Offline Preemptive Scheduling of Jobs in Smart Grids*

Anshu Ranjan, Pramod Khargonekar, Sartaj Sahni
University of Florida,
aranjan@cise.ufl.edu, ppk@ece.ufl.edu, sahni@cise.ufl.edu

***Abstract*—We consider the scheduling of flexible electric loads in a smart grid so as to minimize peak power demand. Specifically, we focus on the case when the loads are preemptable, their power requirement and duration are known in advance, and they have the same earliest start time and the same deadline. Our main results are (a) when power requests are scheduled preemptively, the peak power demand can be reduced by up to 50% relative to when these requests are scheduled non-preemptively, (b) preemptive scheduling to minimize peak power demand is NP-hard, (c) schedules with minimum peak power demand may be constructed using integer linear programming, and (d) the next-fit decreasing height heursitic may be used to quickly obtain schedules whose peak power demand is at most two times that of the optimal schedule when all jobs are preemptable and at most three times the optimal when only some jobs are preemptable. Experimental results for the integer linear program and the heuristic are also presented. Our experiments indicate a significant reduction in peak power when preemption is exploited. For example, on our data sets recharging collections of electric and plug-in hybrid vehicles without preemption required up to 26% more peak power than when this was done preemptively.**

I. Introduction

Total installed generation capacity in an electric energy system (controlled by a utility company or an idependent system operator) is determined, in large part, by the peak load served by the system. For example, the potential large scale adoption of electric vehicles (EVs) could cause the peak load to increase dramatically unless charging of electric vehicles is carefully managed. Motivated by the desire to integrate (uncertain and variable) renewable electric generation and maximize the use existing electricity grid infrastructure assets, there has been a siginificant interest in leveraging the inherent flexibility in certain types of electric energy consumption loads. Examples include washers, dryers, water heaters, EV charging, etc. With installation of smart meters and two-way communications between loads and central electric utility, it is or will soon be possible to address individual loads for residential and commercial customers. Thus, it is envisioned that demand side flexibility could be harnessed for goals such as peak reduction, renewable integration, and optimization of grid operations.

Consider a scenario in which utilizing the two-way communication infrastructure, consumers can inform the utility company of planned future electricity usage. Thus, for example, a consumer could inform the utility company at 10am of his/her need to run his/her washing machine or dishwasher or to charge his/her car between 6pm (earliest start time) that evening and 6am (deadline) the following morning and that for this purpose he/she would need pKW of power for a duration of d hours. The utility company could then aggregate all such consumer requests and schedule them so as to minimize its cost. Once the cost-optimal schedule has been determined, the utility company can inform the consumer or in the case of smart appliances, it can remotely turn appliances on and off as per the optimal schedule.

In general, these types of demand scheduling algorithms will need to be embedded in the day-ahead and intra-day power system operations and control frameworks. We envision a rolling horizon approach in which a scheduling algorithm optimizes load schedule for the next time horizon. As new load requests arrive randomly, the scheduling algorithm is rerun for the next time horizon. Such a scheme can be thought of as a version of the moving horizon control paradigm. Thus, in this paper, we focus on offline scheduling of flexible electric loads to minimize peak power demand. Minimizing the peak power demand has signficant potential to impact capital investment needs in generation equipment. We defer a study of other optimization metrics such as minimizing total cost to the provider using a cost model that accounts for (say) forecast availability of wind and solar power in the scheduling period to future research. For simplicity, in the rest of the paper, we will use the term "cost-optimal schedule" to mean a schedule that minimizes the peak power demand. For the offline scheduling model, we limit ourselves to the simplified case when all consumer requests specify the same earliest start time and the same deadline. Also, we will use the term job to denote a consumer load request. A job is characterized by an earliest start time, a duration, a deadline, and a power requirement.

We consider preemptive and non-preemptive schedules. In a preemptive schedule, a request for p units of power for a duration of d may be satisfied by scheduling the job for noncontiguous periods of time of total duration d; at each time instance during which the request is scheduled, it is provided p units of power. On the other hand, in a non-premptive schedule a request for p units of power for a duration of d may be met only by providing the request p units of power for a contiguous interval of d units. Our main contributions are: i.

1) We study the relationship between the preemptive and non-preemptive offline power scheduling problems with the 2D strip packing problem and show that while non-premptive offline power scheduling and 2D strip packing are identical problems, the premptive versions of these problems are different. Preemptive offline power scheduling becomes identical to preemptive 2D strip packing when constraints are added

*This research was supported, in part, by the National Science Foundation under grants CNS0829916, CNS0905308, and CNS-1239274.

to the latter.

2) We show that when power requests are scheduled preemptively, the peak power demand can be reduced by up to 50% relative to when these requests are scheduled non-preemptively and this bound is tight.
3) We show that the preemptive offline power scheduling problem can be solved in $O(n)$ time, where n is the number of consumer requests, when all requests are for the same amount of power. However, when all requests are for the same duration, the problem is NP-hard. Prior research has establised that the non-preemptive offline version is NP-hard when either all requests have the same power requirement or all have the same duration.
4) We develop integer linear programming (ILP) formulations for the preemptive and mixed (i.e., some requests may be scheduled preemptively while others must be scheduled non-preemptively) offline power scheduling problems under the assumption that jobs must be scheduled in integral quantums of time.
5) We extend the simplest of the heuristics proposed for 2D strip packing, next-fit decreasing height, to the offline preemptive and mixed power scheduling problems and derive tight absolute and asymptotic performance bounds.
6) Extensive computational experiments are conducted to assess the performance of our proposed solutions in practice.

The rest of the paper is organized as follows. In Section II, we discuss related work. In Section III, we study properties of the preemptive power scheduling problem including its relationship with strip packing. Our ILP formulations for preemptive and mixed power scheduling are developed in Section IV. In Section V we extend the next-fit decreasing height heuristic to preemptive and mixed power scheduling and derive tight performance bounds. Experimental results are presented in Section VI. Finally, we conclude in Section VII.

II. Related Work

Cost-optimal scheduling of jobs has been considered by Koutsopoulos and Tassiulas [1]. They propose using the iterative load balancing algorithm of [2] for the case when jobs have (possibly) different earliest start times and deadlines, the power-cost function is convex, and all jobs may be scheduled preemptively. Koutsopoulos and Tassiulas show that this iterative algorithm converges to the optimal schedule. We show later in this paper that this offline scheduling problem is NP-hard even when all jobs have the same duration, same earliest start time, and the same deadline. Hence the convergence rate of the iterative algorithm of [2] must be rather slow.

The non-preemptive offline scheduling problem is shown to be NP-hard in [1]. This is the case even for the simple case when all jobs have the same earliest start time, the same deadline, and the same power requirement [1]. Koutsopoulos and Tassiulas [1] also study the online power demand scheduling problem and propose asymptotically optimal scheduling policies for infinite time horizons.

Demand side power management and strip packing are related to cost-optimal power scheduling. In demand side power management [3], the focus is to regulate the energy usage patterns of consumers through pricing policy. Game theoretic approaches to demand side power management are proposed in [4]–[9], for example.

Various researchers have studied and developed algorithms for "smart charging" of electric and plug-in hybrid vehicles (EV and PHEV) [10]–[17] drawing on techniques from control and optimization literature. Our work deals with related problems but uses very different techniques leading to new results that complement existing literature.

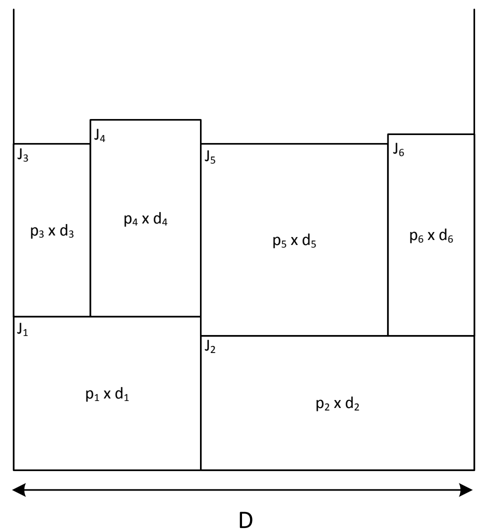


Fig. 1: Strip packing 5 jobs J_1 - J_6 into a rectangle of width D

We draw on the literature on the strip packing problem which is described next. The objective of strip packing is to pack, without rotation, a set of rectangles into a minimum height rectangle of specified width. We can model each job in an instance of the offline power scheduling problem as a rectangle whose width is the duration of the job and whose height is the power requirement of the job. When all jobs have the same earliest start time (without loss of generality, we may assume the common earliest start time to be 0) and the same deadline (say D), the cost-optimal non-preemptive scheduling problem is identical to strip-packing the corresponding rectangles into a minimum height rectangle whose width is D. Figure 1 provides such an example. The schedule defined by this strip packing services job J_1 from 0 to d_1, J_2 from d_1 to $d_1 + d_2 = D$, J_3 from 0 to d_3, J_4 from d_3 to $d_3 + d_4$, J_5 from $d_3 + d_4$ to $d_3 + d_4 + d_5$, and J_6 from $d_3 + d_4 + d_5$ to $d_3 + d_4 + d_5 + d_6 = D$. The peak power for this schedule is $p_1 + p_3 = p_1 + p_4 = p_2 + p_5 = p_2 + p_6$. The NP-hardness of non-preemptive cost-optimal scheduling follows from the corresponding result for strip packing [18], [19]. Coffman et al. [18], [19] note that strip packing is NP-hard even when restricted to rectangles that have the same height or the same width. Hence, cost-optimal non-preemptive scheduling of power demands is NP-hard even when all earliest start times are the same, all deadlines are the same and all demands have the same power requirement (but different durations) or have the same duration (but different power requirements).

Several heuristics [18]–[26] as well as a fully polynomial approximation scheme [27] and polymial approximation schemes [28], [29] have been proposed for strip packing. Coffman et al. [19] analyze the performance of three heuristics– NFDH (next-fit decreasing height), FFDH (first-fit decreasing height), and SF (split fit)–for strip packing. They obtain the following asymptotic performance bounds for these three heuristics under the assumption that rectangle heights have been normalized so that the tallest rectangle has a height of 1:

$$NFDH(I) \leq 2OPT(I) + 1 \tag{1}$$

$$FFDH(I) \leq 1.7OPT(I) + 1 \tag{2}$$

$$SF(I) \leq 1.5OPT(I) + 2 \tag{3}$$

where $OPT(I)$ is the minimum height into which the instance I can be packed.

Because the strip packing problem is equivalent to the offline non-preemptive cost-optimal scheduling problem when all jobs have the same earliest start time and the same deadline, the heuristics of [19] (as well as all other strip packing heuristics and approximation schemes) provide the same performance bounds when used for the latter problem.

The relationship between strip packing and OCOSP has also been studied by Alamdari et al. [30] and [31]. [30] and [31] propose a fully polynomial time approximation scheme for preemptive OCOSP. This FPTAS involves the use of linear programs. Although these linear programs may be solved in polynomial time using the ellipsoid method, the ellipsoid method is not very practical. Hence, the FPTAS is not expected to run in acceptable time [31]. Alamdari et al. also propose an $O(n^2)$ algorithm with a performance ratio of 3/2, an $O(n \log(nM))$ algorithm with a performance ratio of 5/3 (this algorithm has at most 3 preemptions per job), and an $O(n \log^2 n \log(nM)/ \log\log n)$ algorithm with performance ratio 5/3 that has at most 1 preemption per job.

III. Properties of Preemptive Scheduling of Power Demands

A. Relationship to Strip Packing

Henceforth, we shall use OCOSP as an acronym for the offline cost-optimal scheduling problem when all jobs have the same earliest start time and the same deadline. Although non-preemptive OCOSP is identical to strip packing, there does not appear to be a corresponding equivalence between preemptive OCOSP and any reasonable preemptive version of strip packing. In preemptive strip packing we would permit the cutting of rectangles into pieces and then pack these pieces into a minimum height rectangle of specified width. With rectangle height corresponding to power requirement and width to duration, cutting of rectangles can be permitted only along the width (so we can create narrower rectangles but not shorter ones) as in preemptive OCOSP, preemption is permitted only along the duration axis. However, restricting rectangle cutting to the width dimension isn't sufficient. To see this, note that a rectangle of height 2 and width 15 can be packed into a rectangle of height 6 and width 5 by cutting the rectangle into 3 pieces of width 5 (Figure 2). However, if the height 2 width 15 rectangle represents a job with power requirement 2KW and duration 15 hours, cutting and packing in this way would correspond to delivering 6KW for a 5 hour period, which is not acceptable and would probably destroy the device being operated.

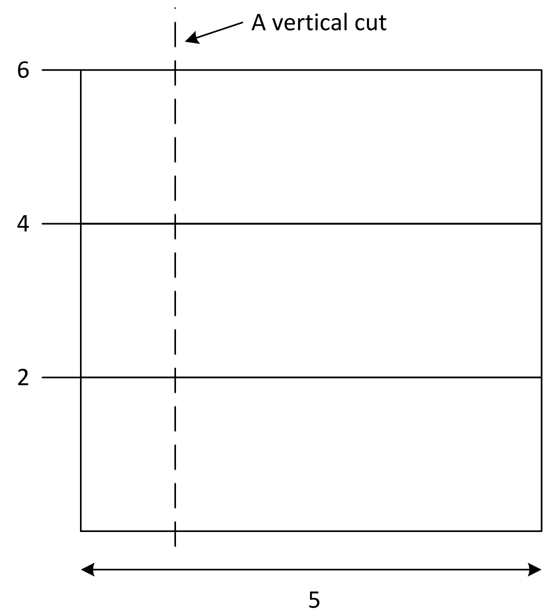


Fig. 2: Preemptive strip packing of a 2×15 rectangle into a rectangle whose height is 6 and width is 5

Thus, besides restricting rectangle cutting to the width dimension we need another restriction on strip packing to get an equivalence with preemptive OCOSP. We need to limit ourselves to packings with the property that there is no vertical cut that has two pieces of the same rectangle. In the preemptive strip packing of Figure 2, every vertical cut includes all 3 pieces of the original 2×15 rectangle.

B. Relationship to Non-preemptive OCOSP

Consider an n job instance I of OCOSP. Let d_i be the duration of job i and p_i its power requirement, $1 \leq i \leq n$. Let D be the common deadline of the n jobs. We may assume that $d_i \leq D$, $1 \leq i \leq n$. Suppose that all jobs are preemptable. In this section we address the question "How much better can an optimal preemptive schedule for I be than an optimal non-preemptive one?".

Theorem 1: Let $OPT(pre)$ and $OPT(non)$, respectively, denote the maximum power demands of optimal preemptive and non-preemptive schedules for I. $OPT(pre) \leq OPT(non) \leq 2OPT(pre)$ and this bound is tight.

Proof: Since every non-preemptive schedule is also a preemptive one (but with no preemptions) $OPT(pre) \leq$

$OPT(non)$. The tightness of this bound is easily established by considering instances in which the duration of each job is D, the common deadline. For these instances, $OPT(pre) = OPT(non) = \sum p_i$.

To prove $OPT(non) \leq 2OPT(pre)$, we note that [22] develops an algorithm for non-preemptive strip packing (and hence for non-preemptive OCOSP), which by Theorem 1.1 of [22], succeeds whenever

$$2A \leq uD - (2p_{\max} - u)_+(2d_{\max} - D)_+,$$

where $p_{\max} = \max\{p_i\}$, $d_{\max} = \max\{d_i\}$ $(1 \leq i \leq n)$, $A = \sum p_i d_i$, $x_+ = \max\{x, 0\}$, $u \geq p_{\max}$ is the height of the rectangle into which the strip packing is done (i.e., the maximum power demand), and D is the width of this rectangle. From this, it follows that

$$OPT(non) \leq 2\max\{p_{\max}, A/D\}.$$

To see this, let $u = 2\max\{p_{\max}, A/D\}$. If $u = 2p_{\max} \leq 2A/D$, then from Theorem 1.1 of [22], the algorithm of [22] is able to pack the rectangles corresponding to the jobs into a $u \times D$ rectangle because

$$uD - (2p_{\max} - u)_+(2d_{\max} - D)_+ = uD = 2p_{\max}D \geq 2A.$$

When $u = 2A/D \geq 2p_{\max}$,

$$uD - (2p_{\max} - u)_+(2d_{\max} - D)_+ = uD = 2A.$$

Since $OPT(pre) \geq \max\{p_{\max}, A\}$, $OPT(non) \leq 2OPT(pre)$. Figure 3 gives an example for which $OPT(non)/OPT(pre) \rightarrow 2$ as $n \rightarrow \infty$ thus establishing the tightness of the bound. ■

We note that while Alamdari et al. [31] demonstrate that preemption can reduce peak power by a factor of 2, they do not show, as we do, that a larger reduction is not possible.

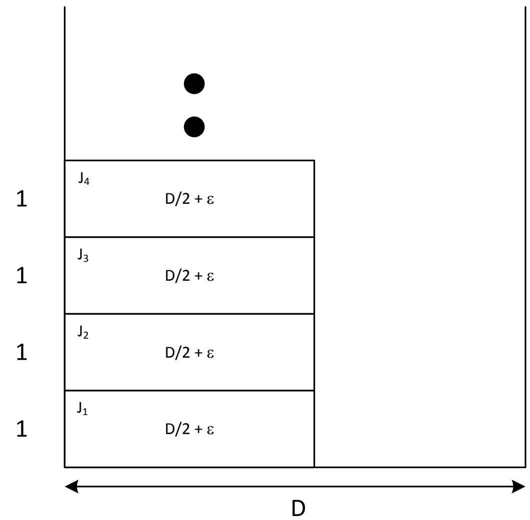
C. Case When All Jobs Have Same Power Requirement

It is interesting to note that while non-preemptive OCOSP is NP-hard when all jobs have the same power requirement, preemptive OCOSP with jobs that have the same power requirement can be solved in $O(n)$ time, where n is the number of jobs. The algorithm *PreemptiveEqualPower* given in Figure 4 does this. In this algorithm, p is the common power required by the jobs. The algorithm assumes that $D \geq d_i$, $1 \leq i \leq n$. We note that when $D < d_j$ for some j, the deadline D isn't sufficient for the duration required by the jth job. The correctness of the algorithm follows from the observation that the cost (i.e., peak power demand) of the schedule generated by *PreemptiveEqualPower* is $p\lceil \sum d_i/D \rceil$, which is optimal.

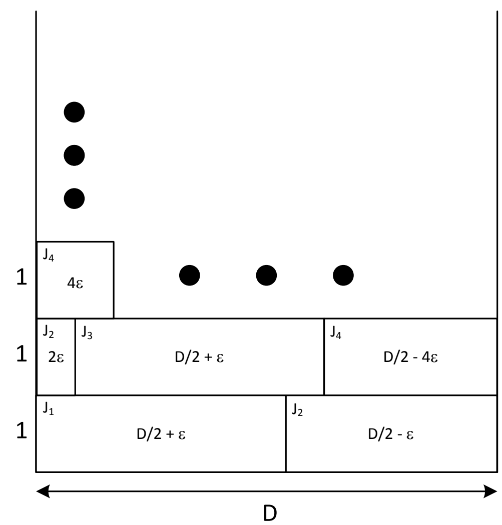
D. Case When All Jobs Have The Same Duration

When all jobs have the same duration but different power requirements, preemptive OCOSP is NP-hard. This follows from the fact that the partition problem, in which we are given n positive integers s_1, s_2, $\cdots$ s_n and wish to determine whether these may be partitioned into two groups that have the same sum, is NP-complete.

Theorem 2: Preemptive OCOSP is NP-hard when all jobs have the same duration.



(a) Optimal scheduling for non-preemptive version



(b) Optimal scheduling for preemptive version

Fig. 3: Example showing that the bound $OPT(non) \leq 2OPT(pre)$ is asymptotically tight where n jobs need to be scheduled each having identical power requirement and duration. The duration of each job is $D/2 + \epsilon$. a) $OPT(non) = n$, b) $OPT(pre) = \lceil n(W/2 + \epsilon)/W \rceil \rightarrow 2$ as $n \rightarrow \infty$.

Proof: Let s_1, s_2, $\cdots$, s_n be any instance IP of the partition problem. Construct a corresponding instance $IPOCOSP$ of the preemptive OCOSP problem. This $IPOCOSP$ has n jobs, the duration of each is 1, the power requirement p_i of job i is s_i, and the common deadline of the n jobs is $D = 2$. We may verify that $IPOCOSP$ has a preemptive schedule with maximum power demand $\sum p_i/2$ iff IP has a partition. Hence, preemptive OCOSP with equal duration jobs is NP-

Algorithm $PreemptiveEqualPower$
{ // optimal preemptive schedule when each job requires power p

```
    pMax = 0; time = D;
    for (i = 1; i <= n; i++)
       // schedule job i
       if (time + d_i <= D)
          {
             output schedule job i from time to time + d_i;
             time += d_i
          }
       else
          {
             if (time == D)
                {
                   output schedule job i from 0 to d_i;
                   time = d_i;
                }
             else
                {
                   output schedule job i from time to D as well as from 0 to d_i - D + time;
                   time = d_i - D + time;
                }
             pMax += p;
          }
    output Max power demand is pMax;
}
```

Fig. 4: Algorithm to construct a cost-optimal preemptive schedule when all jobs have the same power requirement

hard. ■

From Theorem 2, it follows that the general preemptive OCOSP problem is also NP-hard. We note that Alamdari et al. [31] have independently shown that preemptive OCOSP with equal duration jobs is NP-hard.

IV. ILP Formulations

In this section, we develop Integer Linear Programming (ILP) formulations for the preemptive and mixed OCOSP problems. For the ILP, we assume that a quantum Δ of time is specified such that whenever a job is scheduled, it is scheduled for an integer number of these quantums. A quantum may, for example, be 1 second, 1 minute, 5 minutes and so on. With this assumption, all d_i and D may be assumed to be multiples of Δ. Further, we may assume that there are no preemptions within any Δ time interval. Without loss of generality, we assume in the following that $\Delta = 1$ and so, a Δ interval is a unit interval.

A. Preemptive OCOSP

Let x_{ij} be a binary valued variable with the interpretation that x_{ij} is 1 iff job i is scheduled in the jth unit interval, where the jth unit interval is from time $j-1$ to time j. Our ILP formulation for preemptive OCOSP is given below.

$$\text{minimize } P \tag{4}$$

subject to:

$$\sum_{j=1}^{D} x_{ij} = d_i, \quad 1 \le i \le n \tag{5}$$

$$\sum_{i=1}^{n} p_i x_{ij} \le P, \quad 1 \le j \le D \tag{6}$$

$$x_{ij} \in \{0,1\}, \quad 1 \le i \le n, \ 1 \le j \le D \tag{7}$$

Here, P represents the maximum power demand of the schedule defined by the x_{ij}s, p_i is the power requirement of job i and d_i is its duration. Equation 4 seeks to minimize the maximum power demand while Equation 5 ensures that each job is scheduled for as many unit intervals as needed to complete it and Equation 6 ensures that the total power required by the jobs scheduled in unit interval j does not exceed the schedule's maximum power demand of P. Equation 7 ensures the binary property of the x_{ij}s.

The correctness of the above ILP follows from the observations (a) preemption in the time dimension but not in the power dimension is permitted and (b) in any unit interval a job is scheduled at most once and so the corresponding strip packing has no vertical cut that includes the same job twice.

The total number of variables in our ILP formulation is nD and the total number of constraints is $nD + n + D$.

B. Mixed OCOSP

Our mixed OCOSP ILP replaces each preemptable job i with d_i nonpremptable jobs that have a duration of 1 and a power requirement of p_i. Constraints are added to the ILP for non-preemptive OCOSP so that two unit interval jobs that correspond to the same preemptable job are not scheduled for the same time. Let $[t_j, t_j + d_j]$ denote the time interval for which the non-preemptable job j is scheduled. Let i_1, i_2, $\cdots$, i_{d_i} denote the unit interval non-preemptable jobs that replace the preemptable job i. To ensure that no two of these unit interval jobs are scheduled at the same time, we add the constraints

$$t_{i_1} < t_{i_2}, \ t_{i_2} < t_{i_3}, \ \cdots, \ t_{i_{d_i - 1}} < t_{i_{d_i}}$$

to the ILP for the non-preemptive OCOSP.

So, we need only consider the development of an ILP for non-preemptive OCOSP. We use 4 sets of binary variables l_{ij}, r_{ij}, a_{ij}, and b_{ij}, $1 \le i < j \le n$, where n is the number of jobs in the non-preemptive OCOSP instance. Assigning a 1 to l_{ij} has the interpretation job i is scheduled to the left of job j (i.e., job i completes at or before job j starts) and assigning 1 to r_{ij} means that job i is scheduled to the right of job j. Note that when $l_{ij} = r_{ij} = 0$ the scheduling of jobs i and j overlaps and that $l_{ij} = r_{ij} = 1$ is an infeasible assignment of values. To explain the significance of a_{ij} and b_{ij}, we note that every non-preemptive schedule may be drawn as a packing of rectangles, whose width is job duration and height is power requirement, into a rectangle of width D and height equal to the maximum power demand of the schedule(see Figure 1). Although many different such drawings are possible for a given schedule, it is sufficient for our purpose that at least one such drawing exists.

Consider such a drawing. The binary variable a_{ij} is 1 iff job i is above job j (i.e., the y value of the bottom edge of job i's rectangle is greater than or equal to the y value of the top edge of job j's rectangle). b_{ij} is 1 iff i is below j in the drawing of the schedule. Let M be a large integer.

Our ILP for the non-preemptive OCOSP problem follows. y_i denotes the y value of the bottom edge for the rectangle for job i in the drawing for the schedule.

$$\text{minimize } P \tag{8}$$

$$t_i + d_i \leq D, \quad 1 \leq i \leq n \tag{9}$$

$$y_i + p_i \leq P, \quad 1 \leq i \leq n \tag{10}$$

$$t_i + d_i \leq t_j + M * (1 - l_{ij}), \quad 1 \leq i < j \leq n \tag{11}$$

$$t_j + d_j \leq t_i + M * (1 - r_{ij}), \quad 1 \leq i < j \leq n \tag{12}$$

$$y_i + p_i \leq y_j + M * (1 - b_{ij}), \quad 1 \leq i < j \leq n \tag{13}$$

$$y_j + p_j \leq y_i + M * (1 - a_{ij}), \quad 1 \leq i < j \leq n \tag{14}$$

$$a_{ij} + b_{ij} + l_{ij} + r_{ij} \geq 1, \quad 1 \leq i < j \leq n \tag{15}$$

$$a_{ij},\ b_{ij},\ l_{ij},\ r_{ij} \in \{0,1\}, \quad 1 \leq i < j \leq n \tag{16}$$

$$t_i, y_i \geq 0, \quad 1 \leq i \leq n \tag{17}$$

Equation 9 ensures that all jobs complete by their common deadline while Equation 10 ensures that the maximum power demand of the schedule does not exceed P. Equations 11 and 12 together eliminate the infeasible assignment $l_{ij} = r_{ij} = 1$ because d_i and d_j are greater than 0. Similarly, Equations 13 and 14 together eliminate the infeasible assignment $a_{ij} = b_{ij} = 1$. Equation 15 requires at least one of a_{ij}, b_{ij}, l_{ij} and r_{ij} be 1. Together with Equations 11-14, this means that i must be to the left or right of j or i must be above or below j or both. Equivalently, an assignment of binary values to the as, bs, ls, and rs together with nonnegative integer values to the ts and ys satisfies Equations 9 through 15 iff it corresponds to a non-preemptive schedule with deadline D and maximum power demand no more than P.

The ILP for non-preemptive OCOSP has $2n$ integer variables, $2n(n-1)$ binary variables, and $(5n^2 - n)/2$ constraints excluding those of Equations 16 and 17.

V. NFDH Heuristic For Preemptive and Mixed OCOSP

A. Description of the NFDH heuristic

As noted in Section II, the NFDH heuristic for strip packing was analyzed in [19], where it was shown that when rectangles' heights are normalized to 1, $NFDH(I) \leq 1 + 2OPT(I)$. When heights are not normalized in this way, $NFDH(I) \leq 3OPT(I)$ and this bound is tight [19]. We note that an asymptotic approximation bound for an algorithm A has the format

$$A(I) \leq \alpha OPT(I) + \beta \tag{18}$$

where $A(I)$ is the value of the solution produced by algorithm A working on instance I and $OPT(I)$ is the value of the optimal solution for that instance. In an absolute approximation bound, $\beta = 0$.

Hence, the tight asymptotic approximation bound for NFDH has $\alpha = 2$ and $\beta = 1$ for normalized instances while the tight absolute approximation bound has $\alpha = 3$. Since strip packing is identical to non-preemptive OCOSP, NFDH may be used for non-preemptive OCOSP to obtain power schedules such that $NFDH(non) \leq 3OPT(non)$ (or, $NFDH(non) \leq 1 + 2OPT(non)$ for normalized instances), where $NFDH(non)$ is the maximum power demand of the schedule constructed using NFDH. Although NFDH does not provide the best performance guarantee of the known heuristics for strip packing, it is the simplest of the known heuristics and, in practice, performs almost as well as the others. In this section, we extend NFDH to preemptive and mixed OCOSP and derive tight asymptotic and absolute performance bounds.

Figure 5 gives the NFDH heuristic for non-preemptive OCOSP.

```
Algorithm NFDHNonpreemptive
{
  // non-preemptive next fit decreasing height scheduling
  // sort the n jobs into decreasing order of height and reindex them
  // by this sorted ordering so that job 1 has maximum height;
  pMax = 0; time = D;
  for (i = 1; i <= n; i++)
    // schedule job i
    if (time + d_i <= D)
      {
        output schedule job i from time to time + d_i;
        time += d_i;
      }
    else
      {
        output schedule job i from 0 to d_i;
        time = d_i;
        pMax += p_i;
      }
  output Max power demand is pMax;
}
```

Fig. 5: Non-preemptive power scheduling using NFDH

Let i_1, i_2, $\cdots$, i_k be the values of the **for** loop index i at which $pMax$ changes. We observe that $i_1 = 1$. Let P_j be the value of $pMax$ after it has been updated in the iteration $i = i_j$. We note that $P_1 = p_1$ and P_k is the terminal value of $pMax$. The schedule constructed by $NFDHNonpreemptive$ may be drawn as a packing of rectangles that represent jobs into a rectangle whose height is P_k and width is D in such a way that the rectangle bottoms are aligned to one of the values $y_j = P_j - p_{i_j}$, $1 \leq i \leq k$. Figure 6 gives an example. Each y_j defines a *level* at which rectangles are placed in the drawing.

Our adaptation of $NFDHNonpreemptive$ to $NFDHPreemptive$ and $NFDHMixed$ alters the behavior of $NFDHNonpremptive$ whenever the job being scheduled in the **for** loop satisfies $time + d_i > D$ and $time < D$. In this case, $NFDHPreemtive$ schedules job i from $time$ to D as well as from 0 to $d_i - D + time$ and then updates $time$

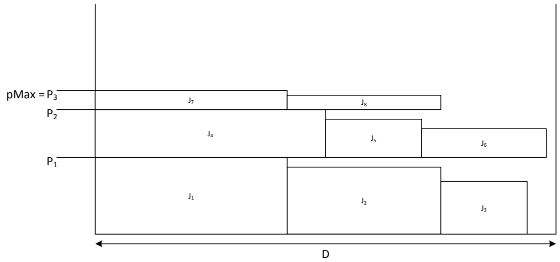


Fig. 6: Level schedule obtained by $NFDHNonpreemptive$

to $d_i - D + time$ (as in Algorithm $PremptiveEqualPower$) and updates $pMax$ to $pMax + p_i$. That is, the job is preemptively scheduled. In the case of $NFDHMixed$, job i is preemptively scheduled in this manner only if it designated as a preemptable job.

B. Performance Bounds

We establish asymptotic and absolute approximation bounds on the performance of *NFDHPreemptive* and $NFDHMixed$. It is well known that it is possible, at times, to derive asymptotic approximation bounds with a smaller α than that for the best absolute approximation bound. For example, it is known that the absolute bound $NFDH(I) \leq 3OPT(I)$ ($\alpha = 3$) is best possible for non-preemptive NFDH [19] while $\alpha = 2$ yields a tight asymptotic approximation bound for NFDH. This difference in α values carries over to the case of $NFDHPreemptive$ and $NFDHMixed$.

Theorem 3: For every instance of preemptive OCOSP, $NFDH(pre) < OPT(pre) + p_{\max}$ and this bound is tight.

Proof: Consider the level drawing (see Figure 3b for an example of a preemptive level drawing) of the schedule generated by $NFDHPreemptive$ for an n job instance of preemptive OCOSP. Let k be the number of levels and let i_1, $\cdots$, i_k be the values of the **for** loop index that started these levels. The height of level j is $h_j = p_{i_j}$ and its area A_j equals $h_j D$. Let a_j be the area of level j that is actually occupied by jobs and let $A = \sum_1^n p_i d_i$ be the sum of the areas of the jobs. Since,

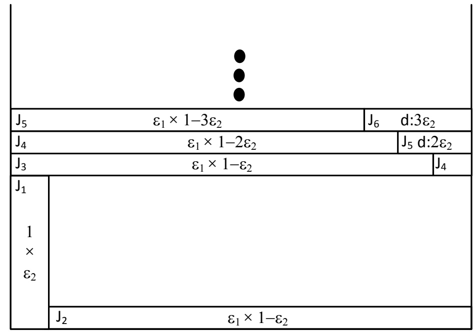
$$\begin{aligned} h_j &= p_{i_j} \leq p_{i_j - 1}, && j > 1, \\ a_j &\geq h_{j+1} D, && j < k. \end{aligned}$$

Hence,

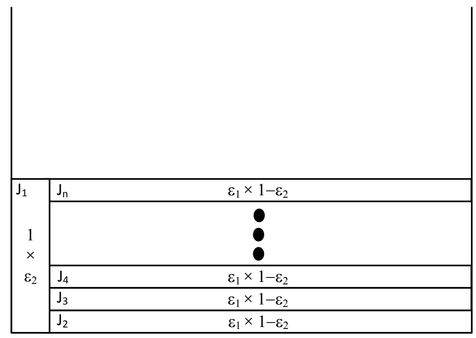
$$h_j \leq a_{j-1}/D, \qquad j > 1.$$

From this, we get

$$\begin{aligned} NFDH(pre) &= \sum_1^k h_j \\ &= h_1 + \sum_2^k h_j \\ &\leq h_1 + \sum_2^k a_{j-1}/D \end{aligned}$$



(a) Scheduling using *NFDHPreemptive*



(b) Optimal scheduling

Fig. 7: Example showing that the bound of Theorem 3 is tight. It consists of one job with dimension $(p \times t)$: $1 \times \epsilon_2$, and $n-1$ jobs of dimension $\epsilon_1 \times (1 - \epsilon_2)$ a) $NFDH(Pre) \to 2$ as $\epsilon_1, \epsilon_2 \to 0$ b) $Opt(Pre) = 1$

$$\begin{aligned} &< h_1 + A/D \\ &\leq h_1 + OPT(pre) \\ &= p_{\max} + OPT(pre). \end{aligned}$$

The tightness of the bound is established by the example of Figure 7. ■

When the power requirements of jobs are normalized so that $p_{\max} = 1$, the bound of Theorem 3 becomes $NFDH(pre) \leq OPT(pre) + 1$. By comparison, the corresponding asymptotic performance bound for $NFDHNonpreemptive$ when power requirements are normalized in this way is $NFDH(non) \leq 2OPT(non) + 1$ (Equation 1).

Theorem 4: For every instance of preemptive OCOSP, $NFDH(pre) < 2OPT(pre)$ and this bound is tight.

Proof: In the proof for Theorem 3, we showed that for every instance of preemptive OCOSP, $NFDH(pre) < h_1 + A/D = p_{\max} + A/D$. It is easy to see that $OPT(pre)$ $\geq \max\{p_{\max}, A/D\}$. Hence, $NFDH(pre)/OPT(pre) <$ $(p_{\max} + A/D)/\max\{p_{\max}, A/D\}$. When, $p_{\max} > A/D$, we get

$$NFDH(pre)/OPT(pre) < (p_{\max} + A/D)/p_{\max} = 1 + A/(Dp_{\max}) < 2.$$

Similarly, when $p_{\max} \leq A/D$, we get

$$NFDH(pre)/OPT(pre) < (p_{\max} + A/D)/(A/D) = p_{\max}/(A/D) + 1 \leq 2.$$

The tightness of the bound is established by the example of Figure 7. ■

We note that the bound of Theorem 4 has been independently established in [31], where NFDHPreemptive is referred to as the shelf algorithm.

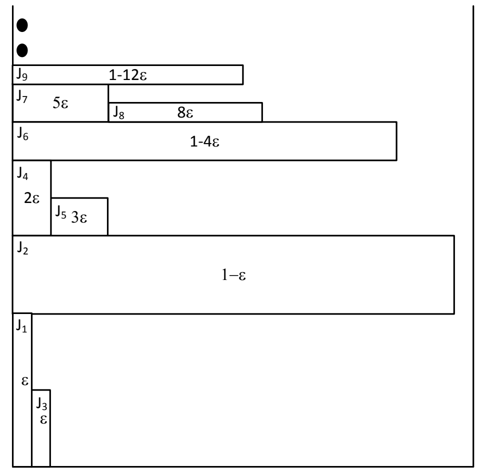
Theorem 5: For every instance of mixed OCOSP, $NFDH(mixed) < 2OPT(mixed) + p_{\max}$ and $NFDH(mixed) < 3OPT(mixed)$ and these bounds are tight.

Proof: The proof begins with the observation that $NFDH(mixed) \leq NFDH(non)$. In [19], it is shown that $NFDH(non) < p_{\max} + 2A/D$ (actually, [19] has $\leq$ in place of $<$ but it easy to see that the proof of [19] also holds with $<$). We observe that $OPT(mixed) \geq \max\{p_{\max}, A/D\}$. The asymptotic and absolute bounds of the current theorem are now easily established.

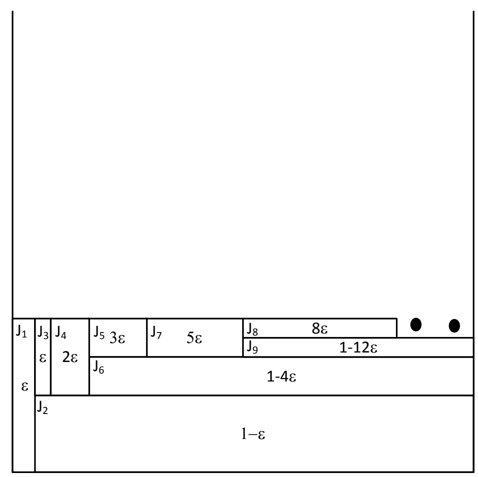
The example of Figure 3 of [19] established the tightness of the asymptotic performance bound for $NFDH(mixed)$. Although the tightness of the absolute bound of 3 for $NFDHNonpreemptive$ is known in the literature [19] an example showing this appears not to have been published. So we provide, in Figure 8 such an example, which also establishes the tightness of the absolute approximation bound for $NFDH(mixed)$. ■

VI. Experimental Evaluation

In section VI-A, we evaluate the performance of $NFDHPreemptive$ and $NFDHMixed$ using the benchmark data sets used in Ntene et al. [21] to evaluate algorithms for strip packing as well as our own data sets. These datasets represent power scheduling instances with random power demands and durations. In section VI-C, we evaluate them using a dataset that has only two different power demands and models the recharging of electric and plug-in hybrid electic vehicles in a charging garage.



(a) Sheduling using $NFDHMixed$



(b) Optimal Scheduling

Fig. 8: Example showing that the absolute approximation bound of Theorem 5 is tight. All jobs are non-preemptive and have dimensions $(p \times t)$ according to two series: i) $1 \times \epsilon, \frac{1}{2} \times \epsilon$, $\frac{1}{2} \times 2\epsilon, \frac{1}{4} \times 3\epsilon$, $\frac{1}{4} \times 5\epsilon$, $\frac{1}{8} \times 8\epsilon$ and so on, ii) $\frac{1}{2} \times (1-\epsilon)$, $\frac{1}{4} \times (1-4\epsilon)$, $\frac{1}{8} \times (1-12\epsilon)$ and so on. a) $NFDH(Mixed) = 1{+}1{+}\frac{1}{2}{+}\frac{1}{4}{+}... = 3$ b) $Opt(Mixed) = 1$

A. $NFDHPreemptive$ and $NFDHMixed$

The data sets used in [21] are described below.

- *Mumford-Valenzuela Benchmark* - Mumford-Valenzuela et al. [23] created two kinds of datasets: *Nice* and *Path*. The aspect ratios of the rectangles in *Nice* and *Path* are in the ranges $1/4 \leq Height/Width \leq 4$ and $1/100 \leq Height/Width \leq 100$, respectively. Additionally, the ratio of the areas of any two rectangles in a given dataset is less than or equal to 7 and 100 respectively for *Nice* and *Path*.
- *Hopper and Turton Benchmark* - The Hopper and Turton [24] benchmark comprises 7 subsets of 3 instances each. The number of rectangles is between 17 and 197. The rectangles were generated randomly so that their aspect ratio is at most 7.
- *Burke Benchmark* - The Burke [32] benchmark was created by cutting a large rectangle repeatedly. Burke et al. [32] generated 13 datasets of which 12 were used by us. The data is random in nature bounded only by the smallest dimension allowed for any rectangle.

We note that for the benchmark data used in [21] the optimal solutions are known. Table I gives the number of rectangles in the benchmark instances of [21] as well as the number of instances that have the same number of jobs.

In addition to the above benchmark data sets, we created five classes of data with the property that instances in the same class have the same number of rectangles. Each class comprises three subsets of instances with each subset containing 10 instances. The instances within a subset are characterized by a parameter γ, where γ is the fraction of rectangles in the instance whose width exceeds $D/2$. The values of γ used by us were 0.2, 0.6, and 0.8. Our instances had $n \in \{25, 50, 100, 200, 500\}$.

The rectangles in each of the test instances were mapped to jobs for scheduling by $NFDHPreemptive$ and $NFDHMixed$ by creating one job for each rectangle. The job's power requirement equals the height of the rectangle while the rectangle's width is the job's duration. For $NFDHPreemptive$, all jobs were declared to be preemptable while for $NFDHMixed$ jobs were declared preemptable with probability ρ. In our experiments, we used $\rho \in \{0.25, 0.50, 0.75\}$.

For each benchmark instance, we computed the performance ratio PR, which is defined to be the ratio of the maximum power demand of the schedule produced by the version of NFDH in use and that of the optimal schedule. We note that $PR \geq 1$, $PR = 1$ only for an optimal schedule, and smaller PR values are better than large ones. For instances with the same value of n, we computed the average PR as well as the standard deviation in the PR. For our data sets, we could not compute PR as the power demand of the optimal schedule is not known. Instead, we computed the improvement ratio IR, which is defined to be the ratio of the maximum power demand of the schedule created by $NFDHNonpreemptive$ and that of the schedule created by $NFDHPreemptive$ or $NFDHMixed$. We note that large IR values are desirable. Average IR values and standard deviations were computed as for PR.

The average PRs and the standard deviations are reported in Table I for the benchmark data used in [21] and the average IRs and standard deviations are reported in Table II for the data sets constructed by us. The weighted average of the IRs of Table I and the PRs of Table II are plotted in Figures 9 and 10, respectively. Note that *NFDHPreemptive* corresponds to the case when the percentage of preemptive jobs is 100% and *NFDHNonpreemptive* corresponds to the case when the percentage of preemptive jobs is 0%. As can be seen from the tables and plots, the PR reduced by up to 10% relative to $NFDHNonpreemptive$ when preemption was permitted and IR ratios as high 1.68 were achieved by $NFDHPreemptive$. The PR decreases almost linearly as the percentage of preemptve jobs increases and the IR increases as the probability ρ of a job being preemptable increases.

B. ILP for Preemptive Scheduling

We evaluated the effectiveness of our ILP for preemptive scheduling (Equations 4-7) in finding good schedules when used in conjunction with the CPLEX 12.5 ILP solver and a time bound of 150 seconds. We limited our experiments to the benchmark data sets by Hopper and Turton [24] and by Burke et al. [32]. Our experiments were conducted on a dual core Intel E2180 2.00GHZ PC with 3GB memory. We note that the

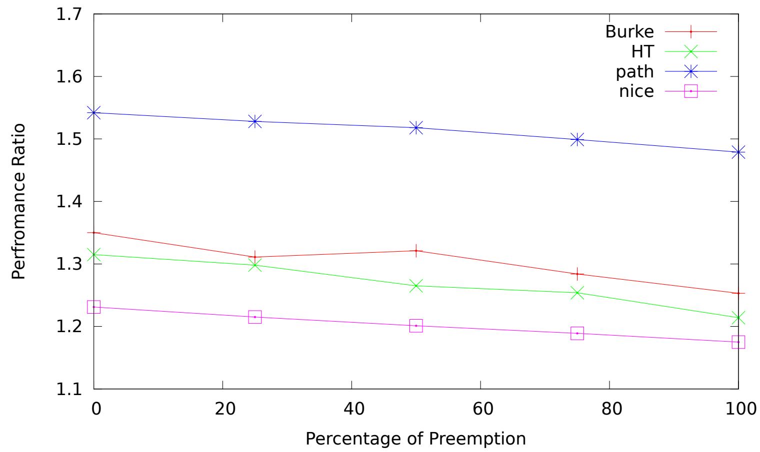
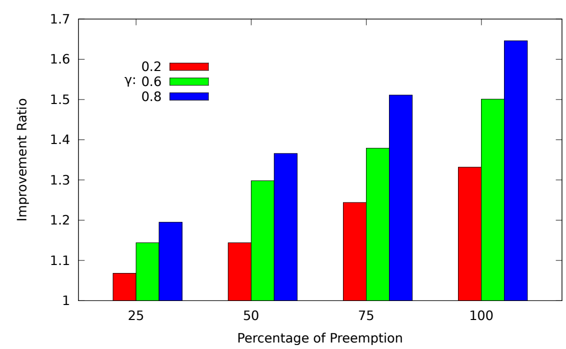
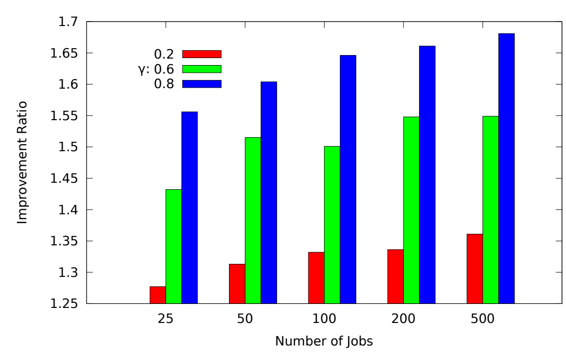


Fig. 9: Weighted average performance for benchmark data of [21]. (Smaller performance ratios indicate better performance)



(a) IR for 100 jobs



(b) IR for *NFDHPreemptive*

Fig. 10: Plots derived from Table II

TABLE I: Experimental results for benchmark data used in [21]

Benchmark	Total	Num of Jobs	$NFDHNonpreemptive$		$NFDHPreemptive$		$NFDHMixed$					
							0.25		0.50		0.75	
			PR	stddev	PR	stddev	PR	stddev	PR	stddev	PR	stddev
Burke	1	10	1.2	0	1.15	0	1.2	0	1.2	0	1.15	0
	2	20-30	1.29	0.184	1.14	0.057	1.25	0.127	1.2	0.141	1.26	0.226
	2	40-80	1.513	0.159	1.463	0.159	1.513	0.159	1.513	0.159	1.488	0.124
	3	50-70	1.377	0.253	1.293	0.27	1.34	0.262	1.35	0.269	1.31	0.299
	3	100-300	1.333	0.055	1.187	0.077	1.24	0.053	1.304	0.058	1.193	0.068
	1	500	1.267	0	1.237	0	1.263	0	1.267	0	1.253	0
HT	3	49	1.378	0.092	1.267	0.058	1.372	0.096	1.289	0.051	1.306	0.098
	3	73	1.289	0.107	1.156	0.038	1.274	0.097	1.204	0.034	1.189	0.022
	3	97	1.269	0.076	1.167	0.038	1.258	0.068	1.194	0.038	1.186	0.048
	3	196-197	1.277	0.092	1.21	0.063	1.254	0.079	1.246	0.063	1.227	0.044
	3	25	1.267	0.133	1.133	0.115	1.267	0.133	1.244	0.102	1.178	0.077
	3	16-17	1.417	0.176	1.333	0.076	1.35	0.087	1.367	0.104	1.4	0.15
	3	28-29	1.311	0.038	1.233	0.058	1.311	0.038	1.311	0.038	1.289	0.019
path	50	25	1.544	0.199	1.489	0.162	1.532	0.191	1.51	0.176	1.498	0.171
	50	50	1.568	0.16	1.486	0.136	1.554	0.155	1.529	0.148	1.518	0.155
	50	100	1.552	0.16	1.485	0.16	1.538	0.159	1.527	0.166	1.507	0.166
	50	200	1.525	0.146	1.47	0.157	1.508	0.148	1.5	0.152	1.49	0.157
	10	500	1.447	0.13	1.413	0.138	1.435	0.134	1.431	0.134	1.424	0.135
nice	50	25	1.336	0.082	1.267	0.079	1.315	0.088	1.297	0.082	1.284	0.077
	50	50	1.262	0.057	1.188	0.052	1.24	0.061	1.217	0.053	1.199	0.053
	50	100	1.203	0.04	1.15	0.038	1.188	0.037	1.179	0.033	1.168	0.035
	50	200	1.15	0.026	1.114	0.025	1.14	0.028	1.135	0.03	1.127	0.03
	10	500	1.101	0.01	1.073	0.012	1.092	0.012	1.081	0.011	1.078	0.011

time required by $NFDHPreemptive$ on these data sets was less than 0.02 second. For 13 of the 33 benchmark instances, the CPLEX solver terminated in less than 150 seconds finding (quite naturally) an optimal schedule. For the remaining 20 instances, the peak power demand of the constucted schedule was between 0.67% and 4.38% more than the optimal power demand, the average over the optimal was 0.96% and the standard deviation was 1.09.

Although the ILP forumlation did quite well on the benchmark data, which were comprised of instances that had at most 300 jobs, we note that this formulation does not scale to large instances that have (say) thousands of jobs.

C. Charging Hybrid and Electric Vehicles

In [17], Pedro and Guillermo Sanchez have analyzed the use of a consumption management control algorithm for an electric vehicle charging infrastructure for a mix of Plug-in Hybrid Electric Vehicles (PHEV) and Electric Vehicles (EV). They evaluated their algorithm using instances comprised of 50 vehicles; the ratio of the number of PHEV to EV vehicles in an instance was one of 0:100, 20:80, 40:60, 60:40, 80:20 and 100:0; the maximum battery charge for EVs was 23kWh and for PHEVs was 7.2 kWh; and the charging rate for EVs and PHEVs was 3.8kW and 2.2kW, respectively. We use a similar setup to evaluate the algorithms of this paper for an offline vehicle charging environment in which the vehicles to be charged become available at a preset start time and charging must complete D units later. In our experiments, we used a total duration (D) of 10 hours, which is sufficient to charge a fully drained EV or PHEV battery and corresponds to an overnight charge interval from (say) 8pm to 6am. To conform to the integrality constraint of our preemptive ILP, we defined a unit of time to be 0.1 hours. So, we set $D = 100$ (i.e., 10 hours) and the maximum time needed to charge an EV was set to 61 time units ($\lceil 23/3.8 * 10 \rceil$) and that for a PHEV to 33 units

TABLE II: Experimental results for data generated by us

n	γ	$NFDHPreemptive$		$NFDHMixed$					
				0.25		0.50		0.75	
		IR	stddev	IR	stddev	IR	stddev	IR	stddev
25	0.2	1.277	0.072	1.058	0.054	1.121	0.095	1.215	0.105
	0.6	1.432	0.09	1.086	0.068	1.236	0.075	1.34	0.109
	0.8	1.556	0.05	1.15	0.045	1.309	0.091	1.419	0.084
50	0.2	1.313	0.047	1.054	0.034	1.117	0.038	1.216	0.046
	0.6	1.515	0.055	1.166	0.055	1.343	0.057	1.43	0.067
	0.8	1.604	0.049	1.142	0.051	1.376	0.086	1.507	0.057
100	0.2	1.332	0.03	1.068	0.023	1.144	0.045	1.244	0.034
	0.6	1.501	0.033	1.144	0.034	1.298	0.053	1.379	0.046
	0.8	1.646	0.03	1.195	0.022	1.366	0.065	1.511	0.035
200	0.2	1.336	0.022	1.086	0.017	1.152	0.03	1.234	0.033
	0.6	1.548	0.019	1.159	0.032	1.318	0.039	1.431	0.023
	0.8	1.661	0.016	1.208	0.021	1.383	0.048	1.533	0.014
500	0.2	1.361	0.025	1.078	0.008	1.161	0.014	1.255	0.024
	0.6	1.549	0.017	1.162	0.017	1.308	0.026	1.426	0.023
	0.8	1.681	0.017	1.196	0.019	1.39	0.029	1.546	0.029

TABLE III: Experimental Results for hybrid and electric vehicles datasets with with Non-preemptive NFDH heuristic

PHEV:EV	Number of Vehicles						Average
	50	100	200	500	800	1000	
0:100	714.40	1497.20	2872.80	7493.60	11635.60	14850.40	6510.67
20:80	640.00	1307.20	2596.00	6479.60	10294.80	12517.20	5639.13
40:60	563.60	1076.00	2107.60	5459.20	8361.20	10532.40	4683.33
60:40	412.00	907.60	1724.00	4188.40	6996.80	8520.00	3791.47
80:20	311.60	618.80	1242.00	3161.60	5104.80	6328.80	2794.60
100:0	211.20	435.60	831.60	2125.20	3357.20	4210.80	1861.93
Average	475.47	973.73	1895.67	4817.93	7625.07	9493.27	

TABLE IV: Experimental Results for hybrid and electric vehicles datasets with Preemptive NFDH heuristic heuristics

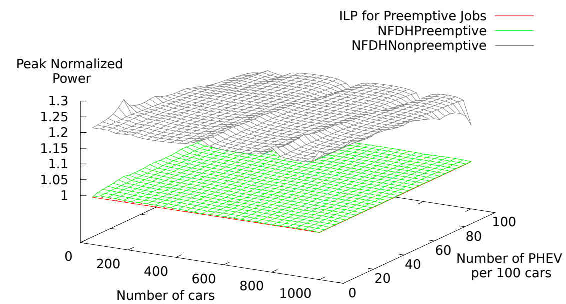
PHEV:EV	Number of Vehicles						Average
	50	100	200	500	800	1000	
0:100	585.20	1200.80	2325.60	5943.20	9241.60	11810.40	5184.47
20:80	512.80	1050.80	2110.40	5177.20	8169.20	10058.00	4513.07
40:60	465.60	891.20	1759.60	4383.20	6776.40	8494.40	3795.07
60:40	348.80	753.20	1412.80	3430.00	5749.60	6984.40	3113.13
80:20	274.40	531.60	1059.60	2678.40	4297.60	5375.20	2369.47
100:0	198.00	391.60	743.60	1918.40	3014.00	3770.80	1672.73
Average	397.47	803.20	1568.60	3921.73	6208.07	7748.87	

($\lceil 7.2/2.2*10 \rceil$). We also scaled the power required by EVs and PHEVs by 10 to 38 and 22, respectively. For our expriment, we generated instances with $n \in \{50, 100, 200, 500, 800, 1000\}$ vehicles. For each value of n we generated 5 instances for each of the 6 ratios for PHEV:EV used in [17]. Hence, we had a total of 180 instances. For each EV (PHEV) the required charge time was randomly selected to be between 1 and 61 (1 and 33). Note that in the worst case *NFDHPreemptive* will result in peak power at most twice the optimal solution. This can be inferred by the example in Figure 7, in which the power requirement for each job can be one of only two values.

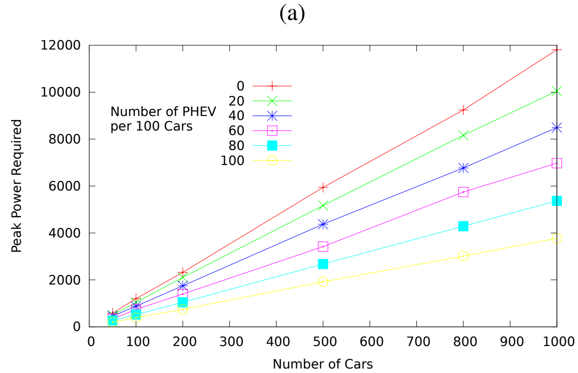
Table III-V gives the average peak power required by the charging schedule generated by $NonpreemptiveNFDH$, $PreemptiveNFDH$, and our preemptive ILP formulation (the ILP was limited to 150 seconds per instance) for each set of 5 instances with the same n (total number of vehicles) and same EV:PHEV ratio. Figure 11(a) plots the peak power required by the generated schedules normalized by the peak power required by the schedule generated by the preemptive ILP. The plots for ILP (constant normalized peak power value of 1) and $NFDHPreemptive$ are indistinguishable as these algorithms are very competitive. Figure 11(b) plots the peak power required by the schedules generated by our premptive ILP formulation. The *NFDHNonpreemptive* schedules required up to 26% more peak power than those generated by *NFDHPreemptive* thereby demontrating the effectiveness of preemption in an auto recharging application. Interestingly,

TABLE V: Experimental Results for hybrid and electric vehicles datasets with Preemptive ILP

PHEV:EV	Number of Vehicles						Average
	50	100	200	500	800	1000	
0:100	585.20	1200.80	2325.60	5943.20	9241.60	11810.40	5184.47
20:80	502.00	1040.00	2108.00	5167.60	8162.40	10050.00	4505.00
40:60	456.00	880.80	1747.60	4373.20	6769.20	8490.40	3786.20
60:40	340.40	746.00	1398.80	3418.80	5744.80	6974.40	3103.87
80:20	265.60	522.80	1047.20	2677.20	4290.00	5370.80	2362.27
100:0	198.00	391.60	743.60	1918.40	3014.00	3770.80	1672.73
Average	391.20	797.00	1561.80	3916.40	6203.67	7744.47	



(a)



(b)

Fig. 11: (a) Normalized power and (b) Peak power for preemptive ILP schedule

our ILP for preemptive scheduling, when run with a time limit of 150 seconds, performed better than *NFDHPreemptive* in a majority of the cases. The ILP gave better results for 92 of the tested 180 instances, gave the same result on another 75 instances, and required more peak power than the corresponding $NFDHPreemptive$ schedule on the remaining 13 instances. For the cases in which the ILP was better, the differences averaged 11.5 (or 1.15 kW) with a max value of 28 (or 2.8 kW) and the standard deviation of the difference was 6.5 (or 0.65 kW). For the 13 cases in which *NFDHPreemptive* did better, the maximum difference was 10 (or 1 kW) with an average of 4.9 (or 0.49 kW) and a standard deviation of 2.5 (or 0.25 kW). For 50, 100, 200, 500, 800 and 1000 cars, the ILP was better by up to 22, 28, 22, 22, 20, and 22 units, respectively. It is evident that the peak power required increases as the fraction of hybrid vehicles decreases.

Both *NFDHPreemptive* and *NFDHNonpreemptive* ran in less than 0.005 seconds per instance (on average). The ILP ran in less than 150 seconds for the cases in which the ratio PHEV:EV was either 100:0 or 0:100 and for almost all other cases, the threshold of 150 seconds was reached. We note that the ILP formulation for non-preemptive scheduling was unable to solve even our smallest instance with 50 vehicles and PHEV:EV ratio of 0:100. It ran out of memory after running for over 6 hours and the best result it had by that time (608 or 60.8kW) was about 5% more than the schedule generated by the preemptive ILP (570 or 57.0 kW), which took 0.16 seconds on this instance and obtained the optimal solution.

VII. Conclusion

The smart power grid supports two-way communication between power provider and consumers. Using this two-way communication, it is possible to schedule consumer requests so as to reduce peak power demand and hence reduce cost to both providers and consumers. In this paper, we have focussed on the preemptive and mixed offline scheduling of consumer requests for the case when all requests have the same start time and the same deadline. We have shown that preemptive scheduling can reduce peak power demand by up to 50% relative to optimal non-preemptive schedules. We have developed integer linear programming (ILP) formulations for the preemptive and mixed offline power scheduling problems and have extended the simplest of the heuristics proposed for 2D strip packing, next-fit decreasing height, to the offline preemptive and mixed power scheduling problems and derived tight absolute and asymptotic performance bounds. Although the next-fit decreasing height heuristic constructs preemptive schedules that have a peak power demand that is at most 50% more than that of optimal schedules for the benchmark datasets used by us, the worst-case (absolute) performance ratio for this heuristic is shown by us to be 2.

Although the preemptive ILP takes too much time to be useful on general instances with thousands of jobs, it outperforms $NFDHPreemptive$ when limited to 150 seconds/instance and instances have up to 1000 jobs with at most 2 different pewer requirements. So, it is quite practical as a power management tool for the overnight charging of vehicles in a reasonably large garage.

Finally, the bounds derived in this paper for preemptive scheduling using $NFDHPreemptive$ apply also to the preemptive extension of FFDH (first fit decreasing height).

References

[1] I. Koutsopoulos and L. Tassiulas, “Control and optimization meet the smart power grid: Scheduling of power demands for optimal energy management,” in *Proceedings of the 2Nd International Conference on Energy-Efficient Computing and Networking*, ser. e-Energy ’11. New York, NY, USA: ACM, 2011, pp. 41–50. [Online]. Available: http://doi.acm.org/10.1145/2318716.2318723

[2] Peaksaver program. [Online]. Available: https://www.peaksaver.com/peaksaver_THESL.html

[3] C. Gellings, "The concept of demand-side management for electric utilities," *Proceedings of the IEEE*, vol. 73, no. 10, pp. 1468–1470, 1985.

[4] A.-H. Mohsenian-Rad, V. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *Smart Grid, IEEE Transactions on*, vol. 1, no. 3, pp. 320–331, 2010.

[5] S. Dave, M. Sooriyabandara, and L. Zhang, "Application of a game-theoretic energy management algorithm in a hybrid predictive-adaptive scenario," in *Innovative Smart Grid Technologies (ISGT Europe), 2011 2nd IEEE PES International Conference and Exhibition on*, 2011, pp. 1–6.

[6] Z. Zhu, J. Tang, S. Lambotharan, W. Chin, and Z. Fan, "An integer linear programming and game theory based optimization for demand-side management in smart grid," in *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, 2011, pp. 1205–1210.

[7] Q. Zhu, Z. Han, and T. Basar, "A differential game approach to distributed demand side management in smart grid," in *Communications (ICC), 2012 IEEE International Conference on*, 2012, pp. 3345–3350.

[8] H. K. Nguyen, J. Song, and Z. Han, "Demand side management to reduce peak-to-average ratio using game theory in smart grid," in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, 2012, pp. 91–96.

[9] F. L. Meng and X. J. Zeng, "A stackelberg game approach to maximise electricity retailer's profit and minimse customers' bills for future smart grid," in *12th UK Workshop on Computational Intelligence (UKCI)*, 2012.

[10] X. Geng and P. Khargonekar, "Demand management of electric vehicle loads," in *American Control Conference (ACC), 2013*, 2013, pp. 5821–5826.

[11] F. Koyanagi and Y. Uriu, "Modeling power consumption by electric vehicles and its impact on power demand," in *Electrical Engineering in Japan*, vol. 120, no. 4, 1997, pp. 40–47.

[12] J. A. P. Lopes, F. J. Soares, P. M. Almeida, and M. Moreira da Silva, "Smart Charging Strategies for Electric Vehicles: Enhancing Grid Performance and Maximizing the Use of Variable Renewable Energy Resources," in *Proceedings 24th International Electric Vehicle Symposium and Exposition (EVS24)*, World Electric Vehicle Association (WEVA). World Electric Vehicle Association (WEVA), May 2009, pp. 1–11.

[13] L. Gan, U. Topcu, and S. Low, "Optimal decentralized protocol for electric vehicle charging," *Power Systems, IEEE Transactions on*, vol. 28, no. 2, pp. 940–951, 2013.

[14] Z. Ma, D. Callaway, and I. Hiskens, "Decentralized charging control for large populations of plug-in electric vehicles: Application of the nash certainty equivalence principle," in *Control Applications (CCA), 2010 IEEE International Conference on*, 2010, pp. 191–195.

[15] X. Yin, Z. Ma, and L. Dong, "Decentralized loading coordinations for large-population plug-in electric vehicles and a few controllable bulk loads," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, 2011, pp. 3092–3097.

[16] C. Ahn, C.-T. Li, and H. Peng, "Decentralized charging algorithm for electrified vehicles connected to smart grid," in *American Control Conference (ACC), 2011*, 2011, pp. 3924–3929.

[17] P. Sanchez-Martin and G. Sanchez, "Optimal electric vehicles consumption management at parking garages," in *PowerTech, 2011 IEEE Trondheim*, 2011, pp. 1–7.

[18] B. Baker, E. Coffman, Jr., and R. Rivest, "Orthogonal packings in two dimensions," *SIAM Journal on Computing*, vol. 9, no. 4, pp. 846–855, 1980. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/0209064

[19] E. Coffman, Jr., M. Garey, D. Johnson, and R. Tarjan, "Performance bounds for level-oriented two-dimensional packing algorithms," *SIAM Journal on Computing*, vol. 9, no. 4, pp. 808–826, 1980. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/0209062

[20] A. Lodi, S. Martello, and M. Monaci, "Two-dimensional packing problems: A survey," *European Journal of Operational Research*, vol. 141, no. 2, pp. 241 – 252, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221702001236

[21] N. Ntene and J. H. Van Vuuren, "A survey and comparison of guillotine heuristics for the 2d oriented offline strip packing problem," *Discret. Optim.*, vol. 6, no. 2, pp. 174–188, May 2009. [Online]. Available: http://dx.doi.org/10.1016/j.disopt.2008.11.002

[22] A. Steinberg, "A strip-packing algorithm with absolute performance bound 2," *SIAM J. Comput.*, vol. 26, no. 2, pp. 401–409, Apr. 1997. [Online]. Available: http://dx.doi.org/10.1137/S0097539793255801

[23] C. L. Mumford-Valenzuela, J. Vick, and P. Y. Wang, "Metaheuristics," M. G. C. Resende, J. P. de Sousa, and A. Viana, Eds. Norwell, MA, USA: Kluwer Academic Publishers, 2004, ch. Heuristics for Large Strip Packing Problems with Guillotine Patterns: An Empirical Study, pp. 501–522. [Online]. Available: http://dl.acm.org/citation.cfm?id=982409.982434

[24] E. Hopper and B. Turton, "An empirical investigation of meta-heuristic and heuristic algorithms for a 2d packing problem," *European Journal of Operational Research*, vol. 128, no. 1, pp. 34 – 57, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221799003574

[25] R. Harren and R. Stee, "Improved absolute approximation ratios for two-dimensional packing problems," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, ser. Lecture Notes in Computer Science, I. Dinur, K. Jansen, J. Naor, and J. Rolim, Eds. Springer Berlin Heidelberg, 2009, vol. 5687, pp. 177–189. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03685-9_14

[26] I. Schiermeyer, "Reverse-fit: A 2-optimal algorithm for packing rectangles," in *Algorithms ESA '94*, ser. Lecture Notes in Computer Science, J. Leeuwen, Ed. Springer Berlin Heidelberg, 1994, vol. 855, pp. 290–299. [Online]. Available: http://dx.doi.org/10.1007/BFb0049416

[27] C. Kenyon and E. Rémila, "A near-optimal solution to a two-dimensional cutting stock problem," *Math. Oper. Res.*, vol. 25, no. 4, pp. 645–656, Nov. 2000. [Online]. Available: http://dx.doi.org/10.1287/moor.25.4.645.12118

[28] R. Harren, K. Jansen, L. Prdel, and R. Stee, "A (5/3+)-approximation for strip packing," in *Algorithms and Data Structures*, ser. Lecture Notes in Computer Science, F. Dehne, J. Iacono, and J.-R. Sack, Eds. Springer Berlin Heidelberg, 2011, vol. 6844, pp. 475–487.

[29] K. Jansen and R. Solis-Oba, "Rectangle packing with one-dimensional resource augmentation," *Discret. Optim.*, vol. 6, no. 3, pp. 310–323, Aug. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.disopt.2009.04.001

[30] S. Alamdari, T. Biedl, T. Chan, E. Grant, K. Jampani, S. Keshav, A. Lubiw, and V. Pathak, "Smart-grid electricity allocation via strip packing with slicing," in *Algorithms and Data Structures*, ser. Lecture Notes in Computer Science, F. Dehne, R. Solis-Oba, and J.-R. Sack, Eds. Springer Berlin Heidelberg, 2013, vol. 8037, pp. 25–36. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40104-6_3

[31] ——. Strip packing with slicing. [Online]. Available: http://csclub.uwaterloo.ca/ egrant/Elyot Research Papers/Strip Packing with Slicing.pdf

[32] E. K. Burke, R. S. R. Hellier, G. Kendall, and G. Whitwell, "A New Placement Heuristic for the Orthogonal Stock-Cutting Problem," *Operations Research*, vol. 52, no. 4, pp. 655–671, Jul. 2004. [Online]. Available: http://www.extenza-eps.com/INF/doi/abs/10.1287/opre.1040.0109