# WORST-CASE ANALYSIS OF HEURISTICS FOR THE BIN PACKING PROBLEM WITH GENERAL COST STRUCTURES

**SHOSHANA ANILY**
*Tel Aviv University, Tel Aviv, Israel*

**JULIEN BRAMEL and DAVID SIMCHI-LEVI**
*Columbia University, New York, New York*

(Received January 1991; revisions received January, July 1992; accepted September 1992)

We consider the famous bin packing problem where a set of items must be stored in bins of equal capacity. In the classical version, the objective is to minimize the number of bins used. Motivated by several optimization problems that occur in the context of the storage of items, we study a more general cost structure where the cost of a bin is a concave function of the number of items in the bin. The objective is to store the items in such a way that total cost is minimized. Such cost functions can greatly alter the way the items should be assigned to the bins. We show that some of the best heuristics developed for the classical bin packing problem can perform poorly under the general cost structure. On the other hand, the so-called next-fit increasing heuristic has an absolute worst-case bound of no more than 1.75 and an asymptotic worst-case bound of 1.691 for any concave and monotone cost function. Our analysis also provides a new worst-case bound for the well studied next-fit decreasing heuristic when the objective is to minimize the number of bins used.

The bin packing problem can be stated as follows: Given a list of $n$ items each with size (0, 1], and an infinite sequence of empty bins, the objective is to assign each item to a bin such that the sum of the item sizes in a bin does not exceed 1, while minimizing the number of bins used.

The bin packing problem is one of the most extensively studied combinatorial problems. It belongs to the class of NP-hard problems and, therefore, the existence of a polynomial-time algorithm to solve the problem optimally is unlikely. As a result, since the early 1970s much research has been conducted to solve the problem to near optimality. The goal of this analysis is to prove that while a given heuristic may fail to find the optimal solution for every instance, it generates a solution which is always guaranteed to be within a certain percentage of the optimal solution. In this sense, it is a performance guarantee on the quality of the solution provided by the heuristic.

Johnson et al. (1974) was one of the first papers to present heuristics with performance guarantees on the bin packing problem. Subsequently, much research has been conducted to try to find heuristics with the best guarantee. An excellent survey of the research on this problem is available in Coffman, Garey and Johnson (1984).

In this paper, we analyze the bin packing problem with a more general cost structure. Unlike the classical bin packing problem, where the cost associated with a given bin is either zero or one, depending on whether it is empty or not, in our model the cost of a bin is a function of the number of items in the bin. This general cost structure may invoke the need for a careful coordination of the distribution of items in the bins. Specifically, we assume that the cost of a bin is a monotone and concave function of the number of items in a bin. The monotonicity and concavity properties are defined as:

*Monotonicity.* The cost of a bin does not decrease by the inclusion of additional items.
*Concavity.* The incremental cost due to the addition of an item to a collection of items is no more than the incremental cost resulting from the addition of an item to a set of items of smaller size.

We briefly describe an example to convince the reader that different cost functions can result in different optimal solutions. Consider a list consisting of four items of size $3/4$ and 16 items of size $1/16$. The optimal solution with respect to the classical bin packing problem is to have four identical bins with one item of size $3/4$ and four items of size $1/16$, with a

*Subject classifications:* Analysis of algorithms: heuristics and worst-case analysis. Mathematics: bin packing and combinatorial optimization.
*Area of review:* OPTIMIZATION.


0030-364X/94/4202-0287$01.25

total cost equal to 4. On the other hand, if the cost of a bin is given by $\sqrt{x}$, where $x$ is the number of items in the bin, the optimal solution consists of four bins with one item of size 3/4 and one bin with all the items of size 1/16, with a total cost equal to $4 + \sqrt{16} = 8$. Note that for cost function $\sqrt{x}$, the optimal solution to the classical bin packing problem has a total cost of $4\sqrt{5} > 8$.

In this paper, we seek heuristics for the general problem that have a fixed worst-case bound. That is, the heuristic is guaranteed to provide a solution whose cost is within a fixed ratio of the optimal cost.

In the next section, we provide a formal definition of our model as well as a number of examples from diverse areas, such as reliability, quality control and cryptography, that motivate our discussion. In Section 2, we present the notation and definitions used throughout the paper.

Since the heuristics we analyze were originally designed for the classical bin packing problem, in Section 3 we provide a brief overview of some of these heuristics. These include the **first-fit**, **best-fit**, **first-fit decreasing**, and **best-fit decreasing** algorithms analyzed in Johnson et al., as well as **next-fit decreasing** analyzed by Baker and Coffman (1981), and **next-fit increasing**.

In Section 4 we derive a characterization theorem that reduces the number of different cost functions that need to be studied for the purpose of worst-case analysis. That is, we demonstrate that to study the worst-case behavior of a particular algorithm, we only need to concentrate on a restricted class of cost functions. In addition, we present a polynomial-time procedure that yields a lower bound on the cost of the optimal solution for any concave cost function. These results are used in subsequent sections to prove some performance results for the bin packing problem with general costs as well as the classical bin packing problem.

For instance, in Section 5 we demonstrate that the lower bound described in Section 4 is also useful in proving some new worst-case results for the classical bin packing problem. To our knowledge, this is the first time a heuristic has been shown to have the so-called absolute performance ratio better than 2 for the classical bin packing problem.

In Section 6, we analyze the performance of the **next-fit increasing** and the **next-fit decreasing** heuristics and find that both have fixed worst-case bounds. We also point out that some of the best heuristics developed for the classical bin packing problem may be arbitrarily bad for the model with general cost structures.

Finally, in Section 7, we look at possible extensions of our results and give some concluding remarks.

## 1. THE MODEL AND MOTIVATION

Let $L = (w_1, w_2, \ldots, w_n)$ be a list of $n$ real numbers, where we call $w_i \in (0, 1]$ the size of item $i$. For simplicity, we also use $L$ as a set, but this should cause no confusion. In this case, we say item $i$ is in list $L$ ($i \in L$) to mean $w_i \in L$. The items are assigned to bins of unit capacity so that the total cost of all bins is as small as possible. The cost of a bin is a function of the number of items assigned to it and is represented by a concave and monotone function $f: \mathcal{N} \to \mathcal{R}$, which specifies a cost $f(j)$ for a bin containing $j$ items. The concavity and monotonicity properties can be expressed as:

*Monotonicity:* if $j \leq k$, $f(j) \leq f(k)$,

*Concavity:* for all $j \geq 1$, $f(j+1) - f(j) \leq f(j) - f(j-1)$.

We also assume, without loss of generality, that no cost is incurred if a bin is empty, i.e., $f(0) = 0$. In addition, we normalize costs so that one unit is incurred for a bin with only one item in it, i.e., $f(1) = 1$. We denote by $\mathcal{F}$ all possible functions $f$ that satisfy the above four conditions.

To motivate the model consider the following partitioning problem: $n$ *components* of size $w_i$ and probability of working $p_i$, $i = 1, 2, \ldots, n$, need to be partitioned into a number, say $m$, of disjoint sets $X_j$, $j = 1, 2, \ldots, m$, with $\sum_{i \in X_j} w_i \leq 1$ for each $j = 1, 2, \ldots, m$ in such a way that $\sum_{j=1}^{m} (1 - \Pi_{i \in X_j^{p_i}})$ is minimized.

Three optimization problems fall into this framework. One is in the area of systems reliability. Components are arranged into units where each unit is a *serial system*. The problem is to assign components to units of a total size no more than the capacity to minimize the expected number of nonworking units. One observes that, for the special case $p_i = p$ for all $i$, minimizing this objective is exactly the problem analyzed in this paper. This is true because $X_j$ is the set of components assigned to unit $j$ and therefore, in this special case, the probability that a unit with $k$ components does not work is $f(k) = 1 - p^k$, a concave and monotone function of the number of items put in the bin.

The previous example has an interesting interpretation in the area of quality control. Assume that items of size $w_i$ are packed into bins (or batches) of unit size and shipped to a client. The client opens each box and decides whether or not to accept the batch if every unit in the batch passes a certain test. The objective of the producer is to assign the items to the bins to minimize the expected number of rejected (or returned) bins. When the probability of passing the test is identical for all items, i.e., it is equal to $p$, this model is identical to our model. In this case, the probability that a bin with $k$ items is rejected is $1 - p^k$ and, therefore, minimizing the expected number of rejected bins is again the problem analyzed in this paper.

The other application is in the area of cryptography. Consider each component to be a coded message of length $w_i$ and with a probability $p_i$ of being decoded. A set of coding keys can each code a total message length of one unit. If message $i$ is decoded, then the coding key is revealed and the entire unit is deciphered. The objective is to find an assignment of messages to keys so that the expected number of revealed keys is minimized. Again, the problem analyzed in this paper is the special case $p_i = p$ for all $i$. Here $X_j$ is the set of messages that are coded with key $j$ and, consequently, when $p_i = p$ for all $i$, the probability of deciphering a key that coded $k$ messages is $f(k) = 1 - p^k$.

Also consider the familiar capacitated vehicle routing where customers have to be served by a fleet of identical vehicles initially located at a given depot. Each customer has a given demand, i.e., the amount of load that must be delivered to that customer. The objective is to find a set of routes that satisfies some constraints to minimize a given objective function. One important constraint is a capacity constraint on the amount of load delivered by a vehicle. Without loss of generality, we may assume that the vehicles' capacity equals 1 and the demand of a customer is no more than 1. Assume now that the cost of a route depends on the length $v$ of the tour and the number of customers $u$ visited in that tour, according to some function $f(v, u)$. It is easy to verify that when $f(\cdot, \cdot)$ is concave, in both its arguments, the bin packing problem and the vehicle routing problem are special cases of this problem. The model in this paper analyzes the case where $f(v, u) = f(u)$, while Bramel (1992) analyzes the case where $f(v, u) = f(v)$. For the latter model, Bramel shows that this model can be solved asymptotically by solving the bin packing problem in specially designed subregions, in the same way as it is done in Simchi-Levi and Bramel (1990) and Bramel et al. (1992). Furthermore, an important inventory routing problem is also a special case, as shown in Anily and Federgruen (1990).

Our model is also related to the class of so-called partitioning problems. In this class of problems, we are given a set of elements $X = \{x_1, x_2, \ldots, x_n\}$ and an attribute $w_i$ for every $x_i \in X$. The cost of a subset $X_l \subseteq X$ depends on its cardinality as well as the subset sum $(\sum_{x_i \in X_l} w_i)$, that is, $f(X_l) = f(|X_l|, \sum_{x_i \in X_l} w_i)$. The problem is to partition $X$ into $m$ disjoint subsets $X_1, X_2, \ldots, X_m$ to minimize $\sum_{l=1}^{m} f(X_l)$, where $m$ may or may not be prespecified. The problem has been analyzed, under some simplified assumptions on the cost structure, in Hwang (1981), Barnes and Hoffman (1984), Chakravarty, Orlin and Rothblum (1985), and Anily and Federgruen (1991). Our model can be viewed as a special case of this class when $f(X_l)$ depends on the cardinality of $X_l$, if $X_l$ is feasible, and has infinite cost otherwise.

In the next section, we introduce the notation and definitions that are used throughout this paper.

## 2. NOTATION AND DEFINITIONS

Let $Z_f^{\mathbf{H}}(L)$ be the cost of the solution produced by a heuristic $\mathbf{H}$ on list $L$, using cost function $f$. If $\mathbf{H}$ produces $m$ nonempty bins and $X_j$ is the set of items in the $j$th bin for $j = 1, 2, \ldots, m$, then $Z_f^{\mathbf{H}}(L) = \sum_{j=1}^{m} f(|X_j|)$. Similarly, let $Z_f^*(L)$ be the cost of the optimal packing of list $L$, with respect to cost function $f$. If the optimal solution consists of $m'$ nonempty bins and $S_j$ is the set of items in the $j$th bin for $j = 1, 2, \ldots, m'$, then $Z_f^*(L) = \sum_{j=1}^{m'} f(|S_j|)$.

We now formalize the worst-case analysis we consider in this paper. Let $\mathscr{L}$ denote all nonempty lists of finite length. Given a heuristic $\mathbf{H}$ for any $L \in \mathscr{L}$ and $f \in \mathscr{F}$, let $R_f^{\mathbf{H}}(L) = [Z_f^{\mathbf{H}}(L)]/[Z_f^*(L)]$. The *absolute performance ratio* for $\mathbf{H}$ with respect to cost function $f \in \mathscr{F}$ is given by

$$R_f^{\mathbf{H}} \equiv \inf\{r \geq 1 \mid R_f^{\mathbf{H}}(L) \leq r, \text{ for all } L \in \mathscr{L}\}.$$

The *asymptotic performance ratio* for $\mathbf{H}$ with respect to cost function $f \in \mathscr{F}$ is given by

$$R_f^{\mathbf{H}}(\infty) \equiv \inf\{r \geq 1 \mid \exists N > 0,\ R_f^{\mathbf{H}}(L) \leq r, \\ \text{for all } L \in \mathscr{L} \text{ with } Z_f^*(L) \geq N\}.$$

The absolute performance ratio for a heuristic $\mathbf{H}$ with respect to cost function $f$ gives, for all possible lists, the heuristic solution's maximum deviation from optimality. The asymptotic performance ratio for $\mathbf{H}$ provides the heuristic solution's maximum deviation from optimality for all lists that are sufficiently "large." We say that $\mathbf{H}$ has no finite absolute performance ratio for the bin packing problem with general cost structures, if $\sup_{f \in \mathscr{F}} R_f^{\mathbf{H}}$ is unbounded. Similarly, we say that $\mathbf{H}$ has no finite asymptotic performance ratio for the bin packing problem with general cost structures if $\sup_{f \in \mathscr{F}} R_f^{\mathbf{H}}(\infty)$ is unbounded. Note that, by definition, $R_f^{\mathbf{H}}(\infty) \leq R_f^{\mathbf{H}}$ for all $\mathbf{H}$ and $f \in \mathscr{F}$.

In this paper, we seek heuristics that work well for all possible cost functions $f \in \mathscr{F}$. We concentrate on the class of heuristics that performs in an identical manner with different cost functions. In other words, with cost functions $f$ and $g$, the heuristic produces exactly the same bins, though the cost of the solution may be different.

The following family of cost functions, which we call *flat cost functions*, plays a key role in our analysis. For any integer $k \geq 1$, the cost function $f_k$ is defined for each integer $j \geq 0$ as:

$$f_k(j) = \begin{cases} j, & \text{if } j < k; \\ k, & \text{if } k \leq j. \end{cases}$$

Note that $f_k \in \mathscr{F}$ for all integers $k \geq 1$. Also, note that $f_1$ is exactly the cost function in the classical bin packing problem, where a unit cost is incurred if a bin has at least one item and no cost is incurred if a bin is empty. To simplify the notation, we let $b^*(L) = Z_{f_1}^*(L)$ and $b^{\mathbf{H}}(L) = Z_{f_1}^{\mathbf{H}}(L)$, i.e., $b^*(L)$ is the number of nonempty bins in an optimal solution to the classical bin packing problem, while $b^{\mathbf{H}}(L)$ is the number of nonempty bins produced by $\mathbf{H}$.

We now define the following terms which we use throughout the paper: Two lists $L_1$ and $L_2$ are *consecutive* if $L_1 \cap L_2 = \emptyset$, i.e., no item belongs to both of them, and for any $i \in L_1$ and $j \in L_2$ we have $w_i \leq w_j$. Similarly, define two bins to be consecutive if the lists consisting of the items from the first bin and the second bin are consecutive lists. Both of these definitions can be generalized in the obvious way to $t$ consecutive lists (or bins) for any integer $t \geq 2$.

Finally, we recall some definitions used throughout the bin packing literature. Call a bin *feasible* if the sum of the item sizes in the bin does not exceed 1. An item is said to *fit* in a bin if the bin resulting from the insertion of this item is a feasible bin. In addition, a bin is *opened* when an item is placed in a bin that was previously empty.

## 3. BRIEF OVERVIEW OF BIN PACKING HEURISTICS

We present a brief description of some of the simpler heuristics for the bin packing problem that have appeared in the literature. The first and most fundamental heuristic is called **next-fit (NF)** and can be described in the following manner. Starting with item 1, place this item in bin 1. Suppose that we are packing item $j$. Let bin $i$ be the highest indexed nonempty bin. If item $j$ fits in bin $i$, then place it there, else place it in a new bin indexed $i + 1$. Slightly more complicated is the **first-fit (FF)** heuristic, which can be described in the following manner. Place item 1 in bin 1. Suppose that we are packing item $j$. Place item $j$ in the lowest indexed bin whose current content does not exceed $1 - w_j$. The **best-fit (BF)** heuristic has also been studied extensively and can be described succinctly in the following manner. Place item 1 in bin 1. Suppose that we are packing item $j$. Place item $j$ in the bin whose current content is the largest but does not exceed $1 - w_j$.

The heuristics described above assign items to bins according to the order they appear in the list without using any knowledge of subsequent items in the list (see, for details, Coffman, Garey and Johnson). These types of heuristics are called *on-line* heuristics. In contrast to this class, *off-line* heuristics accept as input the exact size of all the items in the list and, therefore, assigning the items to bins according to some a priori sequence is possible. For example, the **first-fit decreasing (FFD)** heuristic first sorts the items in nonincreasing order of their sizes and then performs first-fit. A similar interpretation holds for the **best-fit decreasing (BFD)**, **next-fit increasing (NFI)** and **next-fit decreasing (NFD)** heuristics. Table I presents performance ratios for these heuristics on the classical bin packing problem.

## 4. PRELIMINARIES

In this section we touch upon two fundamental ideas which are referred to throughout this paper. The first is Theorem 1 that reduces the number of different cost functions that need to be studied for the purpose of worst-case analysis. The second is a procedure that yields a lower bound on the cost of the optimal solution to the bin packing problem with general costs.

To prove Theorem 1 we need the following lemma.

**Lemma 1.** *For any integer $m \geq 1$, nonnegative real numbers $x_j$, $s_j$, and $\Delta_j$ for $j = 1, 2, \ldots, m$, such that $\Delta_j \geq \Delta_{j+1}$ for $j = 1, 2, \ldots, m - 1$, $s_1 > 0$ and $\Delta_1 > 0$, we have*

$$\frac{\sum_{j=1}^m x_j \Delta_j}{\sum_{j=1}^m s_j \Delta_j} \leq \max_{i=1,2,\ldots,m} \frac{\sum_{j=1}^i x_j}{\sum_{j=1}^i s_j}.$$

**Proof.** Let $R = \max_{i=1,2,\ldots,m} \frac{\sum_{j=1}^i x_j}{\sum_{j=1}^i s_j}$. By this definition, we have

$$\sum_{j=1}^{i} x_j \leq R \sum_{j=1}^{i} s_j \quad \text{for } i = 1, 2, \ldots, m.$$

**Table I**
Asymptotic Performance Ratios

| Heuristic | Classical Bin Packing $R_{f_1}^{H}(\infty)$ | Reference | General Cost Structure $\sup_{f\in F} R_f^{H}(\infty)$ | Reference |
|---|---|---|---|---|
| **BF** | 1.7 | Johnson et al. | $\infty$ | Theorem 4 |
| **BFD** | 1.222 . . . | Johnson et al. | $\infty$ | Theorem 4 |
| **FF** | 1.7 | Johnson et al. | $\infty$ | Theorem 4 |
| **FFD** | 1.222 . . . | Johnson et al. | $\infty$ | Theorem 4 |
| **NF** | 2 | Coffman et al. | $\infty$ | Theorem 4 |
| **NFD** | 1.691 . . . | Baker and Coffman | 2 | Theorem 5 |
| **NFI** | 1.691 . . . | Property 3 | 1.691 . . . | Theorem 6 |

Multiplying the $i$th inequality by the nonnegative value $(\Delta_i - \Delta_{i+1})$, where $\Delta_{m+1} \equiv 0$, and summing the resulting inequalities, we get

$$\sum_{i=1}^{m} (\Delta_i - \Delta_{i+1}) \sum_{j=1}^{i} x_j \leq R \sum_{i=1}^{m} (\Delta_i - \Delta_{i+1}) \sum_{j=1}^{i} s_j.$$

It is easy to see that

$$\sum_{i=1}^{m} (\Delta_i - \Delta_{i+1}) \sum_{j=1}^{i} x_j = \sum_{j=1}^{m} x_j \Delta_j$$

and

$$\sum_{i=1}^{m} (\Delta_i - \Delta_{i+1}) \sum_{j=1}^{i} s_j = \sum_{j=1}^{m} s_j \Delta_j.$$

Hence,

$$\sum_{j=1}^{m} x_j \Delta_j \leq R \sum_{j=1}^{m} s_j \Delta_j.$$

Consider any heuristic **H** that constructs a solution independent of the cost function. The following theorem implies, surprisingly, that for such a heuristic **H** and cost function $f$ there is a function in the family of flat cost functions (i.e., $f_k$ for some $k \geq 1$) which has an absolute performance ratio that dominates the absolute performance ratio of $f$. To this end, define $\Delta_j = f(j) - f(j-1)$, when the cost function $f$ is clear from the context. Note that because $f$ is monotone and concave, $\Delta_j \geq \Delta_{j+1} \geq 0$ for all integers $j \geq 1$.

**Theorem 1.** *For any cost function $f \in \mathcal{F}$, list $L \in \mathcal{L}$ and any heuristic* **H** *that constructs a solution independent of the cost function $f$, there exists an integer $k \geq 1$, such that*

$$\frac{Z_f^{H}(L)}{Z_f^{*}(L)} \leq \frac{Z_{f_k}^{H}(L)}{Z_{f_k}^{*}(L)}. \tag{1}$$

**Proof.** Given a list $L$ with $n$ items and a cost function $f$ for $j = 1, 2, \ldots, n$, let $x_j$ (respectively, $s_j$) denote the number of bins that contains at least $j$ items in the solution produced by **H** (respectively, the optimal solution with respect to cost function $f$). In view of the definition of $\Delta_j$ and the fact that no bin can contain more than $n$ items, we can write $Z_f^{H}(L)$ as $\sum_{j=1}^{n} x_j \Delta_j$ and $Z_f^{*}(L)$ as $\sum_{j=1}^{n} s_j \Delta_j$. Furthermore, $L \neq \emptyset$ implies that $s_1 > 0$, so let $k \geq 1$ be the smallest integer such that

$$\max_{i=1,2,\ldots,n} \frac{\sum_{j=1}^{i} x_j}{\sum_{j=1}^{i} s_j} = \frac{\sum_{j=1}^{k} x_j}{\sum_{j=1}^{k} s_j}.$$

Then, since all the conditions required in Lemma 1 are satisfied, we have

$$\frac{Z_f^{H}(L)}{Z_f^{*}(L)} = \frac{\sum_{j=1}^{n} x_j \Delta_j}{\sum_{j=1}^{n} s_j \Delta_j} \leq \frac{\sum_{j=1}^{k} x_j}{\sum_{j=1}^{k} s_j}.$$

For $j \geq 1$, let

$$\Delta_j' = f_k(j) - f_k(j-1) = \begin{cases} 1, & \text{if } j \leq k; \\ 0, & \text{if } k < j, \end{cases}$$

and because **H** produces exactly the same bins with costs $f$ and $f_k$, we have

$$Z_{f_k}^{H}(L) = \sum_{j=1}^{n} x_j \Delta_j' = \sum_{j=1}^{k} x_j.$$

It is also clear that

$$Z_{f_k}^{*}(L) \leq \sum_{j=1}^{n} s_j \Delta_j' = \sum_{j=1}^{k} s_j,$$

because $Z_{f_k}^{*}(L)$ is the optimal solution to $L$ with respect to cost function $f_k$, while $\sum_{j=1}^{k} s_j$ is the cost using function $f_k$ of a feasible solution, namely the optimal solution to $L$ with respect to $f$. Therefore,

$$\frac{Z_f^{H}(L)}{Z_f^{*}(L)} \leq \frac{\sum_{j=1}^{k} x_j}{\sum_{j=1}^{k} s_j} \leq \frac{Z_{f_k}^{H}(L)}{Z_{f_k}^{*}(L)}.$$

The above property is used in subsequent sections to obtain worst-case results for the **NFI** heuristic. The proof of these results, as well as worst-case results for other heuristics, naturally involve a lower bound on the optimal solution to the bin packing problem. For this purpose, we introduce the following procedure and devote the rest of this section to prove that it yields a lower bound on the minimal cost of the bin packing problem with general cost structures. The procedure, which we call procedure **A**, first orders the list $L$ in nondecreasing order of the item sizes, such that $w_1 \leq w_2 \leq \ldots \leq w_n$. Starting from item 1, assign the maximum possible items to bin 1 such that this bin is feasible. Close bin 1. Suppose that we are packing bin $i$, $i > 1$. If bin $i$ is feasible, then assign the next item to bin $i$, or else close bin $i$ and place the item in a new bin indexed $i + 1$.

It is clear that any nonempty bin generated by this procedure contains one more item than the bin can actually hold, except for the first and possibly the last bin. Hence, the solution produced by procedure **A** does not consist of feasible bins and, therefore, is not a feasible solution to the bin packing problem.

Let $S_j$ be the set of items in the $j$th bin produced by procedure **A** for $j = 1, 2, \ldots, m'$ where $m'$ is the number of nonempty bins. From the construction $\sum_{i \in S_j} w_i > 1$ for $1 < j < m'$ and possibly for $j = m'$. Given a feasible solution to the bin packing problem, let $X_j$ be the set of items in the $j$th bin for $j = 1, 2, \ldots, m$, where $m$ is the number of bins used in this solution. Index these bins so that $|X_1| \geq |X_2| \geq \ldots \geq |X_m|$.

The following property shows that the number of nonempty bins created by procedure **A** is no more than the number of nonempty bins used in any feasible solution.

**Property 1.** *Here* $m' \leq m$.

**Proof.** Note that bins $2, 3, \ldots, m' - 1$ of the solution produced by procedure **A** are not feasible bins, hence $\sum_{i \in L \setminus (S_1 \cup S_{m'})} w_i > m' - 2$. Also note that $\sum_{i \in S_1 \cup S_{m'}} w_i > 1$ by the selection of the items for bin 1 and the fact that items in bin $m'$ are the largest items in $L$. Adding these two inequalities, we have $\sum_{i=1}^{n} w_i > m' - 1$. Since any feasible solution to the bin packing problem always consists of a set of feasible bins, we also have $m \geq \sum_{i=1}^{n} w_i$. Therefore, $m' - 1 < m$, or $m' \leq m$.

For simplicity of notation, let $S_{m'+1} = S_{m'+2} = \ldots = S_m = \emptyset$. Also, for $j = 1, 2, \ldots, m$, define $\bar{S}_j = \cup_{i=1}^{j} S_i$ and $\bar{X}_j = \cup_{i=1}^{j} X_i$, and note that $\bar{S}_m = \bar{X}_m = L$.

The next property shows that if we look at the first $j$ bins of the solution produced by procedure **A** and the first $j$ bins of any feasible solution, **A** can never pack fewer items in the first $j$ bins than can any feasible solution. This is the basic property that provides the lower bound.

**Property 2.** *Here* $|\bar{S}_j| \geq |\bar{X}_j|$ *for* $j = 1, 2, \ldots, m$.

**Proof.** For $j = m$, we have equality. Hence, we only need to show that the claim is true for $j < m$. Assume that $j < m$ and, by contradiction, $|\bar{X}_j| > |\bar{S}_j|$. Then $\sum_{i \in \bar{X}_j} w_i > \sum_{i \in \bar{S}_j} w_i$, because $\bar{S}_j$ consists of the $|\bar{S}_j|$ smallest items in $L$. If we let item $l_j$ be the smallest item not in $\bar{S}_j$, then $\sum_{i \in \bar{X}_j} w_i \geq \sum_{i \in \bar{S}_j} w_i + w_{l_j}$. Note that the following relation also holds:

$$\sum_{i \in \bar{S}_j} w_i + w_{l_j} > j \quad \text{for all } j < m,$$

because bin 1 is the only bin that is not filled over capacity and adding item $l_j$ to it would clearly make bin 1 go over its capacity. Since $\sum_{i \in X_j} w_i \leq 1$ for $j = 1, 2, \ldots, m$, we have

$$j \geq \sum_{i \in \bar{X}_j} w_i \geq \sum_{i \in \bar{S}_j} w_i + w_{l_j} > j \quad \text{for all } j < m,$$

which is a contradiction.

The proof that procedure **A** yields a lower bound will be based on a counting argument. For this purpose we need to construct the $m$ lists of integers denoted by $E_1, E_2, \ldots, E_m$, using the following procedure, which we call procedure **B**. To begin, let $E$ be an empty list. Starting with bin $k = 1$ if $|S_k| > |X_k|$, then *append* the list of integers $(|X_k| + 1, |X_k| + 2, \ldots, |S_k|)$ to list $E$. If $|S_k| < |X_k|$, then *remove* the last $|X_k| - |S_k|$ elements of list $E$. Let $E_k$ be the current list $E$ (i.e., $E_k \leftarrow E$), and increment $k$ until $k > m$.

By the construction of these lists, the number of elements in list $E_k$, denoted by $|E_k|$, is exactly $\sum_{j=1}^{k} \{|S_j| - |X_j|\} = |\bar{S}_k| - |\bar{X}_k|$, which is always nonnegative by Property 2.

The next lemma establishes the main fact used to show that the cost of the solution produced by procedure **A** provides a lower bound on the cost of any feasible solution to the bin packing problem with general cost structures. For this lemma we recall the definition $\Delta_j = f(j) - f(j-1)$.

**Lemma 2.** *For any concave cost function* $f \in \mathcal{F}$ *and any integer* $p$, $1 \leq p \leq m$, *we have*

$$\sum_{j=1}^{p} f(|S_j|) \leq \sum_{j=1}^{p} f(|X_j|) + \sum_{j \in E_p} \Delta_j, \tag{2}$$

*where* $j \in E_p$ *means* $j$ *ranges over the elements of the list* $E_p$.

**Proof.** The proof is by induction. If $p = 1$, then from Property 2, we have $|S_1| \geq |X_1|$. Hence, inequality (2) is satisfied as an equality because

$$f(|S_1|) = f(|X_1|) + \sum_{j=|X_1|+1}^{|S_1|} \Delta_j = f(|X_1|) + \sum_{j \in E_1} \Delta_j.$$

Assume that the claim is true for $p = k - 1$, i.e.,

$$\sum_{j=1}^{k-1} f(|S_j|) \leq \sum_{j=1}^{k-1} f(|X_j|) + \sum_{j \in E_{k-1}} \Delta_j. \tag{3}$$

There are three cases to consider.

**Case 1.** ($|S_k| = |X_k|$) Then we have $f(|S_k|) = f(|X_k|)$ and $E_k = E_{k-1}$. Combining these facts with inequality (3), we see that (2) is satisfied for $p = k$.

**Case 2.** ($|S_k| > |X_k|$) Then

$$f(|S_k|) = f(|X_k|) + \sum_{j=|X_k|+1}^{|S_k|} \Delta_j$$

$$= f(|X_k|) + \sum_{j \in E_k} \Delta_j - \sum_{j \in E_{k-1}} \Delta_j,$$

and, therefore, adding this equation to inequality (3) shows that (2) also holds for $p = k$ in this case.

**Case 3.** ($|S_k| < |X_k|$) First note that if $j \in E_{k-1}$ this integer represents an item that came from some bin $i$ with $i \leq k - 1$, where $|S_i| \geq j > |X_i|$. This is the only case where elements are added to the list. So by the indexing of the bins in the feasible solution, $j > |X_i| \geq |X_k|$. Let $D$ represent the list of $|X_k| - |S_k|$ elements removed by procedure **B** from list $E_{k-1}$ to construct list $E_k$. If $j \in D$, then $j \in E_{k-1}$ and hence, $j > |X_k| \Rightarrow \Delta_j \leq \Delta_{|X_k|}$, since $f$ is concave.

Summing over the elements of the list $D$ we have,

$$\sum_{j \in D} \Delta_j \leq |D| \cdot \Delta_{|X_k|} \leq \sum_{j=|S_k|+1}^{|X_k|} \Delta_j, \tag{4}$$

because $|D| = |X_k| - |S_k|$ and $\Delta_j \geq \Delta_{j+1}$, for all $j \geq 1$. By the definition of $D$, we have

$$\sum_{j \in E_{k-1}} \Delta_j = \sum_{j \in E_k} \Delta_j + \sum_{j \in D} \Delta_j$$

$$\leq \sum_{j \in E_k} \Delta_j + \sum_{j=|S_k|+1}^{|X_k|} \Delta_j \quad \text{(by 4)}$$

$$= \sum_{j \in E_k} \Delta_j + f(|X_k|) - f(|S_k|).$$

Or, equivalently,

$$f(|S_k|) \leq f(|X_k|) + \sum_{j \in E_k} \Delta_j - \sum_{j \in E_{k-1}} \Delta_j.$$

Adding inequality (3) to this inequality, we see that (2) also holds for $p = k$ in this case. Hence, (2) holds for all $p \leq m$ and the proof is complete.

We may conclude the following.

**Theorem 2.** *The cost of the solution produced by procedure* **A** *is a lower bound on the cost of the optimal solution to the bin packing problem with general cost structures.*

**Proof.** Since $|E_k| = |\bar{S}_k| - |\bar{X}_k|$ for $k = 1, 2, \ldots, m$, then $E_m$ is an empty list, so from Lemma 2 we have $\sum_{j=1}^{m} f(|S_j|) \leq \sum_{j=1}^{m} f(|X_j|)$ for any feasible solution, where bin $j$ contains items $X_j$ for $j = 1, 2, \ldots, m$.

Our analysis indicates that a stronger result holds. Indeed, a variety of different procedures yields lower bounds on the optimal solution value to the bin packing problem with general cost structures, as demonstrated by the following corollary.

**Corollary 1.** *Any packing rule that results in consecutive bins where each bin is filled over the capacity, except possibly the last bin (the bin with the largest items), provides a solution whose cost is a lower bound to the cost of any feasible solution to the bin packing problem for all concave cost functions.*

## 5. NEW RESULTS FOR THE CLASSICAL BIN PACKING PROBLEM

In this section, we show that the lower bound provided by procedure **A** reveals new worst-case results for the classical bin packing problem. Specifically, we prove that the absolute performance ratio of the well-studied **NFD** heuristic (see Baker and Coffman) is no more than 1.75. Our proof requires the following property.

**Property 3.** *For any list* $L \in \mathscr{L}$, $b^{\mathbf{NFI}}(L) = b^{\mathbf{NFD}}(L)$.

**Proof.** For any two consecutive lists $L_1$ and $L_2$, let $L \equiv L_1 \cup L_2$. We clearly have

$$b^{\mathbf{NFI}}(L) \leq b^{\mathbf{NFI}}(L_1) + b^{\mathbf{NFI}}(L_2) \tag{5}$$

and

$$b^{\mathbf{NFD}}(L) \leq b^{\mathbf{NFD}}(L_1) + b^{\mathbf{NFD}}(L_2).$$

These inequalities are true because, given a set of consecutive feasible bins, **NFI** and **NFD** can only improve on that solution. Then each heuristic can only improve on each other's solution. Hence, **NFD** and **NFI**, surprisingly, must produce exactly the same number of bins for any list $L$.

**Theorem 3.** *For any list $L \in \mathscr{L}$,*

$$\frac{b^{\mathrm{NFD}}(L)}{b^*(L)} = \frac{b^{\mathrm{NFI}}(L)}{b^*(L)} \leqslant 1.75.$$

**Proof.** We prove the result for **NFI** and by Property 3 the result holds for **NFD** as well. Given any list $L$, let $T$ be the sublist of $L$ consisting of all items whose size is strictly greater than ½. Let $S_j$ be the items in the $j$th bin produced by procedure **A** for $j = 1, 2, \ldots, m$, where bin 1 contains the smallest items and bin $m$, the largest items. Clearly, any two items in $L \setminus T$ can share a bin, and no feasible bin can contain two elements from $T$. Omitting trivial cases, we assume that $m \geqslant 2$ and $L \setminus T \neq \emptyset$. In view of these assumptions, a lower bound on any solution will be $\max\{m, |T|\}$. Define bin $p$ to be the highest indexed bin produced by procedure **A** that has an element of $L \setminus T$.

**Case 1.** ($p = 1$) The solution produced by **A** consists of bin 1 which is feasible, while bins $2, 3, \ldots, m$ have at most two items, each of which is an element of $T$. Clearly, we have $|T| \geqslant 2(m-2) + |S_m|$. Since bin 1 is feasible, **NFI** produces exactly $1 + 2(m-2) + |S_m|$ bins, and therefore

$$\begin{aligned} b^{\mathrm{NFI}}(L) &= 1 + 2(m-2) + |S_m| \\ &\leqslant |T| + 1 \\ &\leqslant |T| + \frac{m}{2} \quad (\text{since } m \geqslant 2) \\ &\leqslant \frac{3}{2}\, b^*(L). \end{aligned}$$

**Case 2.** ($m = p > 1$) In the solution produced by procedure **A**, at most $m - 1$ bins are filled over their capacity. Therefore, if we take the largest item from bin 2 and move it to bin 3, bin 2 will be feasible. If we take the two largest items now in bin 3 and move them to bin 4, bin 3 will be feasible. Continuing this process until bin $m - 1$ we end up with $m - 1$ feasible bins and $m - 2$ items left over, as well as bin $m$. Clearly, these $m - 2$ extra items can be put two in a bin in a consecutive manner since these items are not in $T$. Bin $m$ can be split into two consecutive bins by taking the largest item out if necessary. Hence, the bins created are consecutive, and from (5) we have

$$\begin{aligned} b^{\mathrm{NFI}}(L) &\leqslant m - 1 + \lceil (m-2)/2 \rceil + 2 \\ &\leqslant \frac{3}{2} m + \frac{1}{2} \\ &\leqslant \frac{7}{4} m \quad (\text{since } m \geqslant 2) \\ &\leqslant \frac{7}{4}\, b^*(L). \end{aligned}$$

**Case 3.** ($m > p > 1$ and $T \cap S_p = \varnothing$). The lower bound on the optimal solution is $\max\{m, 2(m-p-1) + |S_m|\}$, because there are $2(m-p-1) + |S_m|$ items in $T$. Hence, using similar reasoning as in case 2,

$$\begin{aligned} &b^{\mathrm{NFI}}(L) \\ &\leqslant p + \lceil (p-1)/2 \rceil + 2(m-p-1) + |S_m| \\ &\leqslant p + \frac{p}{2} + 2(m-p) - 2 + |S_m| \\ &= \frac{3}{2} m + \frac{1}{4}[2(m-p-1) + |S_m|] - \frac{3}{2} + \frac{3}{4}|S_m| \\ &\leqslant \frac{3}{2} m \\ &\quad + \frac{1}{4}[2(m-p-1) + |S_m|] \quad (\text{since } |S_m| \leqslant 2) \\ &\leqslant \frac{7}{4}\, b^*(L). \end{aligned}$$

**Case 4.** ($m > p > 1$ and $T \cap S_p \neq \emptyset$) Let $w_A$ be the size of the largest item in bin $p$, and $w_B$ the size of the second largest item in that bin. Note that $w_A > ½$ by the assumption that $T \cap S_p \neq \emptyset$ and the definition of $p$.

**Subcase 4.1.** ($w_A + w_B \leqslant 1$) The lower bound on the optimal solution is $\max\{m, 2(m-p-1) + |S_m| + 1\}$, because there are $2(m-p-1) + |S_m| + 1$ items in the list $T$. First note that bins $1, 2, \ldots, p-2$ have no items of size greater than ½. So if we let $L_1$ be the items in the first $p - 2$ bins, from case 2 we have $b^{\mathrm{NFI}}(L_1) \leqslant p - 2 + \lceil (p-3)/2 \rceil$. Let $L_2$ be the items in bins $p - 1$ and $p$. We claim that $b^{\mathrm{NFI}}(L_2) \leqslant 3$. Take the largest item from bin $p - 1$ and place it in bin $p$, so bin $p - 1$ is now a feasible bin. Taking the two largest items out of bin $p$ (the items of size $w_A$ and $w_B$) will now leave a feasible bin. Since $w_A + w_B \leqslant 1$, these two items fit in a bin together, and therefore we have constructed three consecutive bins consisting of the items in $L_2$. Hence,

$$\begin{aligned} &b^{\mathrm{NFI}}(L) \\ &\leqslant b^{\mathrm{NFI}}(L_1) + b^{\mathrm{NFI}}(L_2) + 2(m-p-1) + |S_m| \\ &\leqslant p - 2 + \lceil (p-3)/2 \rceil + 3 \\ &\quad + 2(m-p-1) + |S_m| \\ &\leqslant p + \frac{p}{2} + 2(m-p-1) + |S_m| \\ &= \frac{3}{2} m + \frac{1}{4}(2(m-p-1) + |S_m|) + \frac{3}{4}|S_m| - \frac{3}{2} \\ &\leqslant \frac{7}{4}\, b^*(L) \quad (\text{since } |S_m| \leqslant 2). \end{aligned}$$

**Subcase 4.2.** ($w_A + w_B > 1$). The lower bound on the optimal solution is now $\max\{m, 2(m-p) + |S_m|\}$, because there are $2(m-p) + |S_m|$ items that must be in separate bins. Note that bins $1, 2, \ldots, p-1$ have no items of size greater than 1/2. So if we let $L_1$ be those items, we have $b^{\mathrm{NFI}}(L_1) \leqslant p - 1 + \lceil (p-2)/2 \rceil$. Let $L_2$ be the items in bin $p$. Clearly $b^{\mathrm{NFI}}(L_2) = 2$. Hence,

$$
\begin{aligned}
&b^{\mathrm{NFI}}(L)\\
&\leqslant b^{\mathrm{NFI}}(L_1) + b^{\mathrm{NFI}}(L_2) + 2(m-p-1) + |S_m|\\
&\leqslant p - 1 + \lceil (p-2)/2 \rceil + 2\\
&\quad + 2(m-p-1) + |S_m|\\
&\leqslant p + \frac{p}{2} + \frac{1}{2} + 2(m-p-1) + |S_m|\\
&= \frac{3}{2} m + \frac{1}{4}(2(m-p) + |S_m|) + \frac{3}{4}|S_m| - \frac{3}{2}\\
&\leqslant \frac{7}{4} b^*(L) \quad (\text{since } |S_m| \leqslant 2).
\end{aligned}
$$

Therefore, in all possible cases the solution provided by **NFI** is within 1.75 of the optimal solution, and hence, by Property 3 both **NFI** and **NFD** have absolute performance ratios of no more than 1.75 for the classical bin packing problem.

Johnson et al. provide an example where the ratio of the number of bins produced by **NFI** and the number of bins used in the optimal solution is exactly 1.7. This example gives the largest deviation from optimality known.

## 6. ANALYSIS OF HEURISTICS FOR THE GENERAL COST MODEL

In this section we analyze heuristics for the bin packing problem with general cost structures, and give some worst-case results. We start with a theorem whose proof is given in Bramel.

**Theorem 4.** **NF**, **FF**, **BF**, **FFD** *and* **BFD** *have neither finite absolute performance ratios nor finite asymptotic performance ratios for the bin packing problem with general cost structures.*

It is interesting to observe (see Table I) that while **FFD** and **BFD** are among the best known heuristics for the classical bin packing problem, in terms of deviation from optimality, they can perform very poorly with the general cost structure. We, therefore, turn our attention to heuristics that have finite performance ratios for the bin packing problem with general cost structures.

Our first result is that the best absolute and asymptotic performance ratios possible for **NFD** over all cost functions $f$ in $\mathscr{F}$ is 2. For a formal proof, the reader is referred to Bramel.

**Theorem 5.** *For any $L \in \mathscr{L}$ and for all $f \in \mathscr{F}$, the following relation holds $[Z_f^{\mathrm{NFD}}(L)/Z_f^*(L)] \leqslant 2$, and, moreover, for any given $\epsilon > 0$ there exists an $f \in \mathscr{F}$, such that $R_f^{\mathrm{NFD}}(\infty) \geqslant 2^{1-\epsilon} - \epsilon$.*

### 6.1. Next-Fit Increasing

In this subsection, we consider the **NFI** heuristic and prove the following result.

**Theorem 6.** *For any list $L$ and for all $f \in \mathscr{F}$, we have*

$$
\begin{aligned}
&R_f^{\mathrm{NFI}}(L)\\
&= \frac{Z_f^{\mathrm{NFI}}(L)}{Z_f^*(L)}\\
&\leqslant \min\left\{1.75,\ 1.7 + \frac{2}{b^*(L)},\ 1.691\ldots + \frac{3}{b^*(L)}\right\},
\end{aligned}
$$

*and for all $f \in \mathscr{F}$, $R_f^{\mathrm{NFI}}(\infty) \leqslant R_{f_1}^{\mathrm{NFI}}(\infty) = 1.691\ldots$.*

$R_f^{\mathrm{NFI}}(\infty) \leqslant R_{f_1}^{\mathrm{NFI}}(\infty) = 1.691\ldots$.

The proof of Theorem 6 requires several steps. We start with the following lemma.

**Lemma 3.** *For any cost function $f \in \mathscr{F}$ and any two consecutive lists $L_1$ and $L_2$, let $L \equiv L_1 \cup L_2$. Then*

$$Z_f^{\mathrm{NFI}}(L) \leqslant Z_f^{\mathrm{NFI}}(L_1) + Z_f^{\mathrm{NFI}}(L_2).$$

**Proof.** Let $X_1, X_2, \ldots, X_{m_1}$ be the nonempty bins produced by **NFI** on list $L_1$, indexed in the order the bins are opened. Let $Y_1, Y_2, \ldots, Y_{m_2}$ be the nonempty bins produced by **NFI** on list $L_2$, also indexed in the order the bins are opened. We prove the lemma by induction on the value of $|L_2|$.

If $|L_2| = 1$, then $Z_f^{\mathrm{NFI}}(L_2) = 1$. Given the **NFI** solution to $L_1$, if the item in $L_2$ fits in bin $m_1$, then it would be placed there in the **NFI** solution to list $L$. Therefore in this case,

$$Z_f^{\mathrm{NFI}}(L) = Z_f^{\mathrm{NFI}}(L_1) + \Delta_{|X_{m_1}|+1} \leqslant Z_f^{\mathrm{NFI}}(L_1) + Z_f^{\mathrm{NFI}}(L_2).$$

If the item in $L_2$ does not fit in bin $m_1$, then the item would be in a bin by itself with added cost of exactly $\Delta_1 = Z_f^{\mathrm{NFI}}(L_2)$. Therefore, the claim is true for all consecutive lists $L_1$ and $L_2$ with $|L_2| = 1$.

For $k \geqslant 2$ assume that for all consecutive lists $L_1$ and $L_2$ with $|L_2| \leqslant k-1$ we have $Z_f^{\mathrm{NFI}}(L) \leqslant Z_f^{\mathrm{NFI}}(L_1) + Z_f^{\mathrm{NFI}}(L_2)$, where $L \equiv L_1 \cup L_2$. Given two consecutive lists $L_1$ and $L_2$, with $|L_2| = k$, perform heuristic **NFI** separately on each of these lists. Using the notation defined above, let $L_1' = L_1 \cup Y_1$ and $L_2' = L_2 \backslash Y_1$. Therefore, list $L_1'$ consists of list $L_1$ and all the items from the first bin of the **NFI** solution to list $L_2$, while list $L_2'$ consists of all the remaining items. Moreover, because $|Y_1| \geq 1$, we have $|L_2'| \leq k - 1$. From the induction assumption, because $L_1'$ and $L_2'$ are consecutive lists and $L_1' \cup L_2' = L$, we have

$$Z_f^{\mathrm{NFI}}(L) \leq Z_f^{\mathrm{NFI}}(L_1') + Z_f^{\mathrm{NFI}}(L_2').$$

Now, the first $m_1 - 1$ bins of the **NFI** solutions to $L_1$ and $L_1'$ are identical. However, items in the set $Y_1$ may now be placed in bin $m_1$ in the **NFI** solution to $L_1'$. Let $B$ be this set of items from $Y_1$ that are placed in bin $m_1$. Clearly $B \subseteq Y_1$ consists of the $|B|$ smallest items of $Y_1$. In the **NFI** solution to $L_1'$, bin $m_1$ consists of $|X_{m_1}| + |B|$ items and an extra bin $m_1 + 1$ consists of $|Y_1| - |B|$ items, hence we have

$$Z_f^{\mathrm{NFI}}(L_1') = Z_f^{\mathrm{NFI}}(L_1) - f(|X_{m_1}|) + f(|X_{m_1}| + |B|) + f(|Y_1| - |B|).$$

On the other hand, the **NFI** solution to $L_2'$ consists of the same bins as the **NFI** solution to list $L_2$ with the first bin removed, and therefore $Z_f^{\mathrm{NFI}}(L_2') = Z_f^{\mathrm{NFI}}(L_2) - f(|Y_1|)$. Using the induction assumption, we have

$$Z_f^{\mathrm{NFI}}(L) \leq Z_f^{\mathrm{NFI}}(L_1) - f(|X_{m_1}|) + f(|X_{m_1}| + |B|) + f(|Y_1| - |B|) + Z_f^{\mathrm{NFI}}(L_2) - f(|Y_1|).$$

To prove the lemma, we need only show that

$$f(|X_{m_1}| + |B|) - f(|X_{m_1}|) + f(|Y_1| - |B|) - f(|Y_1|) \leq 0,$$

or, equivalently, that

$$f(|X_{m_1}| + |B|) - f(|X_{m_1}|) \leq f(|Y_1|) - f(|Y_1| - |B|).$$

Because $f$ is concave and monotone, it should be clear that this is equivalent to showing that $|X_{m_1}| \geq |Y_1| - |B|$. But this is clearly true because the number of items in bins produced by **NFI** is nonincreasing with the bin index. So $|X_{m_1}| + |B| \geq |Y_1|$ and hence the lemma is true.

It is interesting to observe that if the lists $L_1$ and $L_2$ are not consecutive, the lemma does not necessarily hold, as demonstrated by the following example. Consider lists $L_1 = L_2 = (1/3, 2/3)$ with cost function $f_1$. The solution produced by **NFI** on the union of the lists produces three bins, while the **NFI** solution to each list produces one bin.

In addition, note that Lemma 3 can be generalized to $t$ consecutive lists for any integer $t \geq 2$, and the claim in the lemma is changed in the obvious way. As a result we have the following corollary.

**Corollary 2.** *If a heuristic* **H** *always produces consecutive bins, then for all* $L \in \mathscr{L}$ *and* $f \in \mathscr{F}$, $Z_f^{\mathrm{NFI}}(L) \leq Z_f^{\mathrm{H}}(L)$, *that is,* **NFI** *is the best heuristic among all those that produce consecutive bins.*

We can now present Lemma 4, which is crucial to proving the worst-case results for the **NFI** heuristic. Let $X_j$ be the set of items in the $j$th bin produced by the heuristic **NFI** on list $L$ for $j = 1, 2, \ldots, m$, and let $S_j$ be the set of items in the $j$th bin produced by procedure **A** on list $L$ for $j = 1, 2, \ldots, m'$. In addition, let $Z_f^{\mathrm{A}}(L)$ denote the cost of the solution produced by procedure **A** using cost function $f$. From Property 1 we know that $m' \leq m$. So for notational purposes let $S_{m'+1} = S_{m'+2} = \ldots = S_m = \emptyset$, and as in the previous section define $\overline{X}_j = \cup_{i=1}^{j} X_i$ and $\overline{S}_j = \cup_{i=1}^{j} S_i$ for $j = 1, 2, \ldots, m$.

The following lemma characterizes the absolute performance ratio of **NFI** on the family of flat cost functions described in Section 2.

**Lemma 4.** *For any cost function* $f_k$, $k \geq 1$, *and for all* $L \in \mathscr{L}$ *we have*

$$\frac{Z_{f_k}^{\mathrm{NFI}}(L)}{Z_{f_k}^{*}(L)} \leq \frac{Z_{f_k}^{\mathrm{NFI}}(L)}{Z_{f_k}^{\mathrm{A}}(L)} \leq 1 + \frac{1}{k}.$$

**Proof.** All the bins in the solution produced by procedure **A** are filled over the capacity, except bin 1 and possibly bin $m'$. We now construct a feasible packing from this solution. For each bin $i$, $i = 1, 2, \ldots, m'$, we open a new bin to which we transfer the largest element of $S_i$. Clearly, the new packing consists of consecutive bins and for any cost function $f \in \mathscr{F}$ its total cost equals:

$$\sum_{i=1}^{m'} [f(|S_i| - 1) + f(1)] = \sum_{i=1}^{m'} f(|S_i| - 1) + m'.$$

In view of Corollary 2,

$$Z_f^{\mathrm{NFI}}(L) \leq \sum_{i=1}^{m'} f(|S_i| - 1) + m'.$$

Let $I_{(B)}$ be the indicator function of the Boolean variable $B$. Consider any cost function of the form $f_k$ for

$k \geq 1$, then we have

$$\begin{aligned}\frac{Z_{f_k}^{\mathrm{NFI}}(L)}{Z_{f_k}^{\mathrm{A}}(L)} &\leq \frac{\sum_{i=1}^{m'} f_k(|S_i| - 1) + m'}{\sum_{i=1}^{m'} f_k(|S_i|)} \\ &= \frac{\mathscr{A}}{\sum_{i=1}^{m'} f_k(|S_i|)} \\ &= 1 + \frac{\sum_{i=1}^{m'} I_{(|S_i|>k)}}{\sum_{i=1}^{m'} (|S_i| \cdot I_{(|S_i|\leq k)} + k \cdot I_{(|S_i|>k)})} \\ &\leq 1 + \frac{1}{k}.\end{aligned}$$

where $\mathscr{A} = \sum_{i=1}^{m'} f_k(|S_i|) + \sum_{i=1}^{m'} [f_k(|S_i| - 1) + f_k(1) - f_k(|S_i|)]$.

The second equality follows from the fact that

$$f_k(|S_i| - 1) + f_k(1) - f_k(|S_i|) = I_{(|S_i|>k)} = \begin{cases} 1, & \text{if } |S_i| > k; \\ 0, & \text{if } |S_i| \leq k. \end{cases}$$

We can now complete the proof of Theorem 6.

**Proof.** Due to Theorem 1, there exists an integer $k \geq 1$ such that

$$\frac{Z_f^{\mathrm{NFI}}(L)}{Z_f^*(L)} \leq \frac{Z_{f_k}^{\mathrm{NFI}}(L)}{Z_{f_k}^*(L)}.$$

If $k = 1$, then from Section 5 we know that **NFI** has an absolute performance ratio of at most 1.75 for the classical bin packing problem. To derive the second value in the bound, we recall the tight asymptotic performance bound for **NFD**, proved by Baker and Coffman. The authors show that $b^{\mathrm{NFD}}(L) \leq 1.691 \ldots b^*(L) + 3$. Therefore, by Property 3, $R_{f_1}^{\mathrm{NFI}}(L) \leq 1.691 \ldots + 3/b^*(L)$. The third value in the upper bound for $k = 1$ is an immediate consequence of Johnson et al., who showed that the number of bins produced by any **BF** or **FF** heuristic is no more than $1.7 + 2/b^*(L)$ times the optimal number of bins. In fact, it is easy to show that for a nondecreasing sequence of items **NF** and **BF** (or **FF**) are identical, and hence we have $Z_{f_1}^{\mathrm{NFI}}(L) \leq 1.7 + 2/b^*(L)$. On the other hand, if $k \geq 2$, then Lemma 4 tells us that the absolute performance ratio for **NFI** with respect to $f_k$ is at most 3/2. Hence the theorem is proven. Clearly from these results, $R_f^{\mathrm{NFI}}(\infty) \leq 1.691 \ldots$ for all $f \in \mathscr{F}$, and the bound is tight for $f_1$.

## 7. EXTENSIONS AND CONCLUSIONS

In this section we look at a number of possible extensions of our analysis and results. We first look at the question of how well a heuristic can perform in terms of asymptotic worst-case performance, if it does not take into account the structure of the cost function $f$. The following theorem, whose proof is given in Bramel, provides a lower bound on the asymptotic performance ratio of any heuristic that has no information about the cost function.

**Theorem 7.** *If* **H** *does not use any information about the cost function $f$, then $R_f^{\mathrm{H}}(\infty) \geq 4/3$.*

This result merely says that the techniques developed in this paper cannot be extended to produce heuristics with asymptotic performance ratios better than 4/3. However, algorithms that make use of the particular objective function may well have superior performance guarantees.

Next, we try to extend our results by relaxing the monotonicity assumption of the cost function $f$. We observe that the proof of Lemma 2 uses only the concavity assumption of the cost function and, therefore, the solution produced by procedure **A** still yields a valid lower bound even if we relax our assumption that $f$ is a monotone function. The monotonicity of the cost function is needed, however, for the purpose of worst-case analysis. If the monotonicity requirement is relaxed, one can construct a nonempty list where the optimal solution has zero cost, which would make any finite worst-case bound impossible to attain.

Another possible extension of our results is based on observing that the cost structure analyzed in the previous sections is a special case of a submodular cost function. Hence, one may consider worst-case analysis of other submodular cost structures. However, the heuristics we have found to have good performance ratios for the previous cost structure do not have good performance ratios for general submodular costs, as can be seen by Theorem 8. Consider the following cost structure: Associated with each item are two attributes, a real number $r_i$ and its size $w_i$. Let the cost of a bin containing items represented by the set $S$ be $F(S) = \max_{i \in S}\{r_i\}$. It is easy to verify that this cost is submodular. Now construct a list in the following manner: Starting with $L = \emptyset$, for each integer $1 \leq i \leq n$, add to the list one item of size $1/2^i$, and attribute value $n$, and $2^i - 1$ items of size $1/2^i$, and attribute value 1. In the end we have a total of $\sum_{i=1}^{n} 2^i$ items. A straightforward calculation shows that the **NFI** and **NFD** heuristics produce a solution with a total cost $n^2$, while the optimal cost is $2n - 1$. We thus have Theorem 8.

**Theorem 8.** *There exist submodular cost structures for which* **NFI** *and* **NFD** *do not have finite performance ratios.*

## ACKNOWLEDGMENT

The research of the second and third authors was supported in part by ONR contract N00014-90-I-1649 and NSF contract DDM-8922712.

## REFERENCES

ANILY, S., AND A. FEDERGRUEN. 1990. A Class of Euclidean Routing Problems With General Route Cost Functions. *Math. Opns. Res.* **15,** 268–285.

ANILY, S., AND A. FEDERGRUEN. 1991. Structured Partitioning Problems. *Opns. Res.* **39,** 130–149.

BAKER, B. S., AND E. G. COFFMAN JR. 1981. A Tight Asymptotic Bound for Next-Fit-Decreasing Bin-Packing. *SIAM J. Alg. Disc. Meth.* **2,** 147–152.

BARNES, E. R., AND A. J. HOFFMAN. 1984. Partitioning, Spectra, and Linear Programming. In *Progress in Combinatorial Optimization.* W. Pulleyblank (ed.). Academic Press, Toronto, 13–26.

BRAMEL, J. 1992. Analytical Results and Algorithms for Vehicle Routing Problems With Bin-Packing Features. Ph.D. Thesis, Columbia University, New York.

BRAMEL, J., E. G. COFFMAN, JR., P. SHOR AND D. SIMCHI-LEVI. 1992. Probabilistic Analysis of Algorithms for the Capacitated Vehicle Routing Problem With Unsplit Demands. *Opns. Res.* **40,** 1095–1106.

CHAKRAVARTY, A. K., J. B. ORLIN AND U. G. ROTHBLUM. 1985. Consecutive Optimizers for a Partitioning Problem With Applications to Optimal Inventory Groupings for Joint Replenishment. *Opns. Res.* **33,** 820–834.

COFFMAN, E. G. JR., M. R. GAREY AND D. S. JOHNSON. 1984. Approximation Algorithms for Bin Packing—An Updated Survey. In *Algorithm Design for Computer System Design,* G. Ausiello, M. Lucertini, and P. Serafini (eds.). Springer-Verlag, Berlin and New York, 49–106.

HWANG, F. K. 1981. Optimal Partitions. *J. Optim. Theory Appl.* **34,** 1–10.

JOHNSON, D. S., A. DEMERS, J. D. ULLMAN, M. R. GAREY AND R. L. GRAHAM. 1974. Worst-Case Performance Bounds for Simple One-Dimensional Packing Algorithms. *SIAM J. Comput.* **3,** 299–325.

SIMCHI-LEVI, D., AND J. BRAMEL. 1990. On the Optimal Solution Value of the Capacitated Vehicle Routing Problem With Unsplit Demands. Working Paper, Columbia University, New York.