + 6/5 \sum_{r \in R_{b-1}^{\text{pre}}} w(r) + 6/5 \sum_{r \in S_{b-1}^{\text{pre}}} w(r) + 6/5c_{b-1} > 1 + 6/5c_{b-1} \\
\quad \text{(11)}
$$

Multiplying both sides of Equation 4 by $H_a$ and Equation 11 by $H_b$ and adding, we establish the lemma for Case A when $s$ is preemptable.

When $s$ is not preemptable, we first prove that

$$
\sum_{r \in R_{b-1}^{\text{non}}} W(r) + 6/5 \sum_{r \in R_{b-1}^{\text{pre}}} w(r) + 6/5 \sum_{r \in S_{b-1}^{\text{pre}}} w(r) + 6/5c_{b-1} \geq 1 + 3/5c_{b-1} \\
\quad \text{(12)}
$$

To establish this equation, we consider two cases depending on the value of $c_{b-1}$.

Case 1a: $0 \leq c_{b-1} \leq 1/3$.

From Lemma 7(b), we see that the LHS of Equation 12 is $\geq 6/5 = 1 + 1/5 \geq 1 + 3/5c_{b-1}$. 

15
Case 1b: $1/3 < c_{b-1} < 1/2$.
Since $s$ is non-preemptable, $w(s) = w(J(s))$. From Lemma 1(b) and $c_{b-1} > 1/3$, it follows that $w(s) > 1/3$. Hence, $W(s) = 6/5w(s) + 1/10$. From this, Observation 2, and Lemma 7(b), we see that the LHS of Equation 12 is $\geq 6/5 + 1/10 = 1 + 3/10 > 1 + 3/5c_{b-1}$ (as $c_{b-1} < 1/2$).

From Lemma 6(b), we know that $c_{b-1} < 1/2$. So, there are no other cases to consider and Equation 12 is proved. Multiplying both sides of Equation 4 by $H_a$ and of Equation 12 by $H_b$, adding the two equations, and using $H_a \geq H_b$, establishes the lemma for Case 1 when $s$ is non-preemptable.

Case 2: Block $a = b - 1$ has only 1 regular rectangle.
Case 2a: Block $a - 1$ has a regular preemptable rectangle other than its $2\text{RegB}$ rectangle.
Now, Equation 6 holds and so,\[\sum_{r \in R_{a-1}^{\text{non}}} W(r) + \frac{6}{5} \sum_{r \in R_{a-1}^{\text{pre}} \cup FPR_{a-1}} w(r) + \frac{6}{5} \sum_{r \in S_{a-1} \cup FPR_{a-1}} w(r) + \frac{6}{5}c_{a-1} \geq 1 + \frac{6}{5}c_{a-1} + \frac{6}{5}c_a\] (13)

Multiplying both sides of Equation 13 by $H_a$ and of Equation 9 by $H_b$, adding the two equations, and using $a = b - 1$ and $H_a \geq H_b$, establishes the lemma for Case 2a.

Case 2b: Block $a - 1$ has no regular preemptable rectangle other than its $2\text{RegB}$ rectangle.
Case 2b(i): $b = t + 1$.
Now, $H_b = 0$ and the non-zero terms in Equation 10 with $I = \{a, b\}$ correspond to $i = a$. Multiplying Equation 4 by $H_{a-1}$, we see that the LHS of Equation 10 is $\geq H_{a-1}[1 + 6/5c_{a-1} + 3/5c_a] \geq H_{a-1}[1 + 6/5c_{a-1}]$, which equals the RHS of Equation 10.

Case 2b(ii): $b < t + 1$ and $\sum_{r \in R_{a-1}^{\text{non}}} W(r) \geq 1$.
Since block $a - 1$ has a $2\text{RegB}$ rectangle, Lemma 4 and $a = b - 1$ give us \[\frac{6}{5} \sum_{r \in FPR_{a-1}} w(r) \geq \frac{6}{5}c_a = \frac{6}{5}c_{a-1}\] (14)

By definition, $c_{a,a-1}$ is the space left in block $a - 1$ after the last regular rectangle (note that $f_a$, which is $2\text{RegA}$, is the first, last, and only regular rectangle in block $a$) is assigned to block $a$ and $c_{a-1,j}, j < a - 1$ is the space left in block $j$ after the last regular rectangle (i.e., the $2\text{RegB}$ rectangle corresponding to $f_{a-1}$) is assigned to block $a - 1$. Suppose that $c_{a,a-1} < c_{a-1,k}$ for some $k, k < a - 1$. We see that there exists a $q, 1 - c_{a-1,k} \leq q \leq 1 - c_{a,a-1}$ such that a portion or all of the $2\text{RegB}$ rectangle of block $a - 1$ is assigned to the interval $[q, l - c_{a,a-1}]$ of block $a - 1$. Hence, because of the stacking constraint, $J(f_a)$ is not assigned in this interval to any other block (whether in $B1$ or $B2$). From the working of $FFDHMix$, however, the $[q, l - c_{a,a-1}]$ segment of the $2\text{RegB}$ rectangle should have been assigned to this (empty) interval of block $k$ (or a lower block) and not to this interval block $a - 1$. Therefore, there is no such $k$ and $c_{a,a-1} \geq \max_{j < a - 1}\{c_{a-1,j}\} = c_{a-1}$. From this, the Case 2b(ii) condition, and Equation 14, we obtain\[\sum_{r \in R_{a-1}^{\text{non}}} W(r) + \frac{6}{5} \sum_{r \in R_{a-1}^{\text{pre}} \cup FPR_{a-1}} w(r) + \frac{6}{5} \sum_{r \in S_{a-1} \cup FPR_{a-1}} w(r) + \frac{6}{5}c_{a-1} \geq \frac{1}{5} \sum_{r \in R_{a-1}^{\text{non}}} W(r) + \frac{6}{5} \sum_{r \in R_{a-1}^{\text{pre}} \cup FPR_{a-1}} w(r) + \frac{6}{5} \sum_{r \in S_{a-1} \cup FPR_{a-1}} w(r) + \frac{6}{5}c_{a-1} \geq 1 + \frac{6}{5}c_{a-1} + \frac{6}{5}c_a\] (15)

Multiplying both sides of Equation 17 by $H_a$ and both sides of Equation 11 by $H_b$, adding together, and using $H_a \geq H_b$, we obtain Equation 10.
Case 2b(iii): $b < t + 1$ and $\sum_{r \in R_{a-1}^{non}} W(r) < 1$.

From Lemma 6(a), at the time $f_0$ is assigned to block $b$, block $a = b - 1$ was more than half full. Note that no rectangle of $J(f_0)$ is assigned to block $a = b - 1$ as such an assignment would place a 2RegB rectangle in block $a$ and this block would then have at least 2 regular rectangles ($f_a$ and the 2RegB rectangle of $J(f_b)$), which contradicts the Case 2 assumption. Hence, at the time $f_0$ is assigned to block $b$ by $FFDHMix$, $f_a$ is the only rectangle in block $a$ (recall that every rectangle assigned to block $a$ is regular). So, $w(f_0) > 1/2$. Hence, the 2RegB rectangle of block $a - 1$, which also belongs to $J(f_0)$ is assigned to some interval $[u, v]$ of block $a - 1$ with $u > 1/2$. The rectangles occupying the interval $[0, u]$ of $a - 1$ are regular non-preemptable (the Case 2b assumption is that block $a - 1$ has no regular preemptable rectangle other than 2RegB). So, $f_{a-1}$ is non-preemptable and since $a - 1$ is a block of $B_1$, $w(f_{a-1}) \leq 1/2$. Hence, there must be at least one other non-preemptable regular rectangle between $f_{a-1}$ and 2RegB. The width of each of the 2 or more non-preemptable regular rectangles assigned to block $a - 1$ must exceed $c_{a-1}$ as otherwise, at least one of these would have been assigned to a block below $a - 1$ rather than to $a - 1$. Using these properties of $R_{a-1}^{non}$ together with the Case 2b(iii) property in Lemma 5, we obtain

$$\sum_{r \in R_{a-1}^{non}} W(r) \geq 6/5 \sum_{r \in R_{a-1}^{non}} w(r) + 6/5c_{a-1} - 1/5$$

So,

$$\sum_{r \in R_{a-1}^{non}} W(r) + 6/5 \sum_{r \in R_{a-1}^{non} \cup FPR_{a-1}} w(r) + 6/5 \sum_{r \in S_{a,a-1} \cup FPR_{a-1}} w(r) + 6/5c_{a,a-1}$$

$$\geq 6/5 \sum_{r \in R_{a-1}^{non}} w(r) + 6/5c_{a-1} - 1/5 + 6/5 \sum_{r \in R_{a-1}^{non} \cup FPR_{a-1}} w(r)$$

$$+ 6/5 \sum_{r \in S_{a,a-1} \cup FPR_{a-1}} w(r) + 6/5c_{a,a-1}$$

$$\geq 6/5 \sum_{r \in R_{a-1}} w(r) + \sum_{r \in S_{a,a-1} \cup FPR_{a-1}} w(r) + c_{a,a-1} + 6/5c_{a-1} - 1/5 + 6/5 \sum_{r \in FPR_{a-1}} w(r)$$

$$\geq 6/5 + 6/5c_{a-1} + 6/5c_{b-1} - 1/5 \text{ (Lemmas 7(a) and 4)}$$

$$= 1 + 6/5c_{a-1} + 6/5c_{b-1}$$

As in Case 2b(ii), this equation and Equation 11 lead to Equation 10.

**Lemma 14** Let $I$ be the subset of $B_1 \cup \{t+1\}$ such that $i \in I$ iff block $i - 1$ of $B_1$ has a 2Reg rectangle.

$$\sum_{i \in I} H_i \left[ \sum_{r \in R_{a-1}^{non}} W(r) + 6/5 \sum_{r \in R_{a-1}^{non} \cup FPR_{a-1}} w(r) + 6/5 \sum_{r \in S_{a,a-1} \cup FPR_{a-1}} w(r) + 6/5c_{a-1} - 1 \right] \geq \sum_{i \in I} H_i [1 + 6/5c_{i-1}]$$  

(18)

**Proof** Follows from Lemmas 12 and 13 and the fact that $I$ may be decomposed into disjoint sets of 2Reg sets.  


2.4 The Bound

**Lemma 15** Let \( Q \) be a set of preemptable and non-preemptable jobs. Let \( S^* \) be a minimum power schedule for \( Q \) and \( S \) the \( \text{FFDHMix} \) schedule. Let \( B_1 \) and \( B_2 \) be as in Section 2.2. Let \( T_2 \) be the set of first rectangles in \( B_2 \). Let \( T_1 \) be the remaining rectangle in \( S^* \).

\[
\sum_{r \in T_1} h(r)W(r) > H(B_1) - 1
\]

**Proof** For every block \( i, 2 \leq i \leq t, \) of \( B_1 \), let \( i' < i \) be some block of \( B_1 \) that is below \( i \). We claim that

\[
\sum_{r \in T_1} h(r)W(r) \geq \sum_{i=2}^{t} H_i \sum_{r \in R_{i-1}^{\text{non}}} W(r) + 6/5H_i \sum_{r \in R_{i-1}^{\text{pre}}} W(r) + 6/5(H_i - H_{i+1}) \sum_{r \in (J_i \cup f_{i'})} W(r) + 6/5H_{i+1} \sum_{r \in S_{i'} \cup \text{FP}_{i'}} W(r)
\]

**Observation 3** To see the validity of this claim, we make the following observations about its RHS.

1. The \( 7(t - 1) \) rectangle sets \( R_{i-1}^{\text{non}}, R_{i-1}^{\text{pre}}, \text{FP}_{i-1} - \text{FP}_{i-1,i'}, F_{i'}, \text{FP}_{i'} \), \( S_{i'} \), \( \text{FP}_{i'} \), \( 2 \leq i \leq t \) are disjoint.

2. Although the RHS may include some rectangles that are not in \( B_1 \), they include no rectangle of \( T_2 \)

3. Since the height of every rectangle in \( R_{i-1} \cup \text{FP}_{i-1} \cup F_{i'} \cup \text{FP}_{i'} \) is at least \( H_i \) and that of each rectangle in \( S_{i'} \cup \text{FP}_{i'} \) is at least \( H_{i+1} \) (note that \( H_{t+1} = 0 \)), the height multiplier of each rectangle in the sums within the square brackets, other than the third sum, is less than or equal to its actual height.

4. The height of a first rectangle \( f_i \) and that of the remaining rectangles \( f_{i'} \) of \( J(f_i) \) is \( H_i \). Also, \( f_i \in R_i \). So, \( f_i, 1 \leq i < t \), is accounted for in sum 1 or 2 but with a multiplier of \( H_{i+1} \) and \( f_i \) is not not in any of the sums. The multiplier \( H_i - H_{i+1} \) for the third sum corrects for the height shortfall in the first two sums for the first rectangles. When \( f_i \) is a preemptable \( 1\text{Reg} \) or \( 2\text{RegB} \) rectangle, some of the rectangles in \( f_{i'} \) are included in \( \text{FP}_i \). The rectangles of \( f_{i'} \) that are included in sums of the RHS other than the third have the height multiplier is \( H_{i+1} \). So, possibly repeating the rectangle in the third sum with a multiplier of \( H_i - H_{i+1} \) still results in no rectangle (even if it appears twice in the RHS) having an effective height multiplier greater than its true height.

Let \( J_1 \) be the jobs of \( Q \) corresponding to the rectangles in \( T_1 \). Some, though possibly not all, of the jobs in \( J_1 \) have representation in the RHS of Equation 19. For every job in \( J_1 \) the sum of the widths of its corresponding rectangles in \( T_1 \) equals the specified job duration while the sum of the corresponding rectangles for this job (which may be of different width from those in \( T_1 \)) in the RHS is less than or equal to the job’s specified duration. Also, the multiplier for the job’s rectangles in the LHS equals the job’s power requirement while that for its rectangles in the RHS does not exceed the power requirement (Observation 3).

\footnote{As noted in Observation 3.4, some rectangles appear twice in the RHS. When adding the widths of rectangles in the RHS, each rectangle is counted only once.}
Consider a job \( q \) for which there is at least one rectangle in the RHS. If \( q \) is non-preemptable regular, then it appears on the RHS once in the first sum as an \( H_i w(q) \) and possibly again in the third sum as \( 6/5(H_i - H_{i+1}) w(q) \), where \( w(q) \) is the duration of \( q \). Since \( W(q) \geq 6/5 w(q) \) (Observation 2), we see that the total contribution of \( q \) to the RHS is \( \leq h(q) W(q) \) while its contribution to the LHS is \( h(q) W(q) \). When \( q \) corresponds a non-preemptable fallback rectangle, its contribution to the LHS is \( h(q) W(q) \) while its contribution to the RHS is at most \( 6/5 h(q) w(q) \). Since, \( W(q) \geq 6/5 w(q) \), \( q \) again contributes at least as much to the LHS as it does to the RHS. When \( q \) is preemptable, it may appear as one or more rectangles in the LHS and RHS. The sum of the widths of \( q \)'s corresponding rectangles in the LHS is \( w(q) \) while in the RHS it is \( \leq w(q) \) (again, when a rectangle of \( q \) appears twice in the RHS, its width is counted only once when summing the widths of the rectangles of \( q \) that appear in the RHS).

Let the widths of the rectangles of \( q \) in \( S^* \) be \( x_1, \ldots, x_m \). The contribution of \( q \) to the LHS is \( h(q) \sum_{j=1}^{m} W(x_j) \geq 6/5 h(q) \sum_{j=1}^{m} x_j = 6/5 h(q) w(q) \), which is greater than or equal to the contribution of \( q \) to the RHS. Hence, Equation 19 is valid.

Next, we show that there is a selection of \( i', 2 \leq i \leq t \) values for which the RHS of Equation 19 is \( \geq H(B1) - 1 \) thereby proving the lemma.

Using Lemma 1, we obtain

\[
\sum_{r \in \{f_i \cup f_{i'} \}} w(r) = w(f_i + f_{i'}) > c_{ii'} + \sum_{r \in S_{ii'} \cup FP_{ii'}} w(r)
\]

Substituting into the RHS of Equation 19, we obtain

\[
\text{RHS of Equation 19} \geq \sum_{i=2}^{t} [H_i \sum_{r \in R_{i-1}^{reg}} W(r) + 6/5 H_i \sum_{r \in R_{i-1}^{pre} \cup FP_{i-1} \cup FP_{i-1, i'}} w(r)]
\]

\[
+ 6/5 H_i \sum_{r \in P_{ii'} \cup FP_{i'} \cup FP_{ii'}} w(r)
\]

\[
+ 6/5 H_{i+1} \sum_{r \in S_{ii'} \cup FP_{ii'}} w(r)
\]

\[
= \sum_{i=2}^{t} [H_i \sum_{r \in R_{i-1}^{reg}} W(r) + 6/5 H_i \sum_{r \in R_{i-1}^{pre} \cup FP_{i-1} \cup FP_{i-1, i'}} w(r)]
\]

\[
+ 6/5 H_i \sum_{r \in S_{ii'} \cup P_{ii'} \cup FP_{i'} \cup FP_{ii'}} w(r)
\]

\[
+ 6/5 (H_i - H_{i+1}) c_{ii'}
\]

(20)

For blocks \( i \geq 2 \) of \( B1 \) that have no 2Reg rectangle in block \( i - 1 \), we select \( i' \) as in Lemma 8 and for the remaining blocks, \( i' = i - 1 \). Using Equation 2 on the former set of blocks and Equation 18 on the latter set to substitute for terms on the RHS of Equation 20 and the
simplification steps of Coffman et al. [8], we get

\[
\text{RHS of Equation 19} > \sum_{i=2}^{t} H_i + \frac{6}{5} \sum_{i=2}^{t} H_i c_{i-1} - 6/5 \sum_{i=2}^{t} H_{i+1} c_{i'}
\]

\[
= \sum_{i=2}^{t} H_i + \frac{6}{5} \sum_{i=1}^{t-1} H_{i+1} c_i - 6/5 \sum_{i=2}^{t} H_{i+1} c_{i'}
\]

\[
= \sum_{i=2}^{t} H_i + \frac{6}{5} \sum_{i=2}^{t-1} H_{i+1} (c_i - c_{i'}) (c_1 = H_{t+1} = 0)
\]

\[= H(B1) - 1 \quad (H_1 = 1 \text{ and } c_i \geq c_{i'}) \]

Theorem 1 \(\text{FFDH} \text{Mix}(Q) < 1.7\text{OPT}(Q) + 1\), for every set \(Q\) comprised of preemptable and non-preemptable jobs.

Proof. \(S^*, T1, T2, B1, B2\) are as in Lemma 15. Let \(A(S^*)\) be the weighted area of the rectangles in \(S^*\). That is

\[
A(S^*) = \sum_{r \in S^*} h(r)W(r) = \sum_{r \in T1} h(r)W(r) + \sum_{r \in T2} h(r)W(r)
\]  

As in Coffman et al. [8], slicing \(S^*\) by “drawing a line through the top and bottom of each rectangle”, using the fact that the width of each slice is 1 and the sum of their heights is \(\text{OPT}(Q)\), using the result of Graham et al. [7] that the weighted sum of every set of numbers whose sum is at most 1 is bounded by 1.7, and summing over the heights of the slices, we obtain

\[A(S^*) \leq 1.7\text{OPT}(Q)\]

Since the width of every rectangle in \(T2\) is more than 1/2, \(W(r) > 1\) for every rectangle in \(T2\). Hence,

\[H(B2) < \sum_{r \in T2} h(r)W(r)\]

So,

\[
\text{FFDH} \text{Mix}(Q) = H(B1) + H(B2)
\]

\[
< \sum_{r \in T1} h(r)W(r) + 1 + \sum_{r \in T2} h(r)W(r) \quad (\text{Lemma 15})
\]

\[= A(S^*) + 1 \leq 1.7\text{OPT}(Q) + 1\]

An OCOSP instance that establishes the tightness of the bound is easily constructed from the instance used in [3] to establish the tightness of the bound for First Fit bin packing; the sizes in that instance become durations in the OCOSP instance and all jobs have the same power requirement.
### 2.5 Comments on the Proof

We make a few remarks about our proof.

1. The weighted area of rectangles in $OPT$ may be less than the weighted area of rectangles in $FFDHMix$ (see Figure 5). This is because $OPT$ and $FFDHMix$ may slice a preemptable job in different ways resulting in different weighted sums. This is the reason why we had to reduce the weight of each preemptable job to its minimum, say $W(d_i)$ to $\frac{6}{5} \times d_i$.

![FFDHMix Scheduling and Optimal Scheduling](image)

Figure 5: $FFDHMix$ and optimal schedule for a job set with dimensions (and corresponding rectangles) for jobs $j_i, 1 \leq i \leq 7$ : $\frac{1}{3} \times \frac{2}{3}$ ($r_1$ in both a and b), $\frac{1}{6} \times \frac{2}{3}$ ($r_3^{pre}$ in a, $r_3^{pre}$ and $r_4^{pre}$ in b), $\frac{1}{12} \times \frac{2}{3}$ ($r_4$ in a, $r_6$ in b), $\frac{1}{12} \times \frac{2}{3}$ ($r_5$ in both a and b) and $\frac{1}{12} \times \frac{2}{3}$ ($r_6$ in a, $r_7$ in b). Of these only $j_3$ is preemptable. The weighted sums of (a) and (b) are $\frac{104}{120}$ and $\frac{100}{120}$ respectively.

2. In [8], a regular rectangle is one which is placed at the topmost level at the time it was scheduled. A preemptable job could have two such rectangles and we need to consider these separately (as a 2-reg job in our proof).

3. Consider a set of non-preemptable jobs $J1$ and a corresponding set $J2$ in which some of the jobs of $J1$ are declared preemptable. $FFDHMix$ may construct a schedule for $J2$ that has a larger peak power demand than the schedule for $J1$ (see Figure 6). Our proof reassures us that the worst-case bound is the same for power demand scheduling of non-preemptive jobs using $FFDH$ and that for scheduling a mixed set of jobs using our adaptation $FFDHMix$ of $FFDH$.
Figure 6: Example showing that application of FFDHMix could result in better result when none of the jobs are preemptable compared to the case when some of the jobs are preemptable for a same set of job. The strip width is 10 and all jobs have same height. The widths of the jobs $j_i$, $1 \leq i \leq 6$ are: 6 ($r_1$ in both a and b), 7 ($r_2^{pre}$ and $r_3^{pre}$ in a, $r_3$ in b), 5 ($r_4$ in a, $r_5$ in b), 5 ($r_5$ in a, $r_6$ in b), 4 ($r_6$ in a, $r_2$ in b) and 3 ($r_7$ in a, $r_4$ in b). Of these, $j_2$ is preemptable in (a).

3 Complexity of FFDHMix

**Theorem 2** The number of levels in an FFDHMix schedule that have unused space is at most $q + 1$, where $q$ is the number of non-preemptive jobs.

**Proof** We show that following the scheduling of the $i$th job by FFDHMix (i.e., $i$th iteration of the outermost while loop of Figure 1), the number of levels with unused space is at most $s_i + 1$, where $s_i$ is the number of non-preemptable jobs among jobs $1, \cdots, i$. When $i = 1$, $s_i$ is 0 or 1 and the number of levels with unused space is at most 1. Assume that the number of levels with unused space is at most $s_u + 1$ following the iteration with $i = u$. If job $u + 1$ is non-preemptive and does not fit into the unused space of an existing level, a new level is started and the number of levels with unused space potentially increases by 1. Since, $s_{u+1} \geq 0$. So, the number of levels with unused space remains at most $s_{u+1} + 1$. When job $u + 1$ is preemptive, we have two cases to consider. In the first, none of the existing levels have unused space. Now, a new level is started and the number of levels with unused space is at most 1. Also, $s_{u+1} \geq 0$. So, the number of levels with unused space is at most $s_{u+1} + 1$. In the second case, at least one existing level has unused space. Now, before a new level is started by FFDHMix, all the unused space on the lowest existing level that has unused space is used. So, the number of levels with unused space following the scheduling of job $u + 1$ is no more than before this scheduling began. Hence, following iteration $u + 1$, the number of levels with unused space is at most $s_u + 1 = s_{u+1} + 1$.

Hence following iteration $n$, where $n$ is the total number of jobs, the number of levels with unused space is at most $s_n + 1 = q + 1$. \hfill \qed
From Theorem 2, it follows that we can implement $FFDHMix$ so as to run in $O(n \log n + nq)$ time by maintaining an ordered list of levels that have unused space; the list is ordered by levels bottom to top. The $n \log n$ term reflects the time needed to sort the jobs into decreasing order of their power (i.e., height) demands.

4 Experimental Evaluation

We conducted experiments by constructing datasets based on typical household consumption. Our simulation model is similar to the one in [14]. We consider a local power distribution network with one energy supplier and multiple subscribers. For each household we randomly select jobs from residential appliances based on the power consumption data released by Office of Energy of Canada in 2005 [25] and Toronto Hydro [26] which is the largest municipal electricity distribution company in Canada. Some examples of appliances and their usages are dishwasher: 1350W for 45-50 hours/month, washing machine: 500W for 55-60 hours/month, PHEV (Plugable hybrid electric vehicle): 2200W for 130-140 hours/month and 5-10 bulbs: 60W for 80-120 hours/month. There are around 40 jobs for each house and roughly half of them are preemptable.

As shown in Figure 7 and Table 1, we observed up to 18% improvement in peak power when compared against $NFDH_{NonPreemptive}$ and around 2% improvement compared to $NFDH_{Mix}$.

A much higher %age, up to 21%, of improvement relative to $NFDH_{Mix}$ is exhibited on the mixed dataset that we constructed in [1] (Figure 8 and Table 2). This dataset comprises 5 sets of data with the property that all instances in one set have same number of rectangles. Each set is further classified into three subsets each consisting of 10 instances. The instances within a subset are characterized by a parameter $\lambda$, which represents the fraction of rectangles with width $> D/2$. Our instances had $\lambda \in \{0.2, 0.6, 0.8\}$. For each instance we varied the %age of preemptable rectangles to three values: 25%, 50% and 75%. Figure 8 and 9 shows the performance of $FFDH_{Mix}$ for different values of %age of preemption and $\lambda$, respectively. As can be seen from Figure 8, the %age improvement is generally higher when the preemption is 25% or 50% compared to when it is 75%. This is expected because both $FFDH_{Mix}$ and $NFDH_{Mix}$ are expected to construct similar schedules as the %age of preemptable jobs increases and eventually construct the same schedule as $NFDH_{Preemptive}$ [1] when all jobs are preemptable. Figure 9 does not show any correlation between %age improvement and variation in $\lambda$.

We also obtained up to 10% (6% when we average among all instances of a benchmark) improvement on benchmark strip packing datasets [24, 23, 22] used in [1] as shown in Figure 10 and Table 3. We see that for this dataset, the improvement decreases almost linearly with increase in the %age of preemptable jobs. This may be due to the fact that all benchmark datasets were created by splitting one large rectangle, and hence both $FFDH_{Mix}$ and $NFDH_{Mix}$ schedules tend to be closer to optimal as more preemption is allowed.

5 Conclusion

In this paper, we have considered an adaptation of the FFDH (first fit decreasing height) algorithm [8] of strip packing for mixed OCOSP. We have analyzed the worst-case performance of this adaptation and shown that the peak power demand of schedules constructed by this adaptation is at most $1 + 1.5OPT$ and that this bound is tight. Experiments indicate that the first fit adaptation $FFDH_{Mix}$ generates schedules with about 2% lower peak power demand.
Figure 7: Improvement over NFDHNonPreemptive and NFDHMix [1] on household simulation dataset

Figure 8: Improvement over NFDHMix on datasets constructed in [1] with variation of %age of Preemption on publicly available data sets relative to NFDHMix; the reduction is up to 21% on synthetic data.
Figure 9: Improvement over $NFDHMix$ on datasets constructed in [1] with variation of $\lambda$

Figure 10: Improvement over $NFDHMix$ on benchmark strip packing datasets with variation of %age of Preemption

References


Table 1: Experimental Results for household data

<table>
<thead>
<tr>
<th>Number of Houses</th>
<th>Improvement over</th>
<th>Improvement over</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NFDH</td>
<td>NFDHMx</td>
</tr>
<tr>
<td>5</td>
<td>11.53</td>
<td>1.51</td>
</tr>
<tr>
<td>10</td>
<td>16.43</td>
<td>2.13</td>
</tr>
<tr>
<td>50</td>
<td>17.89</td>
<td>2.18</td>
</tr>
<tr>
<td>100</td>
<td>18.00</td>
<td>2.18</td>
</tr>
<tr>
<td>500</td>
<td>18.08</td>
<td>2.21</td>
</tr>
</tbody>
</table>

Table 2: Improvement over NFDHMx on dataset constructed in [1].

<table>
<thead>
<tr>
<th>Number of Jobs</th>
<th>A</th>
<th>Improvement</th>
<th>% of Preemption</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>% of Jobs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>25%</td>
<td>50%</td>
</tr>
<tr>
<td>25</td>
<td>0.2</td>
<td>13.67</td>
<td>10.34</td>
</tr>
<tr>
<td>25</td>
<td>0.6</td>
<td>12.00</td>
<td>9.67</td>
</tr>
<tr>
<td>25</td>
<td>0.8</td>
<td>6.24</td>
<td>12.33</td>
</tr>
<tr>
<td>50</td>
<td>0.2</td>
<td>14.68</td>
<td>9.85</td>
</tr>
<tr>
<td>50</td>
<td>0.6</td>
<td>9.66</td>
<td>16.90</td>
</tr>
<tr>
<td>50</td>
<td>0.8</td>
<td>8.03</td>
<td>15.51</td>
</tr>
<tr>
<td>100</td>
<td>0.2</td>
<td>19.03</td>
<td>14.53</td>
</tr>
<tr>
<td>100</td>
<td>0.6</td>
<td>12.26</td>
<td>15.78</td>
</tr>
<tr>
<td>100</td>
<td>0.8</td>
<td>8.60</td>
<td>16.43</td>
</tr>
<tr>
<td>200</td>
<td>0.2</td>
<td>19.73</td>
<td>14.94</td>
</tr>
<tr>
<td>200</td>
<td>0.6</td>
<td>14.44</td>
<td>17.61</td>
</tr>
<tr>
<td>200</td>
<td>0.8</td>
<td>10.42</td>
<td>18.51</td>
</tr>
<tr>
<td>500</td>
<td>0.2</td>
<td>21.95</td>
<td>17.37</td>
</tr>
<tr>
<td>500</td>
<td>0.6</td>
<td>13.90</td>
<td>19.87</td>
</tr>
<tr>
<td>500</td>
<td>0.8</td>
<td>10.24</td>
<td>19.48</td>
</tr>
</tbody>
</table>

Table 3: Improvement over NFDHMx for benchmark strip packing dataset

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Num of Instances</th>
<th>Num of Jobs</th>
<th>% of Preemption</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>25% 50% 75%</td>
</tr>
<tr>
<td>HT</td>
<td>1</td>
<td>10</td>
<td>4.35 4.35 0.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>20-30</td>
<td>7.65 10.92 6.78</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>40-80</td>
<td>1.59 1.59 0.00</td>
</tr>
<tr>
<td>Burke</td>
<td>3</td>
<td>50-70</td>
<td>5.14 6.19 3.21</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>100-300</td>
<td>9.55 5.06 4.19</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>500</td>
<td>0.81 1.35 0.54</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Num of Instances</th>
<th>Num of Jobs</th>
<th>% of Preemption</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>25% 50% 75%</td>
</tr>
<tr>
<td>HT</td>
<td>3</td>
<td>49</td>
<td>6.00 4.75 2.25</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>73</td>
<td>10.45 3.52 1.63</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>97</td>
<td>5.40 4.21 3.11</td>
</tr>
<tr>
<td>Burke</td>
<td>3</td>
<td>196-197</td>
<td>1.77 3.24 1.26</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>25</td>
<td>7.76 2.08 0.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>16-17</td>
<td>4.76 2.47 2.47</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>28-29</td>
<td>6.41 5.48 0.00</td>
</tr>
<tr>
<td>path</td>
<td>50</td>
<td>25</td>
<td>2.85 1.45 2.20</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>50</td>
<td>3.88 2.95 0.43</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>100</td>
<td>3.29 2.78 1.22</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>200</td>
<td>2.62 1.75 0.70</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>500</td>
<td>1.75 1.28 0.79</td>
</tr>
<tr>
<td>nice</td>
<td>50</td>
<td>25</td>
<td>3.30 2.44 1.20</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>50</td>
<td>4.20 2.36 0.80</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>100</td>
<td>3.87 2.26 1.22</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>200</td>
<td>2.56 1.63 0.78</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>500</td>
<td>1.74 0.89 0.28</td>
</tr>
</tbody>
</table>