Fast Approximations for Online Scheduling of Outpatient Procedure Centers

Bjorn Berg
George Mason University, bberg@gmu.edu

Brian Denton
University of Michigan, btdenton@umich.edu

June 18, 2014

Abstract

This paper presents a new model for online decision making. Motivated by the health care delivery application of dynamically allocating patients to procedure rooms in outpatient procedure centers, the online stochastic extensible bin packing problem is described. The objective is to minimize the combined costs of opening procedure rooms and utilizing overtime to complete a day's procedures. The dynamic patient allocation decisions are made in an uncertain environment where the number of patients scheduled and the procedure durations are not known in advance. The resulting optimization model's tractability focuses the paper's attention on approximation methods and a special case that is amenable to decomposition-based solution methods. Theoretical performance guarantees are presented for list-based approximation methods as well as an approximation that is common in practice where procedure rooms are reserved for patient groups in advance. Numerical results based on a real outpatient procedure center demonstrate the favorable results of the list-based approximations based on their average and worst case performances. Further, the numerical experiments show that the policy of reserving procedure rooms for patient groups in advance can perform very poorly. These results are contrary to common practice and favor alternative, and still easy-to-implement, policies.

1 Introduction

The bin packing problem is a well-studied problem and variants arise in many applications including operating room management (Van Houdenhoven et al. 2007), steel slab design (Dawande et al. 2004), and internet advertising (Adler et al. 2002). In this article we are motivated by applications involving the completion of activities (items) within a certain time window (bins). The applications we consider arise in health care delivery contexts. For example, the patient allocation process in an outpatient procedure center (OPC) in which two types of patients, *routine* and *urgent add-ons*, need to be allocated to procedure rooms. Routine patients are allocated to procedure rooms in advance, while the urgent add-on patients need to be allocated at a later time (e.g., the morning of the planned procedures).

Health care delivery applications prompt us to relax several assumptions in the classic bin packing problem. The first is the assumption that the number of items to be allocated is known *a priori*. In this generalization, referred to as the dynamic, or online, bin packing problem (Coffman Jr et al. 1983), items are sequentially allocated to bins following the decision of how many bins to open. The second assumption we relax is to allow for the size of the items to be stochastic to reflect possible uncertainty in service times. Finally, we relax the assumption that the size of the bins is fixed and we allow the size of the bins to be *extended*, i.e., that there is an option to increase the size of the bins at some (overtime) cost. Together, the relaxation of these assumptions lead to the formulation of a new model we refer to as the *dynamic extensible bin packing* (DEBP) problem.

In the problems we consider items are allocated sequentially to bins. Items to be allocated may arise one at a time or in *batches*. The size of the batches (and thus the total number of items) at each stage may not be known with certainty. The size of individual items also may not be known with certainty. This problem can be formulated as a multi-stage stochastic integer program. In the first stage a decision is made about how many bins to open, subject to a fixed cost per bin and an upper limit on the total number of available bins. In each subsequent stage allocation decisions must be made for the assignment of the current batch of items to bins. The size of each batch is not observed prior to each stage; only the probability distribution for the size of each batch is known in advance. Finally, in the last stage the sizes of the items are observed and overtime costs are incurred if the total size of items in a bin exceeds the bin size.

The most general version of this problem is intractable. Therefore, we focus on (a) a special case for which decomposition methods can achieve exact or near optimal solutions and (b) approximations, for the general case, for which the worst case performance ratio can be bounded above. The special case is a three-stage (two-batch) case of the problem. This is motivated by a problem commonly faced by OPC managers in which a batch of *routine* patients is scheduled in advance and a second batch of *add-on* patients arrives on short notice. In this problem, the size of the first batch is known with certainty and the size of the second batch is only observed following the allocation decisions of the first batch. Using non-anticipativity constraints we reformulate this as a two-stage stochastic integer program in which all the binary decisions appear in the first stage. This reformulation allows us to apply decomposition-based solution methods, which we compare for a collection of realistic test instances of the problem based on a Gastroenterology and Hepatology Clinic at Mayo Clinic in Rochester, MN. In addition to this special case we also test approximations for the more difficult problem in which there are more than two batches.

The remainder of this article is organized as follows: In the next section we discuss previous work related to extensible bin packing, online bin packing, and related research from health care settings. In Section 3, we introduce a multi-stage stochastic integer programming formulation for the DEBP problem and identify special cases that are important for health care delivery. In Section 4 we provide bounds on the optimal objective function value for the stochastic integer program and a worst case performance measure bound for a fast approximation method. In Section 5 we present numerical results in the context of a case study of a real OPC. Key insights are summarized in Section 6 along with future research directions.

2 Literature Review

Bin packing problems have been thoroughly examined in various contexts. Due to the combinatorial challenges associated with this NP-Hard problem, much of the focus has been on the development of bounds (Martello and Toth 1990a,b, Fekete and Schepers 2001) and approximation algorithms (Fernandez de La Vega and Lueker 1981, Karmarkar and Karp 1982). The literature reviewed in this section focuses on variants of the bin packing problem that are closely related to the problem considered in this article. However, the interested reader is

referred to a survey by Coffman et al. (1996) for further discussion.

The online bin packing problem received significant attention during the early study of bin packing problems. In the online bin packing problem, items are allocated to bins one at a time without knowledge of future items. Well-known approximation algorithms include *First Fit* (the next item is allocated to the first bin with sufficient remaining capacity), *Next Fit* (the next item is allocated to the same bin as the previous item, or a new bin is opened if capacity is insufficient), *Revised First Fit* (items are classified based on their size and *First Fit* is applied), and *Best Fit* (the next item is allocated to the bin with the least, yet still sufficient, capacity). Performance bounds have been derived for each of these approximations (Ullman 1971, Johnson 1974, Yao 1980). While improved bounds continued to be developed in the past decade, Seiden (2002) notes that attention to the online bin packing problem has diminished.

Online bin packing problems with stochastic item sizes have also been studied. For example, Bentley et al. (1984) analyzed common deterministic online bin packing algorithms, such as First Fit and Best Fit, for stochastic online bin packing. The authors analyzed the expected performance of the algorithms for certain distributions of item size. Hoffmann (1982) demonstrated online algorithms that are based on the distribution of the item sizes can be adjusted through a parameter to obtain a desired expected performance ratio as the number of items increases. Later studies focused on uncertainty in the item arrival process where queues of items develop as a result of limited resource availability. Stochastic online bin packing processes were initially motivated by problems in bandwidth packing (Coffman and Stolyar 2001) where traditional First Fit and Best Fit algorithms were analyzed in the stochastic process setting. Gamarnik and Squillante (2005) further analyzed the stability and performance of such algorithms focusing on the development of matrix-analytic solution methods. Shah and Tsitsiklis (2008) extended the work by Coffman and Stolyar (2001) and Gamarnik and Squillante (2005) focusing on the average queue size relative to an arrival process parameter.

In the *extensible* bin packing problem the size of the bins can be extended, if necessary, at a cost proportional to the size of the bin extension. This problem was initially considered by Dell'Olmo et al. (1998) where the longest processing time approximation was applied and a worst-case performance bound was developed for special cases. Later, Coffman and Lueker (2006) analyzed approximation algorithms for extensible bin packing and provided a

fully polynomial time asymptotic approximation scheme motivated by previous bin packing problem approximation schemes. In the online extensible bin packing problem developed by Ye and Zhang (2004), the authors considered special cases where bins are assumed to be of unequal size as well as the two and three-bin cases, and analyze algorithms and performance bounds. Closely related to the problem considered in this article is the work by Speranza and Tuza (1999) on online algorithms for machine scheduling with extendable work time. Based on a fixed (and known) number of machines, the authors provide worst-case analysis for online algorithms.

Previous work has applied bin packing models to health care settings. Dexter and Traub (2002) used a simulation model to evaluate heuristics for dynamically allocating surgeries to operating rooms for the purpose of maximizing operating room utilization. The *Earliest Start Time* heuristic allocates the next surgery to the operating room with the earliest available start time. The *Latest Start Time* heuristic allocates the next surgery to the operating room with the latest available start time such that the expected surgery duration is less than the available time remaining without using overtime. While the Earliest Start Time heuristic maximized operating room utilization, the Latest Start Time heuristic provided a more balanced workload across operating rooms.

Hans et al. (2008) considered the robust surgery loading problem where surgeries are allocated to operating rooms in order to maximize utilization while limiting the use of overtime. The problem was analyzed as an extensible bin packing problem with unequal bin sizes and item size uncertainty. The authors compare the performance of heuristics, including *First Fit* and *Longest Processing Time*, and conclude that improved operating room management would significantly increase operating room capacity without increasing overtime. The authors highlight the impact of collecting and using historical surgery durations in mitigating the effects of surgery duration variability.

Denton et al. (2010) developed a two-stage stochastic integer program for allocating surgeries to operating rooms. Similar to the problem studied in this article, the duration of surgeries are unknown; however, the problem formulation is for a static (not dynamic) allocation process in which the total number of items to be allocated to bins is known *a priori*. Bounds and valid inequalities were developed to improve the solution time for the two-stage stochastic integer program using the L-shaped method. A comparison of solution quality was

presented based on the optimal solution to the stochastic integer program, the *Longest Processing Time* heuristic, and the solution to a robust optimization model. The authors show that instances of their model corresponding to practical surgery scheduling problems can be solved in reasonable time frames, but the heuristic and the robust formulation obtain solutions that perform very well with much shorter computation time. Further, the heuristic has the benefit of being easy to implement; and the robust formulation has the ability to limit worst-case performance.

This article contributes to the literature in the following ways. First, as opposed to the above cited literature, we consider a new model where the decisions are made sequentially (online), item sizes are stochastic, the bins are extensible, and the number of bins to open is a decision variable. We are not aware of any work that combines these considerations. Second, we include uncertainty in the number of items to be allocated. While previous online bin packing studies often assume no prior information about item size or number of items, in practice it is often the case that historical data can be used for probability distributions for both item sizes and number of items to be allocated. Third, we provide an analysis of the model that yields insight into the optimal solution as well as special cases that commonly arise in health care delivery. Further, we analyze fast approximations that are applicable to large instances of the problem and we provide lower and upper bounds on the performance ratio of these approximations. Finally, we present a case study in which the model is populated using data from a real OPC, to evaluate its practical and managerial importance and implications.

3 Model Formulation

In the traditional bin packing problem there are two primary decisions. First, deciding the appropriate number of bins needed to accommodate a packing, or allocation, of a set of items; and second, deciding on the allocation of items to the bins so as to minimize the number of required bins. An extension of this in which bins are extensible can be stated as

$$\min_{x,y} \left\{ \sum_{j=1}^{m} x_j + c^v \sum_{j=1}^{m} \max\{0, \sum_{i=1}^{n} d_i y_{ij} - S x_j\} | \sum_{j=1}^{m} y_{ij} = 1 \ \forall i; \ x_j, y_{ij} \in \{0,1\} \ \forall j \right\}, \quad (1)$$

where $x \in \mathbb{B}^m$ and $y \in \mathbb{B}^{m \times n}$ are vectors of binary decision variables for which of m bins ($j = 1, ..., m$) to open and how to allocate n items ($i = 1, ..., n$) to bins, respectively, S represents the capacity of the bins, and $c^v \in \mathbb{R}^1_+$ is the per unit time cost of extending a bin. In this model, the number of items to be allocated as well as the size of each item, d_i, are known in advance.

In the DEBP problem, the number of items to be allocated is realized in discrete stages. At each stage, a random number of items is observed. The batch size, $b^t(\sigma^t)$ in stage $t = 1, ..., T$ is a discrete and finite random variable with probabilities $p^t_{\sigma^t}$ for each of the outcomes indexed by $\sigma^t = \emptyset, 0, ..., b^t_u$, where $\emptyset$ represent the termination of the arrival process and b^t_u is an upper limit on the batch size in stage t. The set of $b^t(\sigma^t)$ items that are to be allocated in stage t is represented by $I^t(\sigma^t) = \{0, ..., \sigma^t\}$. The size of each item i within a batch is a random variable, $d_i(\omega^t)$, where ω^t is an index of random scenarios for the collective item sizes at stage t. The DEBP problem can be represented as

$$\min_x \left\{ \sum_{j=1}^{m} c^f x_j + \mathcal{Q}^1(x) |\ x_j \in \{0,1\}\ \forall j \right\}, \tag{DEBP}$$

where $x \in \mathbb{B}^m$ is a vector of binary decisions, $x_1, ..., x_m$, about which of m bins to open at stage 0, $c^f \in \mathbb{R}^1_+$ is the cost of opening a bin, and $\mathcal{Q}^1(x)$ represents the expected *costs-to-go*, referred to as the stage 1 recourse function. This stage 1 recourse function, $\mathcal{Q}^1(x)$, can be written as

$$\min_{y^1} \ E_{\omega^1, \sigma^1} \left[(p^2_\emptyset) c^v \sum_{j=1}^{m} \max\{0, \sum_{i \in I^1(\sigma^1)} d_i(\omega^1) y^1_{ij}(\sigma^1) - S x_j\} \right] + (1 - p^2_\emptyset) \mathcal{Q}^2(y^1) \tag{2a}$$

$$s.t.$$

$$y^1_{ij}(\sigma^1) \leq x_j\ \forall j, \forall i \in I^1(\sigma^1) \tag{2b}$$

$$\sum_{j=1}^{m} y^1_{ij}(\sigma^1) = 1\ \forall i \in I^1(\sigma^1) \tag{2c}$$

$$y^1_{ij}(\sigma^1) \in \{0,1\}\ \forall j, i, \sigma^1, \tag{2d}$$

where $y^1 \in \mathbb{B}^{m \times b^1_u}$ is a vector of item allocation decisions in the first stage. We assume that $c^f \leq S c^v$ to avoid the trivial case where it is optimal to open a single bin. The expectation of extending the bins is taken over all scenarios, indexed by σ^1 and ω^1.

The value of the recourse function in (2a) is the sum of the expected cost of extending the bins in stage 1 if no more items are to be allocated, and the expected costs-to-go if further items are to be allocated. Constraints (2b) and (2c) require that the first stage items in scenario σ^1 be allocated to an open bin.

The recourse function for stage t, $\mathcal{Q}^t(y^{t-1})$, is written as

$$\min_{y^t} \; E_{\omega^t,\sigma^t}\left[(p_\emptyset^{t+1})c^v\sum_{j=1}^{m}\max\{0,\sum_{i\in\{I^k(\sigma^k)|k\le t\}}d_i(\omega^t)y_{ij}^t(\sigma^t)-Sx_j\}\right]+(1-p_\emptyset^{t+1})\mathcal{Q}^{t+1}(y^t) \tag{3a}$$

$$s.t.$$

$$y_{ij}^t(\sigma^t)\le x_j\;\forall j,i\in I^t(\sigma^t) \tag{3b}$$

$$\sum_{j=1}^{m}y_{ij}^t(\sigma^t)=1\;\forall i\in I^t(\sigma^t) \tag{3c}$$

$$y_{ij}^t(\sigma^t)=y_{ij}^{t-1}(\sigma^{t-1})\;\forall j,\forall i\in I^{t-1}(\sigma^{t-1}) \tag{3d}$$

$$y_{ij}^t(\sigma^t)\in\{0,1\}\;\forall j,i,\sigma^t, \tag{3e}$$

where $y^t\in\mathbb{B}^{m\times b_u^t}$ is the vector of item allocation decisions at stage t.

In the DEBP problem, allocation decisions from previous stages are fixed, which is enforced by linking constraints (3d) from stage $t-1$ to stage t. Finally, the recourse function for stage T, $\mathcal{Q}^T(y^{T-1})$, can be written as

$$\min_{y^T} \; E_{\omega^T,\sigma^T}\left[c^v\sum_{j=1}^{m}\max\{0,\sum_{i\in\{I^k(\sigma^k)|k\le T\}}d_i(\omega^T)y_{ij}^T(\sigma^T)-Sx_j\}\right] \tag{4a}$$

$$s.t.$$

$$y_{ij}^T(\sigma^T)\le x_j\;\forall j,i\in I^T(\sigma^T) \tag{4b}$$

$$\sum_{j=1}^{m}y_{ij}^T(\sigma^T)=1\;\forall i\in I^T(\sigma^T) \tag{4c}$$

$$y_{ij}^T(\sigma^T)=y_{ij}^{T-1}(\sigma^{T-1})\;\forall j,\forall i\in\{I^k(\sigma^k)|k=T-1\} \tag{4d}$$

$$y_{ij}^T(\sigma^T)\in\{0,1\}\;\forall j,i,\sigma^T. \tag{4e}$$

There are several notable differences between (DEBP) and the extensible bin packing

problem in (1). For example, the objective in (DEBP) is to minimize the total cost of opening bins and the expected cost extending their capacities. Whereas (1) is an offline problem, where all item information is known before allocation decisions are made, in (DEBP) items arrive sequentially in batches of an uncertain number of items, with uncertain sizes.

3.1 Special Case of an OPC

We emphasize a special case of the above formulation that is relevant to OPCs and hospital operating rooms. In this context, patients are allocated to procedure rooms in two batches. The first batch of patients are scheduled in advance (often 48-72 hours in advance) based on requests for a particular day of service that have arisen weeks or months in advance. The second batch is comprised of urgent add-on patients that arise during the 48-72 hours before the day of service. Thus, in this three-stage problem ($T = 3$), patients are allocated to procedure rooms in two batches. The first batch size is of size n. The size of the second batch, however, is uncertain. With respect to formulation (DEBP), this special case can be interpreted as the scenario resulting from $p^1_n = 1$ and $p^3_\emptyset = 1$ where the first stage batch is of size n with certainty and the arrival process terminates following the arrival of the second batch.

The two-batch problem is a three-stage stochastic integer program (referred to as 3SIP). In the first stage, decisions are made about which bins to open and how to allocate the first batch of items. In the second stage, the second batch of items is allocated to bins. Finally, in the third stage, bin extension decisions are made following the observation of the item sizes. In Appendix A we show how this special case of (DEBP) can be reformulated as a two-stage stochastic program with all the binary decisions in the first stage (2SIP).

4 Model Properties and Fast Approximations

In this section we present lower and upper bounds on the optimal solution for the DEBP problem. We then analyze worst case performance guarantees for approximations that are easy to implement in practice. The analysis of approximations is motivated by the fact that an instance of DEBP (in which $p^1_{b^1_u} = p^2_\emptyset = 1$ and the arrival process terminates after b^1_u items arrive in the first stage with certainty) corresponds to EBP which is known to be NP-Hard,

and thus DEBP is NP Hard as well (Denton et al. 2010). Proofs of the results in this section can be found in Appendix B.

We begin by presenting lower and upper bounds on the optimal value for DEBP, z^*. We first provide the following definition for the set of items in all batches. We let $\mathbb{I} = \cup_{t=1}^{T} I^t(\sigma^t)$ be the union of all item batches and the expected value of the total size of all items across all scenarios is denoted by

$$\alpha = E_{\sigma,\omega}\left[\sum_{i\in\mathbb{I}} d_i(\omega^t)\right].$$

Proposition 4.1 *If the random item sizes, d_i, have a finite support $d_i \in [d_i^L, d_i^U]$ on the item size distribution then the following are bounds on the optimal value of DEBP*

$$c^f\left\lceil \frac{\alpha}{S(1+\frac{c^f}{Sc^v})}\right\rceil \leq z^* \leq \frac{2c^f\sum_{i\in\mathbb{I}} d_i^U}{S(1+\frac{c^f}{Sc^v})}$$

These bounds can be used to fix decision variables prior to the solution process and to prune nodes in the branch-and-bound tree. Further, as we will discuss in more detail in the next subsection, the bounds can be extended in order to evaluate the performance of approximations.

4.1 Performance of Approximation Methods

We begin by presenting some characteristics that all approximations for DEBP have in common. Next, we present results for a particular approximation that is commonly implemented in practice at OPCs, which we refer to as the *reserved bin approximation*. We then present results for an approximation called *List* that allocates items online such that each item is allocated to the least loaded bin.

Definition The performance ratio (PR) of an approximation for a DEBP problem instance $\mathcal{I}$ is the ratio of the approximate objective value, $A(\mathcal{I})$, to the optimal objective value, $Opt(\mathcal{I})$.

Thus, bounds on a performance ratio PR^A for approximation A can be stated as

$$\min_{\mathcal{I}}\left\{\frac{A(\mathcal{I})}{Opt(\mathcal{I})}\right\} \leq PR^A \leq \max_{\mathcal{I}}\left\{\frac{A(\mathcal{I})}{Opt(\mathcal{I})}\right\}.$$

We begin by presenting the following lower bound on PR^A for any approximation, A.

Proposition 4.2 *No approximation method for* (DEBP) *can have a worst-case performance ratio smaller than* $1 + \frac{Sc^v}{6c^f}$.

Next, we consider the asymptotic performance of any *rational approximation*, which we define as any approximation where all nominal bin space is used prior to allocating an item to a bin by using extensible space exclusively. In this context the following proposition holds.

Proposition 4.3 *If there is a finite upper bound on the number of bins in* (DEBP) *then as* $\alpha \to \infty, PR^A \to 1$*, for any rational approximation.*

4.1.1 Performance of the Reserved Bin Approximation

In the reserved bin approximation, denoted by A^r, bins are reserved in advance for batches of items. This is a common practice in environments such as OPCs because of the simplicity it affords managers to plan for an uncertain number of future procedures. However, pre-assigning items to bins is a restriction that may be sub-optimal. An upper bound on the performance ratio for A^r can be obtained by considering the reserved bin problem as multiple separate extensible bin packing problems. In the following theorem let α^t denote the expected value of the sum of item sizes in the stage t batch.

Theorem 4.4 *As* $\alpha \to \infty$*, an upper bound on the performance ratio for the bin reservation approximation,* PR^{A^r}*, is* $1 + \frac{Sc^v}{c^f}$.

4.1.2 List-based Algorithms

Algorithm 1 describes *a-List*, an approximation for the determistic formulation (1) in Section 3, that allocates each item sequentially to the least utilized bin and iteratively increases the number of bins to determine how many bins, m, to select. Speranza and Tuza (1999) present a worst-case performance analysis of online list-based algorithms for a problem similar to DEBP. In this section we extend the results in (Speranza and Tuza 1999) by relaxing the assumption of a fixed and predetermined number of open bins and by differentiating the cost of opening a bin, c^f, and the cost of overtime, c^v.

We let $Opt(\mathcal{I})$ and $LS(\mathcal{I})$ denote the optimal and resulting *a-List* values, respectively. We present two lemmas that can be used to derive bounds on the performance ratio for Algorithm 1. The first lemma is adapted from Lemma 1 in Speranza and Tuza (1999) to account for varying cost parameters (c^v and c^f).

Algorithm 1: a-List approximation

Data: Set of items, $i = 1...n$, set of bins, $j = 1...m$, size of items, d_i, and costs c^v and c^f. L_j denotes the current load (total size of items) allocated to bin j.

Result: Number of bins and assignment of items to bins.

for $j = 1$ *to* m **do**
  $Cost = 0$ and $L_j = 0\ \forall j$
  for $i = 1$ *to* n **do**
    $\hat{j} = argmin_j\{L_j\}$
    $L_{\hat{j}} = L_{\hat{j}} + d_i$
  end
  for $k = 1$ *to* j **do**
    $Cost = Cost + c^v(max(L_k - S, 0))$
  end
  $TotalCost_j = Cost + c^f j$
end
Return $\min_j\{TotalCost_j\}$

Lemma 4.5 *Given a fixed number of bins, m, if $Opt(\mathcal{I}) = mc^f$ and $d_i \leq q$ for all i, then $LS(\mathcal{I}) \leq mc^f + \frac{mc^v q}{4}$.*

The second lemma provides the means to extend results for the performance ratio for the special case of a fixed number of open bins to the more general algorithm *a-List* that varies the number of open bins.

Lemma 4.6 *Suppose there exists a function, $\hat{\mathcal{Q}}^1()$, such that $\mathcal{Q}^1(x) \leq \hat{\mathcal{Q}}^1(x) \leq \beta\mathcal{Q}^1(x)$ for all x, where $\beta \geq 1$. Let $x^* = argmin\{\mathcal{Q}^1(x)\}$ and $\hat{x} = argmin\{\hat{\mathcal{Q}}^1(x)\}$. Then, $\mathcal{Q}^1(x^*) \leq \hat{\mathcal{Q}}^1(\hat{x}) \leq \beta\mathcal{Q}^1(x^*)$.*

Lemmas 4.5 and 4.6 can be used to derive the following bounds on the performance ratio for the *a-List* approximation.

Theorem 4.7 $1 + \frac{Sc^v}{6c^f} \leq PR^{a-List} \leq 1 + \frac{Sc^v}{4c^f}$.

The difference in the performance ratio lower and upper bounds is $\frac{Sc^v}{12c^f}$. For the special case of $c^f = Sc^v$ this correponds to a $\frac{13}{12}$ approximation guarantee. Consistent with intuition, this bound increases as the cost per unit time of overtime, c^v, increases.

Next, we describe a related algorithm for formulation (DEBP), in which the number of items and the item sizes may be uncertain. We assume the number of items is not known a priori and only the probability distribution for the item's size is known. We refer to the algorithm as *a(μ)-List*, to indicate items are allocated to the least utilized bin on the basis of

mean size. The pseudo-code for *a(μ)-List* is described in Algorithm 2. The number of items are indexed by scenario index, σ, and the size of the items by index ω.

Algorithm 2: a(μ)-List approximation

Data: Set of items, $i = 1...n$, set of bins, $j = 1...m$, size of item sizes, $d_i(\omega)$, mean item sizes according to an item's type, $\bar{d}_i$, and costs c^v and c^f. L_j denotes the current load (total size of items) allocated to bin j. The number of items and item sizes are indexed by $\sigma = 1, ..., T$ and $\omega = 1, ..., K$, with probabilities $p(\sigma)$ and $p(\omega)$, respectively.

Result: Number of bins to open.

for $j = 1$ *to* m **do**
  for $\sigma = 1$ *to* T **do**
    for $\omega = 1$ *to* K **do**
      $L_j = 0$ and $C_j = \emptyset \ \forall j$
      for $i = 1$ *to* $n(\sigma)$ **do**
        $\hat{j} = \arg\min_j \{L_j\}$
        $L_{\hat{j}} = L_{\hat{j}} + \bar{d}_i^{\,j}$
        Add i to C_j
      end
      $Cost = 0$
      for $k = 1$ *to* j **do**
        $Cost = Cost + c^v \left(\max \left(\sum_{i \in C_j} d_i(\omega), 0 \right) \right)$
      end
      $TotalCost_j(\sigma, \omega) = Cost + c^f j$
    end
    $TotalCost_j(\sigma) = \left(\sum_\omega p(\omega) TotalCost_j(\sigma, \omega) \right) / K$
  end
  $TotalCost_j = \left(\sum_\sigma p(\sigma) TotalCost_j(\sigma) \right) / T$
end
Return $\min_j \{TotalCost_j\}$

The following theorem presents lower and upper bounds for *a(μ)-List*, for the case of finite support defined by parameter $\theta \in [1, \infty)$.

Theorem 4.8

$$1 + \frac{Sc^v}{6c^f} \leq PR^{m(\mu)-List} \leq 1 + \frac{\theta Sc^v}{4c^f} + (\theta - 1)\frac{Sc^v}{c^f}.$$

As expected from Theorem 4.7, $PR^{m(\mu)-List} \to 1 + \frac{Sc^v}{4c^f}$ as $\theta \to 1$. Theorem 4.8 establishes the worst case performance of *a(μ)-List*. In the next section we provide estimates of the average case performance for typical instances arising in the context of OPCs.

5 Numerical Results

In this section we present numerical results based on practical instances of the model motivated by an OPC at Mayo Clinic in Rochester, MN. In this context, the decisions are how many procedure rooms to plan for, how routine procedures should be allocated to procedure rooms, and how to allocate urgent procedures that arise on short notice. Neither the number of urgent procedures nor the duration for each procedure is known in advance.

The remainder of this section is organized as follows. We begin by describing the parameter estimates used for generating problem instances for experiments. Three exact solution methods of the formulation discussed in Section 3.1 are compared based on their computational performance for a range of problem instances. Results from these methods are used to evaluate performance of the *a(μ)-List* approximation. Finally, some additional approximations are evaluated.

5.1 Design of Test Instances

Test instances were generated based on data from the Division of Gastroenterology and Hepatology at Mayo Clinic. The number of routine procedures, n, was varied between $n = 10$ and $n = 30$. The maximum number of urgent procedures, b_u^2, was varied between $b_u^2 = 0$ and $b_u^2 = 10$ with each scenario's probability, $p_{\sigma^2}^2$, being uniformly distributed. Procedure duration distributions and case mix were sampled from historical procedure data for the uncertain procedure duration parameters, $d_i(\omega)$. Figure 1 presents the procedure type mix for five procedures: Colonoscopy, Esophagogastroduodenoscopy (EGD), Endoscopic Retrograde Cholangiopancreatography (ERCP), Endoscopic Ultrasound (EUS) Colonoscopy, and EUS-EGD. Figure 2 presents the procedure duration density for the five procedure types. The ratio of variable overtime costs, c^v, to the fixed cost of opening a procedure room, c^f, was varied as the cost of opening a procedure room being equivalent to one (high), two (medium), and four (low) hours of overtime. The length of day, S, was set to 480 minutes. A total of 27 test instances resulted from varying the number of routine procedures, urgent procedures, and overtime costs.

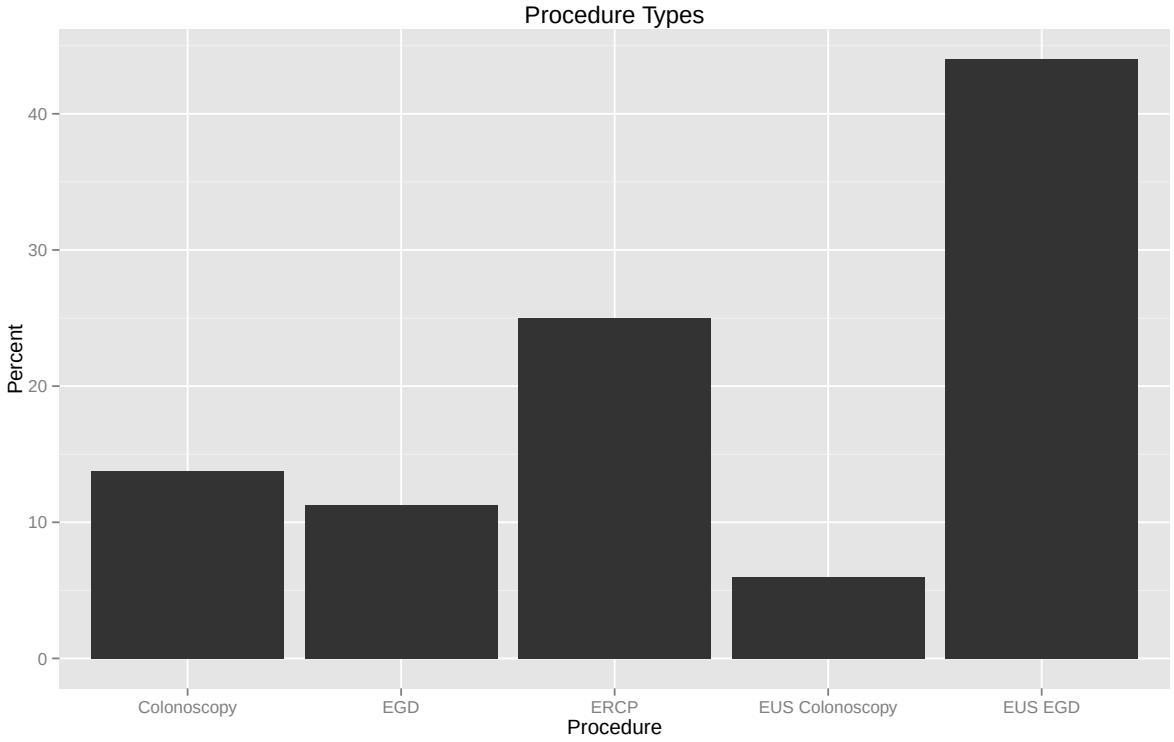

Figure 1: Procedure duration data for five types of procedures were used for model parameters.

5.2 Exact Solution Methods

The stochastic integer program based on the formulation 2SIP in Section 3.1 is difficult to solve due to binary decision variables in the first and second stages. Although the problem is naturally a three-stage decision process, we reformulate it as a two-stage stochastic integer program with all binary decisions in the first stage. The complete model formulation can be found in the appendix. Three solution methods were implemented: (1) traditional branch-and-bound implemented on the extensive formulation, (2) the traditional L-shaped method (Van Slyke and Wets 1969), and (3) the Integer L-shaped method (Laporte and Louveaux 1993). In the traditional L-shaped method, the recourse function is outer-linearized using optimality cuts based on the second stage dual solutions. In this method, the master problem is repeatedly solved with an optimality cut being added at each iteration. In the Integer L-shaped method the recourse function is outer-linearized incrementally at integer feasible nodes in the branch-and-bound tree.

Each problem instance was evaluated using ten random seeds with 1,000 random scenarios. A maximum of 15,000 CPU seconds was allowed for each instance. The computational results

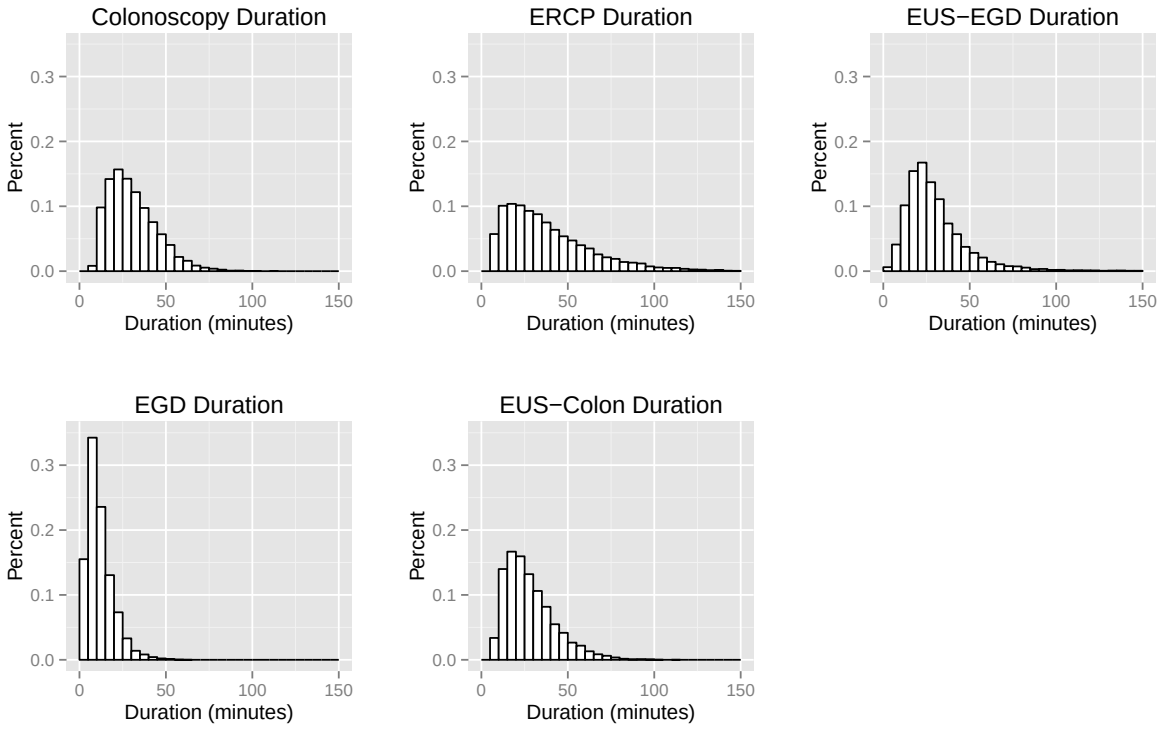

Figure 2: Historical data for procedure durations are based on data from the Divsion of Gastroenterology and Hepatology at Mayo Clinic.

for the solution methods are presented in an aggregated summary in Table 1. For each of the three methods, the average and max optimality gaps across the ten random seeds and 27 problem instances are reported. For problem instances that did not solve to an optimality tolerance of 1%, the optimality gap is calculated as a percentage of the lower bound at termination.

The results in Table 1 demonstrate that the DEBP is challenging. Branch-and-bound applied to the extensive formulation solved more problem instances to optimality than the L-shaped and Integer L-shaped methods (66.67% versus 48.15% and 44.44%, respectively). However, the L-shaped and Integer L-shaped methods were both competitive for harder instances solving more instances to within 10% of optimality (88.89% and 88.89% versus 74.07%, respectively). Further, the L-shaped method provided the lowest average optimality gap (3.21%) and the Integer L-shaped method provided the best worst-case performance optimality gap (29.38%).

The value of the stochastic solution (VSS) was also analyzed. The VSS is defined as the relative improvement over the expected value of using the mean value problem solution.

Method	% Optimal (<1%)	% Optimal (<10%)	Average Gap	Max Gap
Extensive Form	66.67%	74.07%	20.80%	288.05%
L-Shaped	48.15%	88.89%	3.21%	37.46%
Integer L-Shaped	44.44%	88.89%	4.10%	29.38%

Table 1: A summary of the performance of each solution method is aggregated over all of the problem instances.

The best known solution at termination was used in the VSS calculation when the solution methods did not converge to an optimal solution. Thus, the VSS analyzed is a lower bound. In general, the VSS is sensitive to the overtime cost estimates. As overtime costs estimates increase and the ratio $\frac{c^f}{c^v}$ decreases, the VSS increases to as high as 16.39%. However, the VSS for smaller instances is as low as 0%.

5.3 Evaluation of Approximation Methods

In this subsection we present results for the reserved bin approximation, and the *a-List* and *a(μ)-List* approximations. Finally, we present an evaluation of other related approximations.

Figure 3 presents the reserved bin approximation performance ratios for varying cost values of overtime as the maximum number of (uniformly distributed) urgent patients increases (the number of routine patients is fixed at $n = 10$). The general trend illustrated in Figure 3 is a decreasing performance ratio as the expected procedure duration sum increases. The results in Figure 3 suggest that dedicating a procedure room for urgent add-on procedures performs better as the expected demand increases, especially at high overtime cost settings. However, when the number of possible add-on patients is small, reserving a procedure room for these patients can result in poor performance. Figure 3 highlights the potential poor performance of the reserved bin approximation in practice. While this approximation is easy to implement, the need for better performing approximations is clear.

The *a-List* approximation was evaluated by comparing to optimal solutions to formulation (1). The performance ratio was calculated for 1,000 randomly generated problem instances ranging in size from 10 to 40. The optimal solution for each problem instance was obtained by solving the resulting extensible bin packing problem defined in (1). The average performance ratio for the *a-List* approximation, along with a 95% confidence interval, is presented in

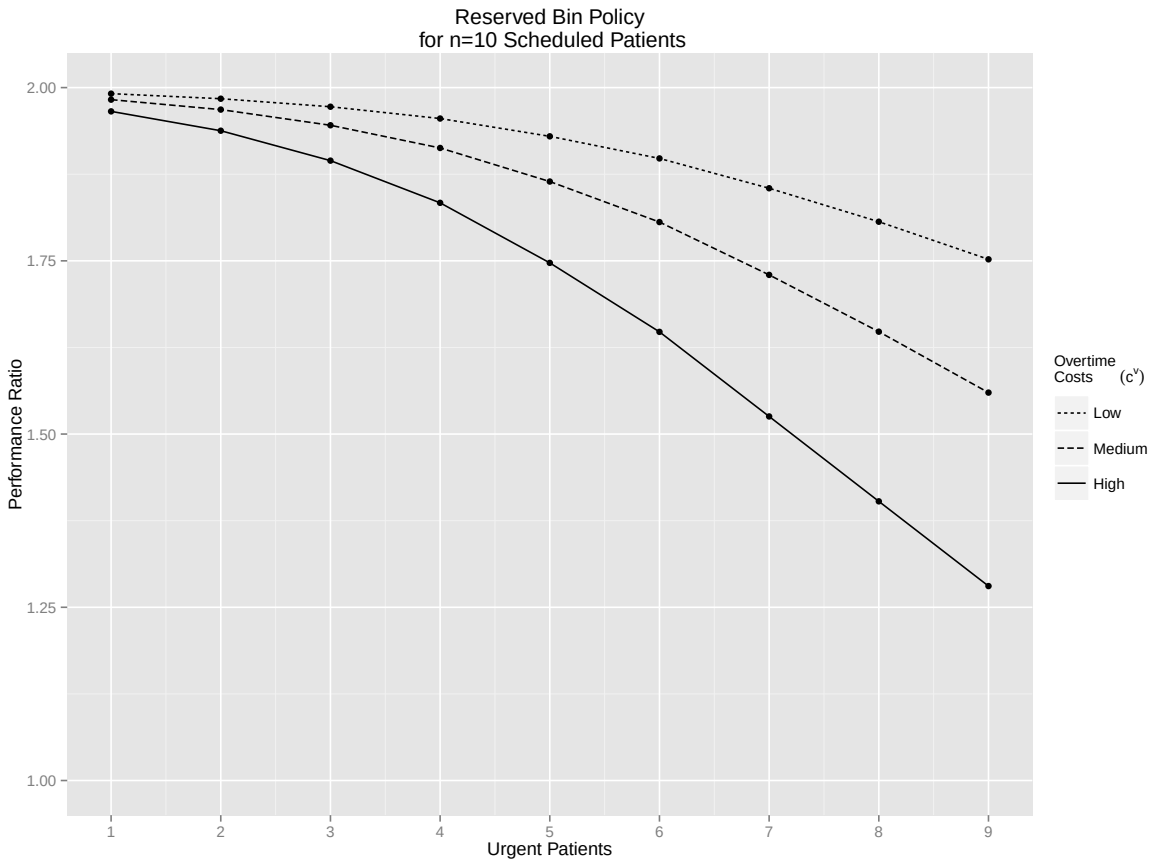

Figure 3: The performance ratio is presented for multiple overtime cost estimates and a range of maximum (uniformly distributed) urgent patients.

Figure 4 for the low, medium, and high estimates of overtime costs, c^v. The results show the *a-List* approximation performs very well with average performance ratios less than 1.025, and much lower than the theoretical worst case performance ratio of Theorem 4.7. The worst case performance from the numerical experiments are illustrated in Figure 5, the maximum performance ratio was 2.16, achieved when $\frac{c^v S}{c^f} = 8$. For lower $\frac{c^v S}{c^f}$ the worst case was typically less than 1.5.

The performance ratio of the *a(μ)-List* approximation was computed for all instances. The *a(μ)-List* approximation results are summarized in Figure 6. They are based on the best solution obtained within 15,000 CPU seconds for the results in Table 1. The worst case performance was less than 20%. In general, the performance ratio tends to increase as the size of the problem, urgent patient demand uncertainty, and overtime costs increase.

The reserved bin approximation was applied to a range of test instances the number of scheduled patients (10, 20, 30), the number of possible urgent patients (0-30), and overtime

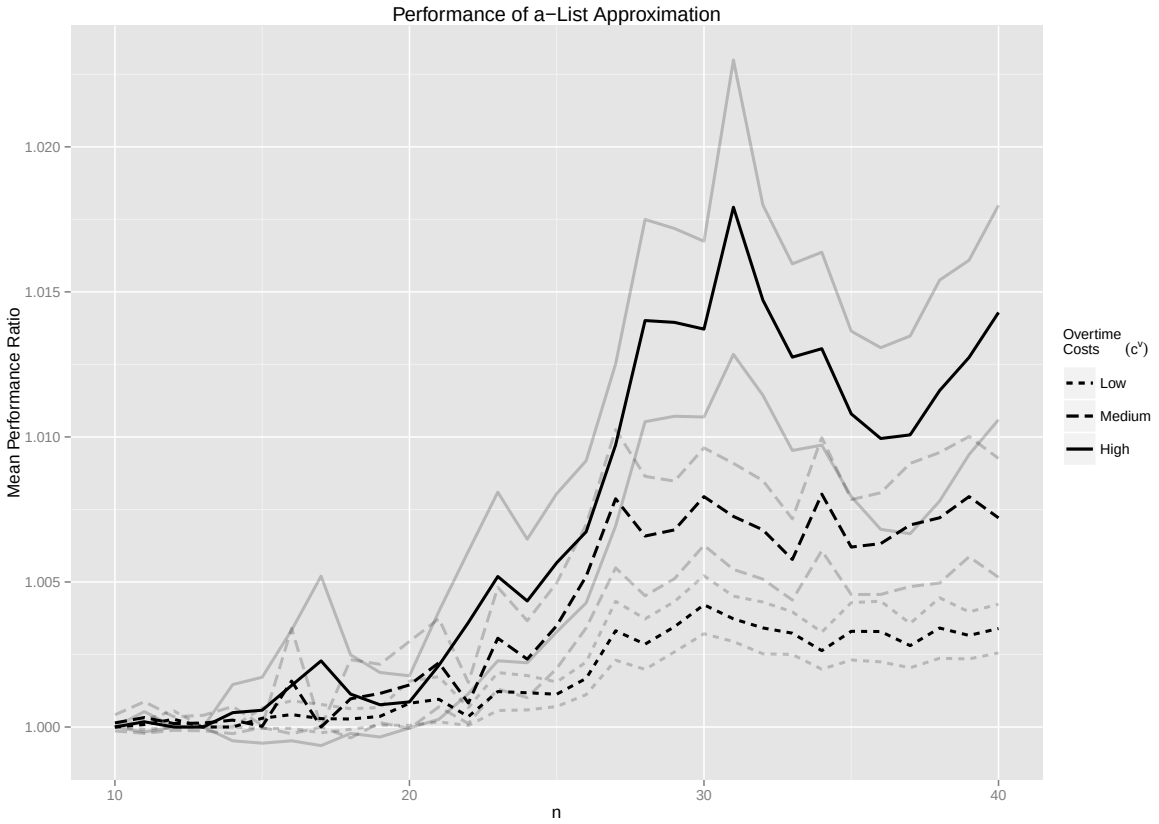

Figure 4: The average performance ratio is presented along with a 95% confidence intervals for low, medium, and high overtime cost estimates.

estimates (low, medium, high) were varied. Within each set of reserved bins (scheduled and urgent), *a-List* was applied. The number of bins for each patient group was identified by increasing the number of open bins until no overtime was used for the reserved bin and *a-List* approximations. The number of bins for each patient group that minimized the expected total costs was retained. Figure 7 presents the performance of applying *a-List* within the reserved bin approximation compared to applying *a-List* in a shared bin context (i.e., not reserved for each patient group). Having demonstrated the strong performance of *a-List*, we use its solution value for comparison as opposed to the optimal solution to the stochastic program in order to evaluate larger problem instances. The performance ratios in Figure 7 decrease as the number of urgent patients, and scheduled patients, increases. Sudden peaks are observed where the reserved bin approximation opens an additional bin and *a-List* is able to maintain more efficient use of share bins at the same demand level. Similar to Figure 3, the reserved bin approximation tends to perform better in instances where the overtime cost is high. Further, consistent with Theorem 4.4, the performance of the reserved bin approximation improves as

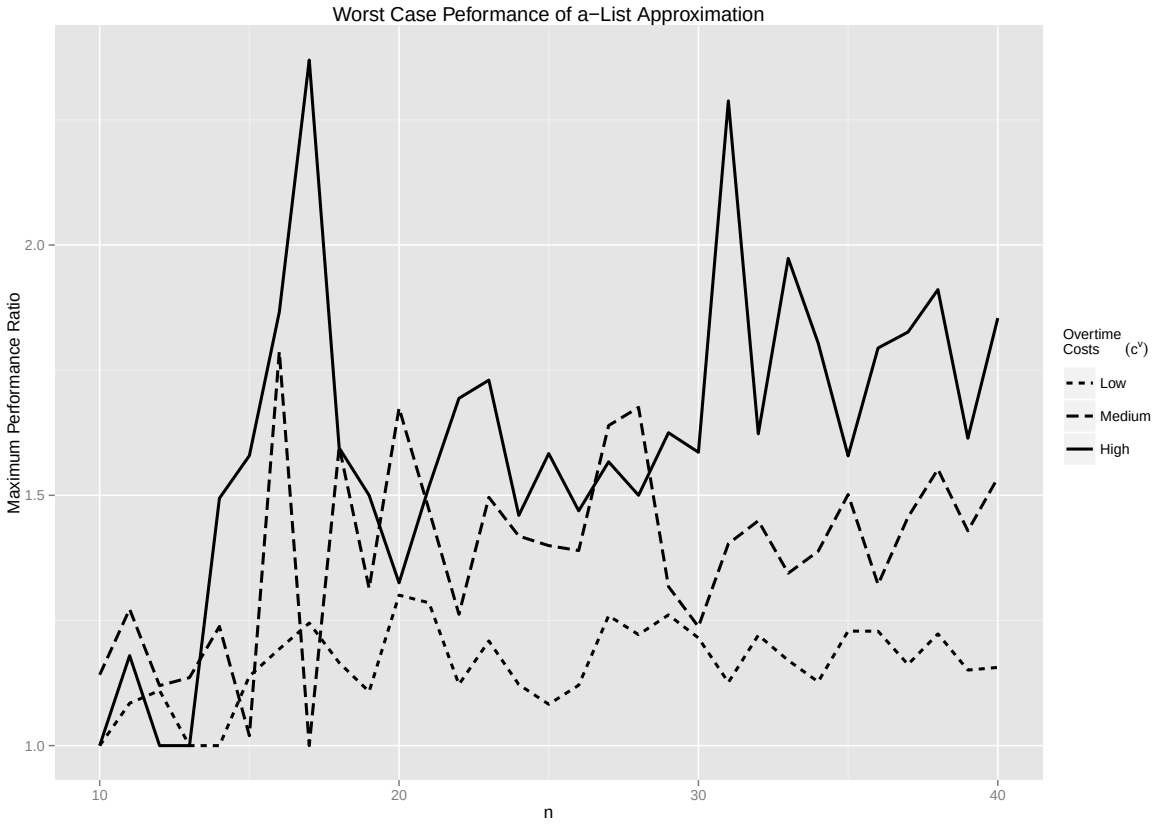

Figure 5: The maximum performance ratios across 1,000 randomly generated problem instances are presented for low, medium, and high overtime cost estimates.

the total expected procedure duration becomes large.

5.4 Other Approximations

In addition to the reserved bin, *a-List*, and *a(μ)-List* approximations, we tested some other allocation approximations motivated by related problems in the literature. Dexter and Traub (2002) evaluated two approximations for dynamically allocating elective surgeries to operating rooms with the objective of maximizing operating room utilization. Their *Earliest Start Time* (EST) and *Latest Start Time* (LST) approximations allocate the next requested surgery to the procedure room with the earliest and latest start time available, respectively. The LST approximation requires that there be sufficient capacity remaining for the surgery to be allocated to the room with the latest start time available. In our implementation, the scheduled procedures are sorted by increasing expected duration and the approximations are then applied. A variant of the Earliest Start Time approximation, *Earliest Start Time by Variance*

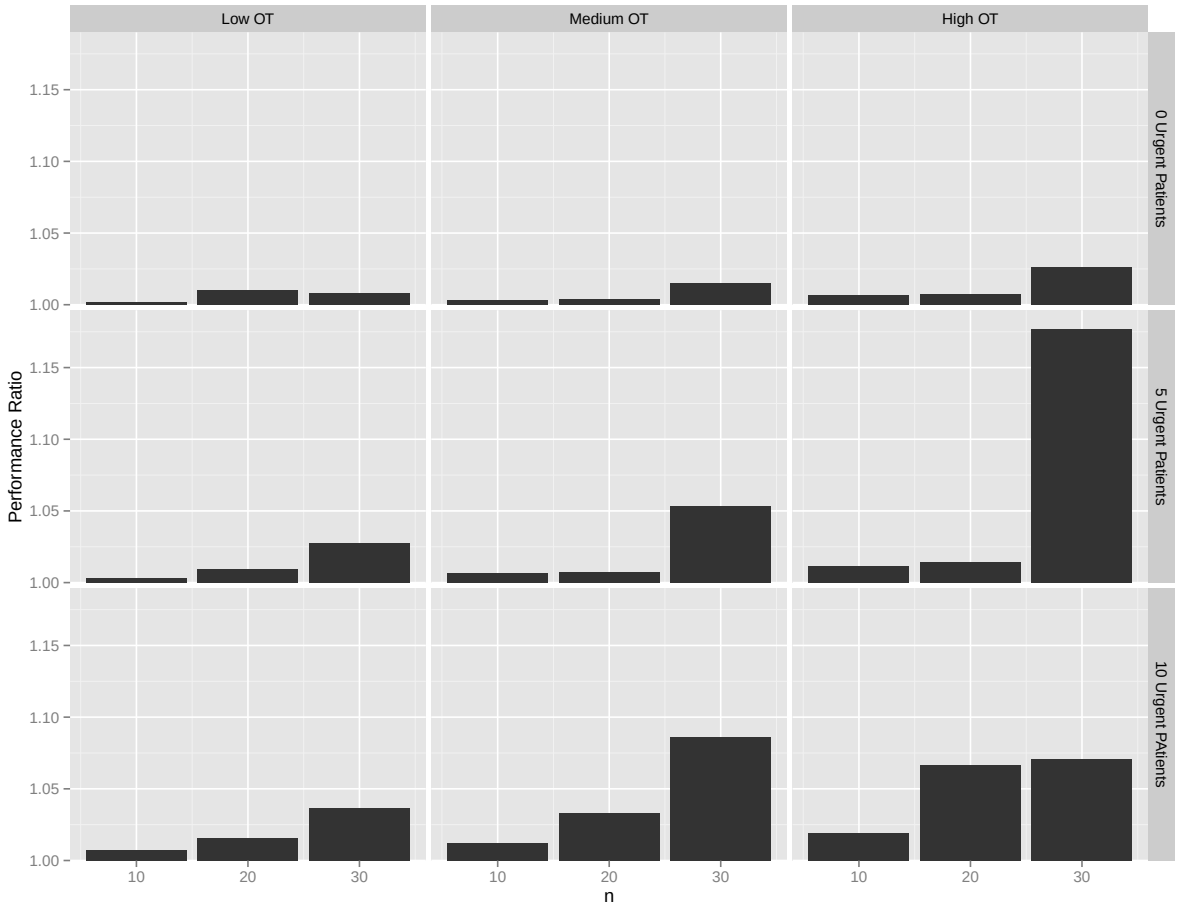

Figure 6: The *a(μ)-List* performance ratio is presented for multiple instances and is based on the results from the stochastic program.

(ESTV), was also evaluated. The ESTV approximation operates in the same manner except the scheduled procedures are sorted by increasing variance instead of expected procedure duration. Finally, the number of procedure rooms to open was decided through an enumeration process where the number of open procedure rooms was fixed and the approximations were applied. The number of open procedure rooms was then incremented and the approximations were again applied. This process continued until there was a solution for each approximation where no overtime was used, and the lowest cost solution was selected.

Using the test instances described in Section 5.2, the performance results for the three approximations are summarized in Table 2. We note that the optimal solutions for the $n = 10$ instances were identified by each of the approximations. In the instances where an optimal solution was not identified with the solution methods, the lower bound at termination was used in the approximation's optimality gap calculation. Thus, the summary of optimality

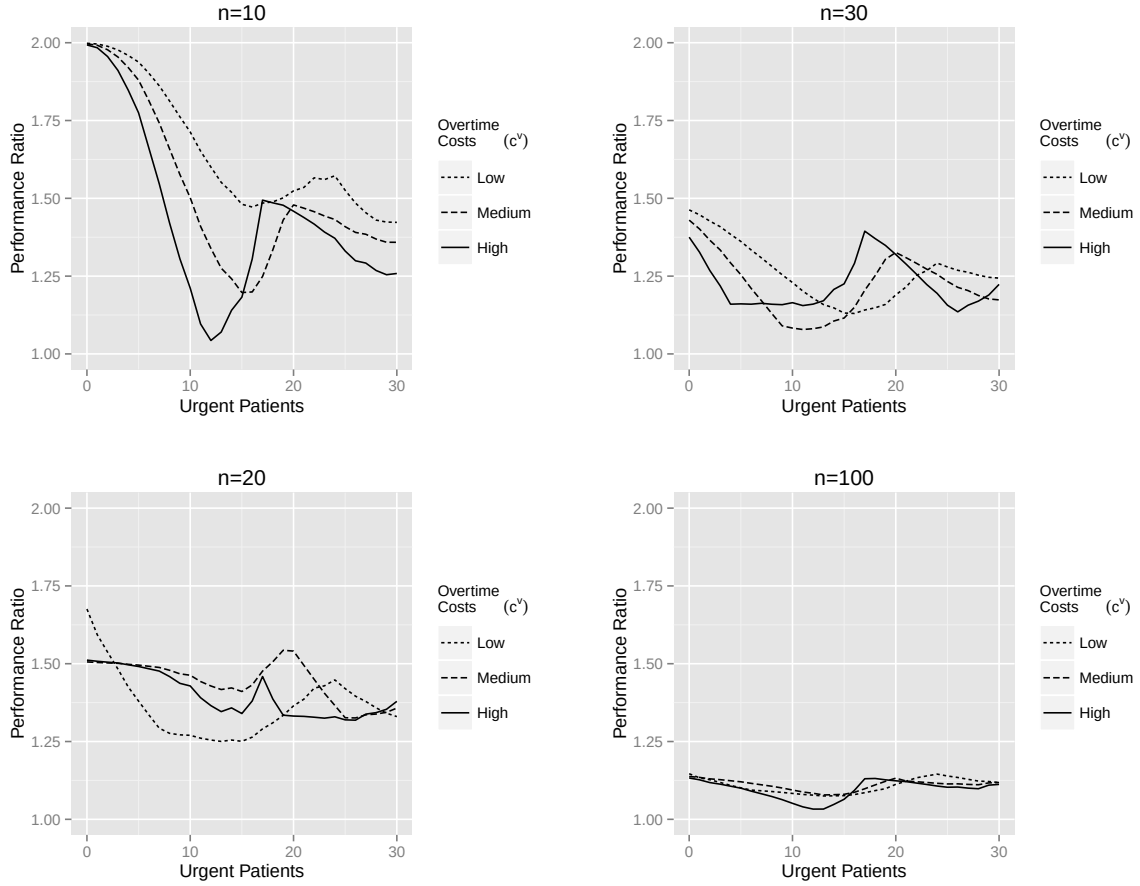

Figure 7: The performance ratio for the reserved bin approximation where *a-List* is applied within the reserved bins is presented for multiple instances and is based on comparing the results from applying *a-List* where bins are shared across patient groups. The horizontal axis represents the maximum number of uniformly distributed urgent patients.

gaps reported in Table 2 represent an upper bound on their performance.

Contrary to previous research, the LST approximations did not perform as well as the EST approximation. This is likely due to the semi-dynamic nature of the problem where in allocating scheduled procedures, the LST approximation will fill a single procedure room before it considers utilizing another procedure room. The ESTV approximation provided the optimal solution for 88.89% of the instances that were solved to optimality. This is in contrast to the LST and EST approximations producing optimal solutions for 66.67% and 83.33% of the instances, respectively. While all three approximations provided average optimality gaps less than 10%, EST and ESTV resulted in lower averages (1.61% and 1.58% versus 6.94%, respectively). Similarly, the two EST approximations both limited worst-case performance to

less than 10% while the LST approximation produced a worst case performance of 36.55% from optimal. Both the EST and the ESTV approximations performed well with small optimality gaps for larger problems with average optimality gaps between 0.69% and 6.99% for the $n = 30$ problem instances.

Approximation	**Within 1% of Optimal**	**Average Gap**	**Max Gap**
Latest Start	66.67%	6.94%	36.55%
Earliest Start	83.33 %	1.61%	7.75 %
Earliest Start by Variance	88.89%	1.58%	8.09%

Table 2: The approximations' performance is summarized and aggregated over all of the experiment instances.

All experiments for exact methods used CPLEX V12.2 on a Dell Linux server with 2 Quad-Core Intel Xeon E5420 2.5GHz CPUs and 16GB shared RAM.

6 Conclusions

In this article we presented a new model, referred to as the dynamic extensible bin packing problem (DEBP). Special cases of the model that are especially relevant to health care delivery applications were discussed. Lower and upper bounds were developed in order to provide insights into the performance of approximations as well as improve the computational efficiency of the described solution methods. Well-known approximations motivated by related problems in the literature were also described and evaluated.

The results presented demonstrate that the DEBP is a computationally challenging problem with some instances not being solved to optimality within 15,000 CPU seconds using decomposition-based solution methods. The *a-List* and *a(μ)-List* approximations were found to perform very well on average across a range of problem sizes and cost parameter estimates. While performance ratios for the *a-List* and *a(μ)-List* approximations were higher for larger test instances with high overtime cost estimates, the average performance ratios were less than 1.02 and 1.2, respectively. In extreme cases where the overtime cost was high, the *a-List* approximation had a worst case performance ratio greater than 2.

The performance of these approximations has implications on the use of probability distributions based on historical data for scheduling. For example, whereas the optimal solution to

the stochastic programming formulation relies on using historical data to populate the model, the approximation methods require only the number to be scheduled and expected procedure durations. In this scheduling context the value of using rich historical data is very low. We found that the reserved bin approximation, which is common in practice, can perform quite poorly. In contrast, other easy-to-implement approximations were shown, both theoretically and numerically, to perform very well across a wide range of test instances. We derived tight bounds on the performance ratio for the case of deterministic item sizes and extended these bounds to the stochastic case.

The VSS was shown to be sensitive to the estimate of overtime costs as well as the size of the problem. In larger problem instances the VSS was as high as 16% for problems where the overtime estimate was high and as low at 0% for problems where the overtime estimate was low. In many smaller problem instances, however, the solution to the semi-deterministic problem was optimal resulting in a VSS of 0%.

The performance of the approximations from the literature was generally robust with the optimal solution being identified for many instances, and small average and maximum optimality gaps otherwise. The Earliest Start Time by Variance provided the optimal solution most often as well as the lowest average optimality gap. Rules of thumb can easily be gleaned from these approximations in a manner that is easy for managers to implement. For example, delaying the scheduling process may afford the collection of information (e.g., procedure mix) that could be used in sorting the procedures and constructing a better schedule.

The DEBP model presented does not include all of the important aspects that may arise in OPC scheduling in practice. For example, certain procedures may require specific equipment or staff and need to be scheduled in particular procedure rooms. The constraints necessary for these settings can easily be added to the model formulation and approximation methods. Additionally, the setting of procedure start times is not addressed in the DEBP. However, any sequence reordering and appointment scheduling can be done without affecting the optimal solutions to DEBP. Finally, as we have shown, the approximation methods can yield poor solutions in rare instances; however, it is reasonable to assume those working most closely with the scheduling system will recognize these situations and adjust accordingly. Although our model does not address all conditions that may arise in practice, it provides a foundation for future study of the diverse range of operating conditions.

Acknowledgement

This material is also based in part upon work supported by the National Science Foundation (NSF) under Grant Number CMMI 0969885. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

Adler, M., P.B. Gibbons, Y. Matias. 2002. Scheduling space-sharing for internet advertising. *Journal of Scheduling* **5**(2) 103–119.

Bentley, J.L., D.S. Johnson, F.T. Leighton, C.C. McGeoch, L.A. McGeoch. 1984. Some unexpected expected behavior results for bin packing. *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. ACM, 279–288.

Coffman, E.G., M.R. Garey, D.S. Johnson. 1996. Approximation algorithms for bin packing: A survey. *Approximation algorithms for NP-hard problems*. PWS Publishing Co., 46–93.

Coffman, E.G., G.S. Lueker. 2006. Approximation algorithms for extensible bin packing. *Journal of Scheduling* **9**(1) 63–69.

Coffman, EG, AL Stolyar. 2001. Bandwidth packing. *Algorithmica* **29**(1) 70–88.

Coffman Jr, E.G., M.R. Garey, D.S. Johnson. 1983. Dynamic bin packing. *SIAM Journal on Computing* **12** 227.

Dawande, M., J. Kalagnanam, H.S. Lee, C. Reddy, S. Siegel, M. Trumbo. 2004. The slab-design problem in the steel industry. *Interfaces* 215–225.

Dell'Olmo, P., H. Kellerer, M.G. Speranza, Z. Tuza. 1998. A 13/12 approximation algorithm for bin packing with extendable bins. *Information Processing Letters* **65**(5) 229–233.

Denton, B. T., A. J. Miller, H. J. Balasubramanian, T. R. Huschka. 2010. Optimal allocation of surgery blocks to operating rooms under uncertainty. *Oper. Res.* **58**(4-Part-1) 802–816.

Dexter, F., R.D. Traub. 2002. How to schedule elective surgical cases into specific operating rooms to maximize the efficiency of use of operating room time. *Anesthesia & Analgesia* **94**(4) 933–942.

Fekete, S.P., J. Schepers. 2001. New classes of fast lower bounds for bin packing problems. *Mathematical programming* **91**(1) 11–31.

Fernandez de La Vega, W., G.S. Lueker. 1981. Bin packing can be solved within 1+ ε in linear time. *Combinatorica* **1**(4) 349–355.

Gamarnik, D., M.S. Squillante. 2005. Analysis of stochastic online bin packing processes. *Stochastic models* **21**(2-3) 401–425.

Hans, E., G. Wullink, M. Van Houdenhoven, G. Kazemier. 2008. Robust surgery loading. *European Journal of Operational Research* **185**(3) 1038–1050.

Hoffmann, U. 1982. A class of simple stochastic online bin packing algorithms. *Computing* **29**(3) 227–239.

Johnson, D.S. 1974. Fast algorithms for bin packing. *Journal of Computer and System Sciences* **8**(3) 272–314.

Karmarkar, N., R.M. Karp. 1982. An efficient approximation scheme for the one-dimensional bin-packing problem. *Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on*. IEEE, 312–320.

Laporte, G., F.V. Louveaux. 1993. The integer l-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters* **13**(3) 133–142.

Martello, S., P. Toth. 1990a. *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc.

Martello, S., P. Toth. 1990b. Lower bounds and reduction procedures for the bin packing problem. *Discrete Applied Mathematics* **28**(1) 59–70.

Seiden, S.S. 2002. On the online bin packing problem. *Journal of the ACM (JACM)* **49**(5) 640–671.

Shah, D., J.N. Tsitsiklis. 2008. Bin packing with queues. *Journal of Applied Probability* **45**(4) 922–939.

Speranza, MG, Zs Tuza. 1999. On-line approximation algorithms for scheduling tasks on identical machines with extendable working time. *Annals of Operations Research* **86** 491–506.

Ullman, JD. 1971. The performance of a memory allocation algorithm. Tech. Rep. 100, Princeton University.

Van Houdenhoven, M., J.M. Van Oostrum, E.W. Hans, G. Wullink, G. Kazemier. 2007. Improving operating room efficiency by applying bin-packing and portfolio techniques to surgical case scheduling. *Anesthesia & Analgesia* **105**(3) 707–714.

Van Slyke, R. M., R. Wets. 1969. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics* **17**(4) 638–663.

Yao, A.C.C. 1980. New algorithms for bin packing. *Journal of the ACM* **27**(2) 207–227.

Ye, D., G. Zhang. 2004. On-line extensible bin packing with unequal bin sizes. *Approximation and Online Algorithms* 355–356.

A 3SIP and 2SIP Formulations

The set of second stage batch items is indexed by k. The size of the second stage batch items in scenario ω is represented by $e_k(\omega)$. Because the number of second stage batch items that is observed in each scenario, σ^2, is less than or equal to b_u^2, not all of the b_u^2 possible items will necessarily be allocated to a bin, and thus a dummy bin is used (notated by $j = m+1$). The dummy bin, denoted by $j = m+1$, receives *non-present* second batch items. The presence of item k in scenario σ^2 is represented by the random parameter $A_k(\sigma^2)$; where $A_k(\sigma^2) = 1$ if item k is present in scenario σ^2, and $A_k(\sigma^2) = 0$ otherwise.

Finally, $\xi^2(\sigma^2)$ and $\xi^3(\omega)$ are random vectors containing second and third stage scenario dependent parameters, respectively ($\xi^2(\sigma^2) = \{A_1(\sigma^2), ..., A_{b_u^2}(\sigma^2)\}$ and $\xi^3(\omega) = \{d_1(\omega), ..., d_n(\omega), e_1(\omega), ..., e_{b_u^2}(\omega)\}$).

In the first stage there are two types of binary decisions that determine which bins are opened and how the first batch items are assigned to bins, x_j and y_{ij}^1, respectively. The first

stage problem for 3SIP is formulated as follows.

$$\min \sum_{j=1}^{m} c^f x_j + \mathcal{Q}^2(x, y^1) \tag{5a}$$

s.t.

$$y_{ij}^1 \leq x_j \qquad \forall (i, j = 1, \ldots, m) \tag{5b}$$

$$\sum_{j=1}^{m} y_{ij}^1 = 1 \qquad \forall (i) \tag{5c}$$

$$y_{ij}^1, x_j \in \{0, 1\} \qquad \forall (i, j) \tag{5d}$$

where

$$\mathcal{Q}^2(x, y^1) = E_{\xi^2}[Q^2(x, y^1, \xi^2(\sigma^2))] \tag{6}$$

represents the expected costs resulting from the first stage decisions and the second stage uncertainty. When the uncertainty regarding the size of the second batch is realized, second stage binary decisions are made allocating the second batch items to bins, $y_{kj}^2(\sigma^2)$. The second stage problem can be formulated as the following.

$$Q^2(x, y^1, \xi^2(\sigma^2)) = \min \mathcal{Q}^3(x, y^1, y^2) \tag{7a}$$

s.t.

$$y_{kj}^2(\sigma^2) \leq x_j \qquad \forall (k, j \leq m) \tag{7b}$$

$$y_{k,m+1}^2(\sigma^2) \leq 1 \tag{7c}$$

$$\sum_{j=1}^{m+1} y_{kj}^2(\sigma^2) = 1 \qquad \forall (k) \tag{7d}$$

$$A_k(\sigma^2) + y_{k,m+1}^2(\sigma^2) = 1 \qquad \forall (k) \tag{7e}$$

$$y_{kj}^2(\sigma^2) \in \{0, 1\} \qquad \forall (k, j) \tag{7f}$$

where

$$\mathcal{Q}^3(x, y^1, y^2) = E_{\xi^3}[Q^3(x, y^1, y^2, \xi^3(\omega))] \tag{8}$$

is the expected costs accrued from extending the bins resulting from the first and second stage

decisions and item size uncertainty, which can be written as

$$Q^3(x, y^1, y^2, \xi^3(\omega)) = \min \sum_{j=1}^{m} c^v o_j(\sigma^2, \omega) \tag{9a}$$

s.t.

$$\sum_{i=1}^{n} d_i(\omega) {y^1}_{ij} + \sum_{k=1}^{b_u^2} e_k(\omega) y_{kj}^2(\sigma^2) - o_j(\sigma^2, \omega) \leq S x_j \qquad \forall (j = 1, \ldots, m, \sigma^2) \tag{9b}$$

$$o_j(\sigma^2, \omega) \geq 0 \qquad \forall (j, \sigma^2) \tag{9c}$$

In constraint (7d), only the second batch items that are present in scenario σ^2 are allocated to open bins. If $A_k(\sigma^2) = 1$, then item k is assigned to a real bin. If $A_k(\sigma^2) = 0$, then item k is assigned to the dummy bin. The dummy bin is assumed to be open, but is not included in the fixed opening costs in the objective (5a). Finally, in the third stage the amount by which each bin needs to be extended is determined based on the previous stages' decisions and observations, and the amount by which the bins are extended is determined by the continuous decision variable, $o_j(\sigma^2, \omega)$.

Using the same notation as defined for 3SIP, the two-stage stochastic programming formulation (2SIP) is defined as follows.

$$\min \sum_{j=1}^{m} c^f x_j + \mathcal{Q}(x, y, y^2) \tag{10a}$$

s.t.

$$y_{ij}^1 \leq x_j \qquad \forall (i, j = 1, \ldots, m) \tag{10b}$$

$$\sum_{j=1}^{m} y_{ij}^1 = 1 \qquad \forall (i) \tag{10c}$$

$$y_{kj}^2(\sigma^2) \leq x_j \qquad \forall (k, j \leq m, \sigma^2) \tag{10d}$$

$$y_{k,m+1}^2(\sigma^2) \leq 1 \tag{10e}$$

$$\sum_{j=1}^{m+1} y_{kj}^2(\sigma^2) = 1 \qquad \forall (k, \sigma^2) \tag{10f}$$

$$A_k(\sigma^2) + y_{k,m+1}^2(\sigma^2) = 1 \qquad \forall (k, \sigma^2) \tag{10g}$$

$$y_{ij}^1, x_j, y_{kj}^2(\sigma^2) \in \{0, 1\} \qquad \forall (i, k, j, \sigma^2) \tag{10h}$$

where

$$\mathcal{Q}(x, y^1, y^2) = E_\xi[Q(x, y^1, y^2, \xi(\omega))] \tag{11}$$

represents the expected costs from allocating both first and second batch items in the first stage and the item size uncertainty that is subsequently realized. The second stage problem in 2SIP is equivalent to the third stage problem in 3SIP.

$$Q(x, y^1, y^2, \xi(\omega)) = \min \sum_{j=1}^{m} c^v o_j(\sigma^2, \omega) \tag{12a}$$

$s.t.$

$$\sum_{i=1}^{n} d_i(\omega) {y^1}_{ij} + \sum_{k=1}^{b_u^2} e_k(\omega) y_{kj}^2(\sigma^2) - o_j(\sigma^2, \omega) \leq S x_j \qquad \forall (j = 1, \ldots, m, \sigma^2) \tag{12b}$$

$$o_j(\sigma^2, \omega) \geq 0 \qquad \forall (j, \sigma^2) \tag{12c}$$

In 2SIP, the second stage of 3SIP is included in extensive form in the first stage by including constraints (10d)-(10g). While the first stage problem is a larger mixed integer program with additional binary decision variables that may be more difficult to solve than the first stage problem for 3SIP, the resulting second stage contains only continuous decision variables. Thus, the recourse function in the 2SIP is convex.

B Proofs

Proof of Proposition 4.1: The lower bound follows from applying Jensen's bound to the continuous relaxation of the problem in which fractional items can be allocated among bins and the fact that $c^f < Sc^v$, which implies that each bin will be allocated a total size of at most

$$S + \frac{c^f}{c^v}.$$

Given that the number of bins must be an integer, the lower bound can be rounded up to the next highest integer, completing the proof of the lower bound.

For the upper bound, $d_i(\omega) \leq d_i^U$ for all i and ω and thus the total item size can be, at

most, $\sum_{i\in\mathbb{I}} d_i^U$. Furthermore, any two bins having utilization less than or equal to

$$\frac{1}{2}(1+\frac{c^f}{c^v S}),$$

could be combined into a single bin at the same or a reduced total cost since $c^f \leq Sc^v$ (by assumption). $\square$

Proof of Proposition 4.2: Consider an instance of DEBP in which there is a single stage with $m = 2$, $n = 4$, and deterministic item sizes $\frac{1}{3}S$, $\frac{2}{3}S$, a, and b arriving in that order. Assume $a + b = \frac{4}{3}S$. an approximation may operate in two ways, either putting the first two items in one or two bins, which we denote as case (i) and (ii), respectively.

Case (i): Assume $a = b = \frac{2}{3}$. It follows that $A(\mathcal{I}) = 2c^f + \frac{Sc^v}{3}$ and therefore

$$\frac{A(\mathcal{I})}{Opt(\mathcal{I})} = \frac{2c^f + \frac{Sc^v}{3}}{2c^f} = 1 + \frac{Sc^v}{6c^f}.$$

Case (ii): Assume $a = S, b = \frac{1}{3}S$. It follows that $A(\mathcal{I}) = 2c^f + \frac{Sc^v}{3}$ and therefore

$$\frac{A(\mathcal{I})}{Opt(\mathcal{I})} = 1 + \frac{Sc^v}{6c^f}. \ \square$$

Proof of Proposition 4.3: Let m^U denote an upper bound on the number of bins. Since $\alpha > Sm^U$, any permutation of items to bins that uses the full amount of each bin's nominal space will result in the same objective function value. Thus, if the number of available bins is bounded, as in the case of a rational approximation, the item allocation decisions are made trivial by the system being overloaded. As $\alpha \to \infty$ and $\alpha > Sm^U$, the optimal and approximation solution value will both tend to $c^f m^U + c^v(\alpha - Sm^U)$. Thus, the performance ratio for any rational approximation in this case would tend to 1. $\square$

Proof of Theorem 4.4: For a given stage t batch of items and optimal solution y^{t*},

$$\begin{aligned} z^{t*} &\leq c^f + c^v E_{\omega^t}\left[\left(\sum_{i=1}^{n} y_{ij}^{t*} d_i(\omega^t) - S\right)^+\right] \\ &\leq c^f + c^v E_{\omega}^{t}\left[\left(\sum_{i=1}^{n} y_{ij}^{t*} d_i(\omega^t)\right)\right] \\ &= c^f + c^v \alpha^t. \end{aligned}$$

Summing over the stages, the following upper bound is obtained.

$$z^* \leq \sum_{t=1}^{T} \left(c^f + c^v \alpha^t \right) = Tc^f + c^v \alpha.$$

Using the lower bound in Proposition 4.1 we can write a bound on the performance ratio of the reserved bin approximation as

$$PR^{A^r} \leq \frac{Tc^f + c^v \alpha}{\left(\frac{c^f \alpha}{S(1+\frac{c^f}{Sc^v})} \right)} = \frac{(Sc^v + c^f)(Tc^f + c^v \alpha)}{c^f c^v \alpha}.$$

Thus, as $\alpha \to \infty$, $PR^{A^r} \leq 1 + \frac{Sc^v}{c^f}$. $\square$

Proof of Lemma 4.5: Assume that $\sum_{i=1}^{n} d_i = mS$. If this is not the case then small items can be added to ensure this without changing the optimal solution. In this case, the *List* approximation generates a solution for which each bin has at most one item that produces overtime. Let o_j be the overtime associated with bin j and s_j be the idle time when the last assigned item that causes overtime, if any. For the p bins with overtime, we can write

$$\sum_{j=1}^{p} o_j = \sum_{j=p+1}^{m} s_j.$$

Since the *a-List* approximation assigns each new item to the bin with the smallest load,

$$\frac{\sum_{j=1}^{p} s_j}{p} \geq \frac{\sum_{j=p+1}^{m} s_j}{m-p}.$$

In other words, the average idle time, once overtime causing items are removed, is higher for bins $1, ..., p$ than for $p+1, ..., m$. Therefore,

$$(m-p) \sum_{j=1}^{p} s_j \geq p \sum_{j=p+1}^{m} s_j = p \sum_{j=1}^{p} o_j.$$

Adding $(m-p) \sum_{j=1}^{p} o_j$ to both sides yields

$$(m-p) \sum_{j=1}^{p} (o_j + s_j) \geq m \sum_{j=1}^{p} o_j.$$

Because $(m-p)\sum_{j=1}^{p}(o_j+s_j) \leq (m-p)pq$ and $(m-p)p \leq (\frac{m}{2})^2$, it follows that

$$m\sum_{j=1}^{m} o_j \leq (\frac{m}{2})^2 q.$$

Thus, $LS(\mathcal{I}) \leq mc^f + \frac{mc^v q}{4}$. $\square$

Proof of Lemma 4.6: The left inequality follows from $\hat{\mathcal{Q}}^1()$ being an upper bounding function. The right inequality follows because $\hat{\mathcal{Q}}^1(\hat{x}) \leq \hat{\mathcal{Q}}^1(x^*) \leq \beta\mathcal{Q}^1(x^*)$. $\square$

Proof of Theorem 4.7: The lower bound is obtained from Proposition 4.2. For the upper bound, the following cases are enumerated:

(a) $\sum_{i=1}^{n} d_i = mS$

(b) $\sum_{i=1}^{n} d_i < mS$

(c) $\sum_{i=1}^{n} d_i > mS$

Case (a): Since $Opt(\mathcal{I}) \geq mc^f$ from Lemma 4.5,

$$\frac{LS(\mathcal{I})}{Opt(\mathcal{I})} \leq \frac{mc^f + \frac{mc^v S}{4}}{mc^f} = 1 + \frac{Sc^v}{4c^f}.$$

Case (b): Let $\sum_{i=1}^{n} d_i = mS - \bar{d}$, $\bar{d} > 0$. Create a new instance, $\mathcal{I}'$, that corresponds to case (a) by adding items with sizes summering to $\bar{d}$ which fit in the idle times available in the optimal solution. It follows that

$$\frac{LS(\mathcal{I})}{Opt(\mathcal{I})} = \frac{LS(\mathcal{I})}{Opt(\mathcal{I}')} \leq \frac{LS(\mathcal{I}')}{Opt(\mathcal{I}')} \leq 1 + \frac{Sc^v}{4c^f}.$$

Case (c): Let $\sum_{i=1}^{n} d_i = mS + \bar{d}$, $\bar{d} > 0$. Cut the sequence of items at the point where the sum is mS. This may divide an item into two parts. Let the set of items summing to mS be items $1, ..., k+1$. The instance $\mathcal{I}'$ based on these items corresponds to case (a). It follows that $LS(\mathcal{I}') \leq mc^f + \frac{mSc^v}{4}$. Consider the remaining items $k+2, ..., n$. Item $k+2$ is attached to item $k+1$ and the rest are assigned according to $List$. Thus, for the complete instance, $LS(\mathcal{I}) \leq mc^f + \frac{mSc^v}{4} + c^v\bar{d} = (1 + \frac{Sc^v}{4c^f})4mc^f + c^v\bar{d} \leq (1 + \frac{Sc^v}{4c^f})(mc^f + c^v\bar{d})$. Since $Opt(\mathcal{I}) \geq mc^f + c^v\bar{d}$, it follows that

$$\frac{LS(\mathcal{I})}{Opt(\mathcal{I})} \leq 1 + \frac{Sc^v}{4c^f}.$$

Applying Lemma 2 yields the final result when m is not fixed. $\square$

Proof of Theorem 4.8: Let $\bar{D}_j$ denote the mean of the sum of the item sizes for items allocated to bin j by *List*. Assume that $d_i(\omega)$ have a finite support such that $\sum_{i=1}^{n} y_{ij} d_i(\omega) \leq \theta \bar{D}_j$ where y_{ij} allocation decisions are determined by *List*.

The lower bound follows from Proposition 4.2 which corresponds to $\theta = 1$ where $d_i(\omega)$ are set to their mean values for all i and ω.

For the upper bound assume there are m bins. The expected overtime can be written as

$$\begin{aligned} E_\omega \left[\sum_{j=1}^{m} o_j(\omega) \right] &\leq \sum_{j=1}^{m} (\theta \bar{D}_j - S)^+ \\ &= \sum_{j=1}^{m} \theta \left(\bar{D}_j - \frac{S}{\theta} \right)^+ \\ &= \sum_{j=1}^{m} \theta \left(\bar{D}_j - \frac{S}{\theta} + S - S \right)^+ \\ &= \sum_{j=1}^{m} \left(\theta (\bar{D}_j - S)^+ + \theta \left(S - \frac{S}{\theta} \right) \right) \\ &\leq \frac{m\theta S}{4} + (\theta - 1) m S. \end{aligned}$$

where the first inequality follows from the assumption that $\sum_{i=1}^{n} y_{ij} d_i(\omega) \leq \theta \bar{D}_j$ for all ω. The third equality results from $(S - \frac{S}{\theta}) \geq 0$. The final inequality follows because $m(\bar{D}_j - S)^+ \leq \frac{mS}{4}$ by Theorem 3 and $\theta(S - \frac{S}{\theta}) = S(\theta - 1)$. Thus, given the number of bins is fixed to m and the unit overtime cost is c^v, $m(\mu) - List(\mathcal{I}) \leq mc^f + \frac{mc^v \theta S}{4} + (\theta - 1) mc^v S$. Dividing by mc^f (the value where no overtime is used) yields the stated upper bound.

Applying Lemma 4.6 yields the final result when m is not fixed. $\square$